

*Dynamic Programming  
and Optimal Control*

*Volume I*

THIRD EDITION

Dimitri P. Bertsekas

Massachusetts Institute of Technology

WWW site for book information and orders

<http://www.athenase.com>



Athena Scientific, Belmont, Massachusetts

Athena Scientific  
Post Office Box 805  
Nashua, NH 03061-0805  
U.S.A.

Email: [info@athenasc.com](mailto:info@athenasc.com)  
WWW: <http://www.athenasc.com>

Cover Design: Ann Gallagher, [www.gallagherdesign.com](http://www.gallagherdesign.com)

© 2005, 2000, 1995 Dimitri P. Bertsekas  
All rights reserved. No part of this book may be reproduced in any form  
by any electronic or mechanical means (including photocopying, recording,  
or information storage and retrieval) without permission in writing from  
the publisher.

#### Publisher's Cataloging-in-Publication Data

Bertsekas, Dimitri P.  
Dynamic Programming and Optimal Control  
Includes Bibliography and Index  
1. Mathematical Optimization. 2. Dynamic Programming. I. Title.  
QA402.5 .B465 2005      519.703      00-91281

ISBN 1-886529-26-4

## ABOUT THE AUTHOR

Dimitri Bertsekas studied Mechanical and Electrical Engineering at the National Technical University of Athens, Greece, and obtained his Ph.D. in system science from the Massachusetts Institute of Technology. He has held faculty positions with the Engineering-Economic Systems Dept., Stanford University, and the Electrical Engineering Dept. of the University of Illinois, Urbana. Since 1979 he has been teaching at the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (M.I.T.), where he is currently McAfee Professor of Engineering.

His research spans several fields, including optimization, control, large-scale computation, and data communication networks, and is closely tied to his teaching and book authoring activities. He has written numerous research papers, and thirteen books, several of which are used as textbooks in MIT classes. He consults regularly with private industry and has held editorial positions in several journals.

Professor Bertsekas was awarded the INFORMS 1997 Prize for Research Excellence in the Interface Between Operations Research and Computer Science for his book "Neuro-Dynamic Programming" (co-authored with John Tsitsiklis), the 2000 Greek National Award for Operations Research, and the 2001 ACC John R. Ragazzini Education Award. In 2001, he was elected to the United States National Academy of Engineering.

**ATHENA SCIENTIFIC**  
**OPTIMIZATION AND COMPUTATION SERIES**

1. Convex Analysis and Optimization, by Dimitri P. Bertsekas, with Angelia Nedić and Asuman E. Ozdaglar, 2003, ISBN 1-886529-45-0, 560 pages
2. Introduction to Probability, by Dimitri P. Bertsekas and John N. Tsitsiklis, 2002, ISBN 1-886529-40-X, 430 pages
3. Dynamic Programming and Optimal Control, Two-Volume Set, by Dimitri P. Bertsekas, 2005, ISBN 1-886529-08-6, 840 pages
4. Nonlinear Programming, 2nd Edition, by Dimitri P. Bertsekas, 1999, ISBN 1-886529-00-0, 791 pages
5. Network Optimization: Continuous and Discrete Models, by Dimitri P. Bertsekas, 1998, ISBN 1-886529-02-7, 608 pages
6. Network Flows and Monotropic Optimization, by R. Tyrrell Rockafellar, 1998, ISBN 1-886529-06-X, 634 pages
7. Introduction to Linear Optimization, by Dimitris Bertsimas and John N. Tsitsiklis, 1997, ISBN 1-886529-19-1, 608 pages
8. Parallel and Distributed Computation: Numerical Methods, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1997, ISBN 1-886529-01-9, 718 pages
9. Neuro-Dynamic Programming, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1996, ISBN 1-886529-10-8, 512 pages
10. Constrained Optimization and Lagrange Multiplier Methods, by Dimitri P. Bertsekas, 1996, ISBN 1-886529-04-3, 410 pages
11. Stochastic Optimal Control: The Discrete-Time Case, by Dimitri P. Bertsekas and Steven E. Shreve, 1996, ISBN 1-886529-03-5, 330 pages

## *Contents*

<b>1. The Dynamic Programming Algorithm</b>	
1.1. Introduction . . . . .	p. 2
1.2. The Basic Problem . . . . .	p. 12
1.3. The Dynamic Programming Algorithm . . . . .	p. 18
1.4. State Augmentation and Other Reformulations . . . . .	p. 35
1.5. Some Mathematical Issues . . . . .	p. 42
1.6. Dynamic Programming and Minimax Control . . . . .	p. 46
1.7. Notes, Sources, and Exercises . . . . .	p. 51
<b>2. Deterministic Systems and the Shortest Path Problem</b>	
2.1. Finite-State Systems and Shortest Paths . . . . .	p. 64
2.2. Some Shortest Path Applications . . . . .	p. 68
2.2.1. Critical Path Analysis . . . . .	p. 68
2.2.2. Hidden Markov Models and the Viterbi Algorithm .	p. 70
2.3. Shortest Path Algorithms . . . . .	p. 77
2.3.1. Label Correcting Methods . . . . .	p. 78
2.3.2. Label Correcting Variations - $A^*$ Algorithm . .	p. 87
2.3.3. Branch-and-Bound . . . . .	p. 88
2.3.4. Constrained and Multiobjective Problems . . . .	p. 91
2.4. Notes, Sources, and Exercises . . . . .	p. 97
<b>3. Deterministic Continuous-Time Optimal Control</b>	
3.1. Continuous-Time Optimal Control . . . . .	p. 106
3.2. The Hamilton-Jacobi-Bellman Equation . . . . .	p. 109
3.3. The Pontryagin Minimum Principle . . . . .	p. 115
3.3.1. An Informal Derivation Using the HJB Equation	p. 115
3.3.2. A Derivation Based on Variational Ideas . . . .	p. 125
3.3.3. Minimum Principle for Discrete-Time Problems	p. 129
3.4. Extensions of the Minimum Principle . . . . .	p. 131
3.4.1. Fixed Terminal State . . . . .	p. 131
3.4.2. Free Initial State . . . . .	p. 135

*Contents*

3.4.3. Free Terminal Time . . . . .	p. 135
3.4.4. Time-Varying System and Cost . . . . .	p. 138
3.4.5. Singular Problems . . . . .	p. 139
3.5. Notes, Sources, and Exercises . . . . .	p. 142
<b>4. Problems with Perfect State Information</b>	
4.1. Linear Systems and Quadratic Cost . . . . .	p. 148
4.2. Inventory Control . . . . .	p. 162
4.3. Dynamic Portfolio Analysis . . . . .	p. 170
4.4. Optimal Stopping Problems . . . . .	p. 176
4.5. Scheduling and the Interchange Argument . . . . .	p. 186
4.6. Set-Membership Description of Uncertainty . . . . .	p. 190
4.6.1. Set-Membership Estimation . . . . .	p. 191
4.6.2. Control with Unknown-but-Bounded Disturbances . . . . .	p. 197
4.7. Notes, Sources, and Exercises . . . . .	p. 201
<b>5. Problems with Imperfect State Information</b>	
5.1. Reduction to the Perfect Information Case . . . . .	p. 218
5.2. Linear Systems and Quadratic Cost . . . . .	p. 229
5.3. Minimum Variance Control of Linear Systems . . . . .	p. 236
5.4. Sufficient Statistics and Finite-State Markov Chains . . . . .	p. 251
5.4.1. The Conditional State Distribution . . . . .	p. 252
5.4.2. Finite-State Systems . . . . .	p. 258
5.5. Notes, Sources, and Exercises . . . . .	p. 270
<b>6. Suboptimal Control</b>	
6.1. Certainty Equivalent and Adaptive Control . . . . .	p. 283
6.1.1. Caution, Probing, and Dual Control . . . . .	p. 289
6.1.2. Two-Phase Control and Identifiability . . . . .	p. 291
6.1.3. Certainty Equivalent Control and Identifiability . . . . .	p. 293
6.1.4. Self-Tuning Regulators . . . . .	p. 298
6.2. Open-Loop Feedback Control . . . . .	p. 300
6.3. Limited Lookahead Policies . . . . .	p. 304
6.3.1. Performance Bounds for Limited Lookahead Policies . . . . .	p. 305
6.3.2. Computational Issues in Limited Lookahead . . . . .	p. 310
6.3.3. Problem Approximation - Enforced Decomposition . . . . .	p. 312
6.3.4. Aggregation . . . . .	p. 319
6.3.5. Parametric Cost-to-Go Approximation . . . . .	p. 319
6.4. Rollout Algorithms . . . . .	p. 325
6.4.1. Discrete Deterministic Problems . . . . .	p. 335
6.4.2. Q-Factors Evaluated by Simulation . . . . .	p. 342
6.4.3. Q-Factor Approximation . . . . .	p. 361
	p. 363

*Contents*

6.5. Model Predictive Control and Related Methods . . . . .	p. 366
6.5.1. Rolling Horizon Approximations . . . . .	p. 367
6.5.2. Stability Issues in Model Predictive Control . . . . .	p. 369
6.5.3. Restricted Structure Policies . . . . .	p. 376
6.6. Additional Topics in Approximate DP . . . . .	p. 382
6.6.1. Discretization . . . . .	p. 382
6.6.2. Other Approximation Approaches . . . . .	p. 384
6.7. Notes, Sources, and Exercises . . . . .	p. 386

**7. Introduction to Infinite Horizon Problems**

7.1. An Overview . . . . .	p. 402
7.2. Stochastic Shortest Path Problems . . . . .	p. 405
7.3. Discounted Problems . . . . .	p. 417
7.4. Average Cost per Stage Problems . . . . .	p. 421
7.5. Semi-Markov Problems . . . . .	p. 435
7.6. Notes, Sources, and Exercises . . . . .	p. 445

**Appendix A: Mathematical Review**

A.1. Sets . . . . .	p. 459
A.2. Euclidean Space . . . . .	p. 460
A.3. Matrices . . . . .	p. 461
A.4. Analysis . . . . .	p. 465
A.5. Convex Sets and Functions . . . . .	p. 467

**Appendix B: On Optimization Theory**

B.1. Optimal Solutions . . . . .	p. 468
B.2. Optimality Conditions . . . . .	p. 470
B.3. Minimization of Quadratic Forms . . . . .	p. 471

**Appendix C: On Probability Theory**

C.1. Probability Spaces . . . . .	p. 472
C.2. Random Variables . . . . .	p. 473
C.3. Conditional Probability . . . . .	p. 475

**Appendix D: On Finite-State Markov Chains**

D.1. Stationary Markov Chains . . . . .	p. 477
D.2. Classification of States . . . . .	p. 478
D.3. Limiting Probabilities . . . . .	p. 479
D.4. First Passage Times . . . . .	p. 480

**Appendix E: Kalman Filtering**

E.1. Least-Squares Estimation . . . . .	p. 481
E.2. Linear Least-Squares Estimation . . . . .	p. 483
E.3. State Estimation – Kalman Filter . . . . .	p. 491
E.4. Stability Aspects . . . . .	p. 496
E.5. Gauss-Markov Estimators . . . . .	p. 499
E.6. Deterministic Least-Squares Estimation . . . . .	p. 501

**Appendix F: Modeling of Stochastic Linear Systems**

F.1. Linear Systems with Stochastic Inputs . . . . .	p. 503
F.2. Processes with Rational Spectrum . . . . .	p. 504
F.3. The ARMAX Model . . . . .	p. 506

**Appendix G: Formulating Problems of Decision Under Uncertainty**

G.1. The Problem of Decision Under Uncertainty . . . . .	p. 507
G.2. Expected Utility Theory and Risk . . . . .	p. 511
G.3. Stochastic Optimal Control Problems . . . . .	p. 524

**References . . . . .**

p. 529

**Index . . . . .**

p. 541

**CONTENTS OF VOLUME II****1. Infinite Horizon – Discounted Problems**

1.1. Minimization of Total Cost – Introduction
1.2. Discounted Problems with Bounded Cost per Stage
1.3. Finite-State Systems – Computational Methods
1.3.1. Value Iteration and Error Bounds
1.3.2. Policy Iteration
1.3.3. Adaptive Aggregation
1.3.4. Linear Programming
1.3.5. Limited Lookahead Policies
1.4. The Role of Contraction Mappings
1.5. Scheduling and Multiarmed Bandit Problems
1.6. Notes, Sources, and Exercises

**2. Stochastic Shortest Path Problems**

2.1. Main Results
2.2. Computational Methods
2.2.1. Value Iteration
2.2.2. Policy Iteration
2.3. Simulation-Based Methods
2.3.1. Policy Evaluation by Monte-Carlo Simulation
2.3.2. <i>Q</i> -Learning
2.3.3. Approximations
2.3.4. Extensions to Discounted Problems
2.3.5. The Role of Parallel Computation
2.4. Notes, Sources, and Exercises

**3. Undiscounted Problems**

3.1. Unbounded Costs per Stage
3.2. Linear Systems and Quadratic Cost
3.3. Inventory Control
3.4. Optimal Stopping
3.5. Optimal Gambling Strategies
3.6. Nonstationary and Periodic Problems
3.7. Notes, Sources, and Exercises

**4. Average Cost per Stage Problems**

4.1. Preliminary Analysis
4.2. Optimality Conditions
4.3. Computational Methods
4.3.1. Value Iteration

- 4.3.2. Policy Iteration
- 4.3.3. Linear Programming
- 4.3.4. Simulation-Based Methods
- 4.4. Infinite State Space
- 4.5. Notes, Sources, and Exercises

## 5. Continuous-Time Problems

- 5.1. Uniformization
- 5.2. Queueing Applications
- 5.3. Semi-Markov Problems
- 5.4. Notes, Sources, and Exercises

## References

## Index

# *Preface*

This two-volume book is based on a first-year graduate course on dynamic programming and optimal control that I have taught for over twenty years at Stanford University, the University of Illinois, and the Massachusetts Institute of Technology. The course has been typically attended by students from engineering, operations research, economics, and applied mathematics. Accordingly, a principal objective of the book has been to provide a unified treatment of the subject, suitable for a broad audience. In particular, problems with a continuous character, such as stochastic control problems, popular in modern control theory, are simultaneously treated with problems with a discrete character, such as Markovian decision problems, popular in operations research. Furthermore, many applications and examples, drawn from a broad variety of fields, are discussed.

The book may be viewed as a greatly expanded and pedagogically improved version of my 1987 book "Dynamic Programming: Deterministic and Stochastic Models," published by Prentice-Hall. I have included much new material on deterministic and stochastic shortest path problems, as well as a new chapter on continuous-time optimal control problems and the Pontryagin Minimum Principle, developed from a dynamic programming viewpoint. I have also added a fairly extensive exposition of simulation-based approximation techniques for dynamic programming. These techniques, which are often referred to as "neuro-dynamic programming" or "reinforcement learning," represent a breakthrough in the practical application of dynamic programming to complex problems that involve the dual curse of large dimension and lack of an accurate mathematical model. Other material was also augmented, substantially modified, and updated.

With the new material, however, the book grew so much in size that it became necessary to divide it into two volumes: one on finite horizon, and the other on infinite horizon problems. This division was not only natural in terms of size, but also in terms of style and orientation. The first volume is more oriented towards modeling, and the second is more oriented towards mathematical analysis and computation. I have included in the first volume a final chapter that provides an introductory treatment of infinite horizon problems. The purpose is to make the first volume self-

contained for instructors who wish to cover a modest amount of infinite horizon material in a course that is primarily oriented towards modeling, conceptualization, and finite horizon problems.

Many topics in the book are relatively independent of the others. For example Chapter 2 of Vol. I on shortest path problems can be skipped without loss of continuity, and the same is true for Chapter 3 of Vol. I, which deals with continuous-time optimal control. As a result, the book can be used to teach several different types of courses.

- (a) A two-semester course that covers both volumes.
- (b) A one-semester course primarily focused on finite horizon problems that covers most of the first volume.
- (c) A one-semester course focused on stochastic optimal control that covers Chapters 1, 4, 5, and 6 of Vol. I, and Chapters 1, 2, and 4 of Vol. II.
- (d) A one-semester course that covers Chapter 1, about 50% of Chapters 2 through 6 of Vol. I, and about 70% of Chapters 1, 2, and 4 of Vol. II. This is the course I usually teach at MIT.
- (e) A one-quarter engineering course that covers the first three chapters and parts of Chapters 4 through 6 of Vol. I.
- (f) A one-quarter mathematically oriented course focused on infinite horizon problems that covers Vol. II.

The mathematical prerequisite for the text is knowledge of advanced calculus, introductory probability theory, and matrix-vector algebra. A summary of this material is provided in the appendixes. Naturally, prior exposure to dynamic system theory, control, optimization, or operations research will be helpful to the reader, but based on my experience, the material given here is reasonably self-contained.

The book contains a large number of exercises, and the serious reader will benefit greatly by going through them. Solutions to all exercises are compiled in a manual that is available to instructors from the author. Many thanks are due to the several people who spent long hours contributing to this manual, particularly Steven Shreve, Eric Loiederman, Lakis Polymenakos, and Cynara Wu.

Dynamic programming is a conceptually simple technique that can be adequately explained using elementary analysis. Yet a mathematically rigorous treatment of general dynamic programming requires the complicated machinery of measure-theoretic probability. My choice has been to bypass the complicated mathematics by developing the subject in generality, while claiming rigor only when the underlying probability spaces are countable. A mathematically rigorous treatment of the subject is carried out in my monograph "Stochastic Optimal Control: The Discrete Time

Case," Academic Press, 1978,<sup>†</sup> coauthored by Steven Shreve. This monograph complements the present text and provides a solid foundation for the subjects developed somewhat informally here.

Finally, I am thankful to a number of individuals and institutions for their contributions to the book. My understanding of the subject was sharpened while I worked with Steven Shreve on our 1978 monograph. My interaction and collaboration with John Tsitsiklis on stochastic shortest paths and approximate dynamic programming have been most valuable. Michael Caramanis, Emmanuel Fernandez-Gaucherand, Pierre Humbert, Lennart Ljung, and John Tsitsiklis taught from versions of the book, and contributed several substantive comments and homework problems. A number of colleagues offered valuable insights and information, particularly David Castanon, Eugene Feinberg, and Krishna Pattipati. NSF provided research support. Prentice-Hall graciously allowed the use of material from my 1987 book. Teaching and interacting with the students at MIT have kept up my interest and excitement for the subject.

Dimitri P. Bertsekas  
Spring, 1995

---

<sup>†</sup> Note added in the 3rd edition: This monograph was republished by Athena Scientific in 1996, and can also be freely downloaded from the author's www site: <http://web.mit.edu/dimitrib/www/home.html>.

## Preface to the Second Edition

This second edition has expanded by nearly 30% the coverage of the original. Most of the new material is concentrated in four areas:

- (a) In Chapter 4, a section was added on estimation and control of systems with a non-probabilistic (set membership) description of uncertainty. This subject, a personal favorite of the author since it was the subject of his 1971 Ph.D. thesis, has become popular, as minimax and  $H_\infty$  control methods have gained increased prominence.
- (b) Chapter 6 was doubled in size, to reflect the popularity of suboptimal control and neuro-dynamic programming methods. In particular, the coverage of certainty equivalent, and limited lookahead methods has been substantially expanded. Furthermore, a new section was added on neuro-dynamic programming and rollout algorithms, and their applications in combinatorial optimization and stochastic optimal control.
- (c) In Chapter 7, an introduction to continuous-time, semi-Markov decision problems was added in a separate last section.
- (d) A new appendix was included, which deals with various formulations of problems of decision under uncertainty. The foundations of the minimax and expected utility approaches are framed within a broader context, and some of the aspects of utility theory are discussed.

There are also miscellaneous additions and improvements scattered throughout the text, and a more detailed coverage of deterministic problems is given in Chapter 1. Finally, a new internet-based feature was added to the book, which extends its scope and coverage. Many of the theoretical exercises have been solved in detail and their solutions have been posted in the book's www page

<http://www.athenasc.com/dpbook.html>

These exercises have been marked with the symbol 

I would like to express my thanks to the many colleagues who contributed suggestions for improvement of the second edition.

Dimitri P. Bertsekas  
Fall, 2000

## Preface to the Third Edition

The third edition contains a substantial amount of new material, particularly on approximate dynamic programming, which has now become one of the principal focal points of the book. In particular:

- (a) The subject of minimax control was developed in greater detail, including a new section in Chapter 1, which connects with new material in Chapter 6.
- (b) The section on auction algorithms for shortest paths in Chapter 2 was eliminated. These methods are not currently used in dynamic programming, and a detailed discussion has been provided in a chapter from the author's Network Optimization book. This chapter can be freely downloaded from  
<http://web.mit.edu/dimitrib/www/net.html>
- (c) A section was added in Chapter 2 on dynamic programming and shortest path algorithms for constrained and multiobjective problems.
- (d) The material on sufficient statistics and partially observable Markov decision problems in Section 5.4 was restructured and expanded.
- (e) Considerable new material was added in Chapter 6:
  - (1) An expanded discussion of one-step lookahead policies and associated performance bounds in Section 6.3.1.
  - (2) A discussion of aggregation methods and discretization of continuous-state problems (see Subsection 6.3.4).
  - (3) A discussion of model predictive control and its relation to other suboptimal control methods (see Subsection 6.5.2).
  - (4) An expanded treatment of open-loop feedback control and related methods based on a restricted structure (see Subsection 6.5.3).

I have also added a few exercises, and revised a few sections while preserving their essential content. Thanks are due to Haixia Liu, who worked out several exercises, and to Janey Yu, who reviewed some of the new sections and gave me valuable feedback.

Dimitri P. Bertsekas  
<http://web.mit.edu/dimitrib/www/home.html>  
Summer 2005

# *The Dynamic Programming Algorithm*

## Contents

1.1. Introduction . . . . .	p. 2
1.2. The Basic Problem . . . . .	p. 12
1.3. The Dynamic Programming Algorithm . . . . .	p. 18
1.4. State Augmentation and Other Reformulations . . . . .	p. 35
1.5. Some Mathematical Issues . . . . .	p. 42
1.6. Dynamic Programming and Minimax Control . . . . .	p. 46
1.7. Notes, Sources, and Exercises . . . . .	p. 51

Life can only be understood going backwards,  
but it must be lived going forwards.

Kierkegaard

## 1.1 INTRODUCTION

This book deals with situations where decisions are made in stages. The outcome of each decision may not be fully predictable but can be anticipated to some extent before the next decision is made. The objective is to minimize a certain cost – a mathematical expression of what is considered an undesirable outcome.

A key aspect of such situations is that decisions cannot be viewed in isolation since one must balance the desire for low present cost with the undesirability of high future costs. The dynamic programming technique captures this tradeoff. At each stage, it ranks decisions based on the sum of the present cost and the expected future cost, assuming optimal decision making for subsequent stages.

There is a very broad variety of practical problems that can be treated by dynamic programming. In this book, we try to keep the main ideas uncluttered by irrelevant assumptions on problem structure. To this end, we formulate in this section a broadly applicable model of optimal control of a dynamic system over a finite number of stages (a finite horizon). This model will occupy us for the first six chapters; its infinite horizon version will be the subject of the last chapter as well as Vol. II.

Our basic model has two principal features: (1) an underlying *discrete-time dynamic system*, and (2) a *cost function that is additive over time*. The dynamic system expresses the evolution of some variables, the system's "state", under the influence of decisions made at discrete instances of time. The system has the form

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N - 1,$$

where

$k$  indexes discrete time,

$x_k$  is the state of the system and summarizes past information that is relevant for future optimization,

$u_k$  is the control or decision variable to be selected at time  $k$ ,

$w_k$  is a random parameter (also called disturbance or noise depending on the context),

### Sec. 1.1 Introduction

$N$  is the horizon or number of times control is applied, and  $f_k$  is a function that describes the system and in particular the mechanism by which the state is updated.

The cost function is additive in the sense that the cost incurred at time  $k$ , denoted by  $g_k(x_k, u_k, w_k)$ , accumulates over time. The total cost is

$$g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k),$$

where  $g_N(x_N)$  is a terminal cost incurred at the end of the process. However, because of the presence of  $w_k$ , the cost is generally a random variable and cannot be meaningfully optimized. We therefore formulate the problem as an optimization of the *expected cost*

$$E \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\},$$

where the expectation is with respect to the joint distribution of the random variables involved. The optimization is over the controls  $u_0, u_1, \dots, u_{N-1}$ , but some qualification is needed here; each control  $u_k$  is selected with some knowledge of the current state  $x_k$  (either its exact value or some other related information).

A more precise definition of the terminology just used will be given shortly. We first provide some orientation by means of examples.

#### Example 1.1.1 (Inventory Control)

Consider a problem of ordering a quantity of a certain item at each of  $N$  periods so as to (roughly) meet a stochastic demand, while minimizing the incurred expected cost. Let us denote

$x_k$  stock available at the beginning of the  $k$ th period,

$u_k$  stock ordered (and immediately delivered) at the beginning of the  $k$ th period,

$w_k$  demand during the  $k$ th period with given probability distribution.

We assume that  $w_0, w_1, \dots, w_{N-1}$  are independent random variables, and that excess demand is backlogged and filled as soon as additional inventory becomes available. Thus, stock evolves according to the discrete-time equation

$$x_{k+1} = x_k + u_k - w_k,$$

where negative stock corresponds to backlogged demand (see Fig. 1.1.4).

The cost incurred in period  $k$  consists of two components:

- (a) A cost  $r(x_k)$  representing a penalty for either positive stock  $x_k$  (holding cost for excess inventory) or negative stock  $x_k$  (shortage cost for unfilled demand).

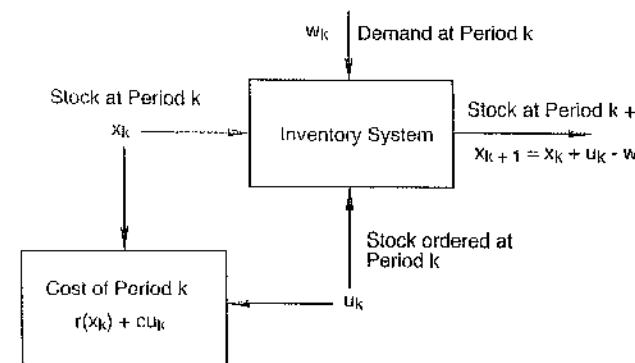


Figure 1.1.1 Inventory control example. At period  $k$ , the current stock (state)  $x_k$ , the stock ordered (control)  $u_k$ , and the demand (random disturbance)  $w_k$  determine the cost  $r(x_k) + cu_k$  and the stock  $x_{k+1} = x_k + u_k - w_k$  at the next period.

(b) The purchasing cost  $cu_k$ , where  $c$  is cost per unit ordered.

There is also a terminal cost  $R(x_N)$  for being left with inventory  $x_N$  at the end of  $N$  periods. Thus, the total cost over  $N$  periods is

$$E \left\{ R(x_N) + \sum_{k=0}^{N-1} (r(x_k) + cu_k) \right\}.$$

We want to minimize this cost by proper choice of the orders  $u_0, \dots, u_{N-1}$ , subject to the natural constraint  $u_k \geq 0$  for all  $k$ .

At this point we need to distinguish between *closed-loop* and *open-loop* minimization of the cost. In open-loop minimization we select all orders  $u_0, \dots, u_{N-1}$  at once at time 0, without waiting to see the subsequent demand levels. In closed-loop minimization we postpone placing the order  $u_k$  until the last possible moment (time  $k$ ) when the current stock  $x_k$  will be known. The idea is that since there is no penalty for delaying the order  $u_k$  up to time  $k$ , we can take advantage of information that becomes available between times 0 and  $k$  (the demand and stock level in past periods).

Closed-loop optimization is of central importance in dynamic programming and is the type of optimization that we will consider almost exclusively in this book. Thus, in our basic formulation, decisions are made in stages while gathering information between stages that will be used to enhance the quality of the decisions. The effect of this on the structure of the resulting optimization problem is quite profound. In particular, in closed-loop inventory optimization we are not interested in finding optimal numerical values of the orders but rather we want to find an *optimal rule for selecting at each period  $k$  an order  $u_k$  for each possible value of stock  $x_k$  that can conceivably occur*. This is an “action versus strategy” distinction.

Mathematically, in closed-loop inventory optimization, we want to find a sequence of functions  $\mu_k$ ,  $k = 0, \dots, N-1$ , mapping stock  $x_k$  into order  $u_k$

so as to minimize the expected cost. The meaning of  $\mu_k$  is that, for each  $k$  and each possible value of  $x_k$ ,

$\mu_k(x_k) =$  amount that should be ordered at time  $k$  if the stock is  $x_k$ .

The sequence  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$  will be referred to as a *policy* or *control law*. For each  $\pi$ , the corresponding cost for a fixed initial stock  $x_0$  is

$$J_\pi(x_0) = E \left\{ R(x_N) + \sum_{k=0}^{N-1} (r(x_k) + c\mu_k(x_k)) \right\},$$

and we want to minimize  $J_\pi(x_0)$  for a given  $x_0$  over all  $\pi$  that satisfy the constraints of the problem. This is a typical dynamic programming problem. We will analyze this problem in various forms in subsequent sections. For example, we will show in Section 4.2 that for a reasonable choice of the cost function, the optimal ordering policy is of the form

$$\mu_k(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < S_k, \\ 0 & \text{otherwise,} \end{cases}$$

where  $S_k$  is a suitable threshold level determined by the data of the problem. In other words, when stock falls below the threshold  $S_k$ , order just enough to bring stock up to  $S_k$ .

The preceding example illustrates the main ingredients of the basic problem formulation:

(a) A *discrete-time system* of the form

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

where  $f_k$  is some function; for example in the inventory case, we have  $f_k(x_k, u_k, w_k) = x_k + u_k - w_k$ .

(b) *Independent random parameters*  $w_k$ . This will be generalized by allowing the probability distribution of  $w_k$  to depend on  $x_k$  and  $u_k$ ; in the context of the inventory example, we can think of a situation where the level of demand  $w_k$  is influenced by the current stock level  $x_k$ .

(c) A *control constraint*; in the example, we have  $u_k \geq 0$ . In general, the constraint set will depend on  $x_k$  and the time index  $k$ , that is,  $u_k \in U_k(x_k)$ . To see how constraints dependent on  $x_k$  can arise in the inventory context, think of a situation where there is an upper bound  $B$  on the level of stock that can be accommodated, so  $u_k \leq B - x_k$ .

(d) An *additive cost* of the form

$$E \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\},$$

where  $g_k$  are some functions; in the inventory example, we have

$$g_N(x_N) = R(x_N), \quad g_k(x_k, u_k, w_k) = r(x_k) + cu_k.$$

- (e) *Optimization over (closed-loop) policies*, that is, rules for choosing  $u_k$  for each  $k$  and each possible value of  $x_k$ .

### Discrete-State and Finite-State Problems

In the preceding example, the state  $x_k$  was a continuous real variable, and it is easy to think of multidimensional generalizations where the state is an  $n$ -dimensional vector of real variables. It is also possible, however, that the state takes values from a discrete set, such as the integers.

A version of the inventory problem where a discrete viewpoint is more natural arises when stock is measured in whole units (such as cars), each of which is a significant fraction of  $x_k$ ,  $u_k$ , or  $w_k$ . It is more appropriate then to take as state space the set of all integers rather than the set of real numbers. The form of the system equation and the cost per period will, of course, stay the same.

Generally, there are many situations where the state is naturally discrete and there is no continuous counterpart of the problem. Such situations are often conveniently specified in terms of the probabilities of transition between the states. What we need to know is  $p_{ij}(u, k)$ , which is the probability at time  $k$  that the next state will be  $j$ , given that the current state is  $i$ , and the control selected is  $u$ , i.e.,

$$p_{ij}(u, k) = P\{x_{k+1} = j \mid x_k = i, u_k = u\}.$$

This type of state transition can alternatively be described in terms of the discrete-time system equation

$$x_{k+1} = w_k,$$

where the probability distribution of the random parameter  $w_k$  is

$$P\{w_k = j \mid x_k = i, u_k = u\} = p_{ij}(u, k).$$

Conversely, given a discrete-state system in the form

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

together with the probability distribution  $P_k(w_k \mid x_k, u_k)$  of  $w_k$ , we can provide an equivalent transition probability description. The corresponding transition probabilities are given by

$$p_{ij}(u, k) = P_k\{W_k(i, u, j) \mid x_k = i, u_k = u\},$$

### Sec. 1.1 Introduction

where  $W(i, u, j)$  is the set

$$W_k(i, u, j) = \{w \mid j = f_k(i, u, w)\}.$$

Thus a discrete-state system can equivalently be described in terms of a difference equation or in terms of transition probabilities. Depending on the given problem, it may be notationally or mathematically more convenient to use one description over the other.

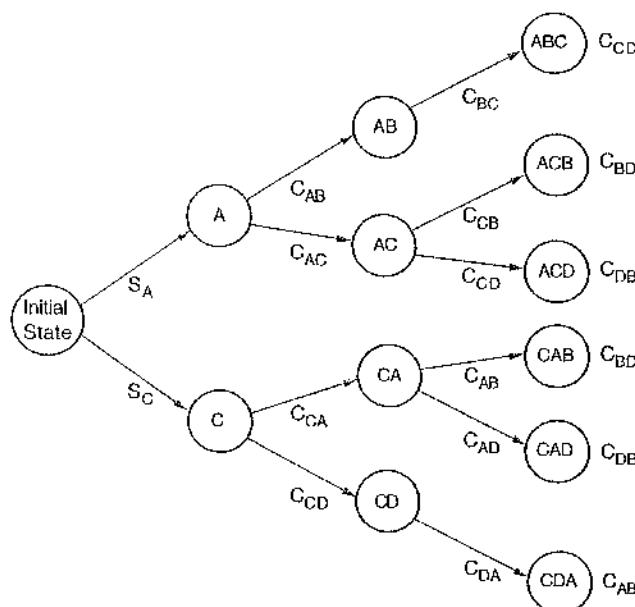
The following examples illustrate discrete-state problems. The first example involves a *deterministic* problem, that is, a problem where there is no stochastic uncertainty. In such a problem, when a control is chosen at a given state, the next state is fully determined; that is, for any state  $i$ , control  $u$ , and time  $k$ , the transition probability  $p_{ij}(u, k)$  is equal to 1 for a single state  $j$ , and it is 0 for all other candidate next states. The other three examples involve stochastic problems, where the next state resulting from a given choice of control at a given state cannot be determined a priori.

#### Example 1.1.2 (A Deterministic Scheduling Problem)

Suppose that to produce a certain product, four operations must be performed on a certain machine. The operations are denoted by A, B, C, and D. We assume that operation B can be performed only after operation A has been performed, and operation D can be performed only after operation B has been performed. (Thus the sequence CDAB is allowable but the sequence CDBA is not.) The setup cost  $C_{mn}$  for passing from any operation  $m$  to any other operation  $n$  is given. There is also an initial startup cost  $S_A$  or  $S_C$  for starting with operation A or C, respectively. The cost of a sequence is the sum of the setup costs associated with it; for example, the operation sequence ACDB has cost

$$S_A + C_{AC} + C_{CD} + C_{DB}.$$

We can view this problem as a sequence of three decisions, namely the choice of the first three operations to be performed (the last operation is determined from the preceding three). It is appropriate to consider as state the set of operations already performed, the initial state being an artificial state corresponding to the beginning of the decision process. The possible state transitions corresponding to the possible states and decisions for this problem is shown in Fig. 1.1.2. Here the problem is deterministic, i.e., at a given state, each choice of control leads to a uniquely determined state. For example, at state AC the decision to perform operation D leads to state ACD with certainty, and has cost  $C_{CD}$ . Deterministic problems with a finite number of states can be conveniently represented in terms of transition graphs such as the one of Fig. 1.1.2. The optimal solution corresponds to the path that starts at the initial state and ends at some state at the terminal time and has minimum sum of arc costs plus the terminal cost. We will study systematically problems of this type in Chapter 2.



**Figure 1.1.2** The transition graph of the deterministic scheduling problem of Example 1.1.2. Each arc of the graph corresponds to a decision leading from some state (the start node of the arc) to some other state (the end node of the arc). The corresponding cost is shown next to the arc. The cost of the last operation is shown as a terminal cost next to the terminal nodes of the graph.

### Example 1.1.3 (Machine Replacement)

Consider a problem of operating efficiently over  $N$  time periods a machine that can be in any one of  $n$  states, denoted  $1, 2, \dots, n$ . We denote by  $g(i)$  the operating cost per period when the machine is in state  $i$ , and we assume that

$$g(1) \leq g(2) \leq \dots \leq g(n).$$

The implication here is that state  $i$  is better than state  $i + 1$ , and state 1 corresponds to a machine in best condition.

During a period of operation, the state of the machine can become worse or it may stay unchanged. We thus assume that the transition probabilities

$$p_{ij} = P\{\text{next state will be } j \mid \text{current state is } i\}$$

satisfy

$$p_{ij} = 0 \quad \text{if } j < i.$$

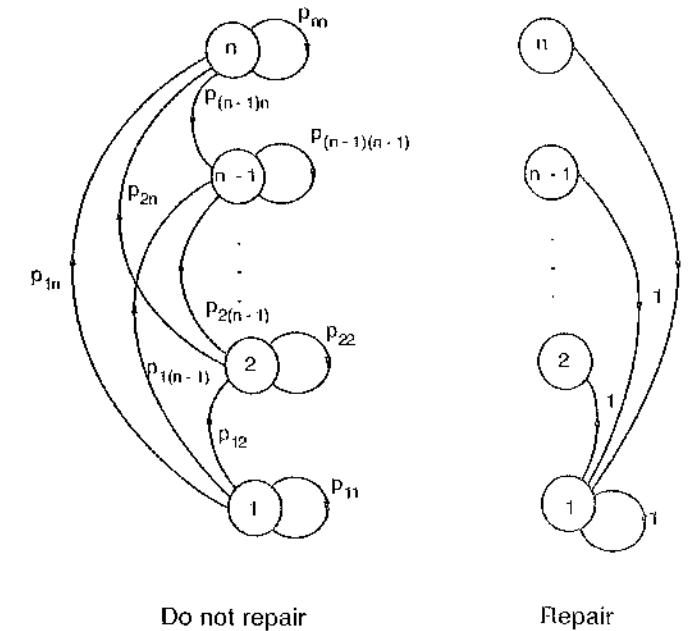
We assume that at the start of each period we know the state of the machine and we must choose one of the following two options:

- (a) Let the machine operate one more period in the state it currently is.
- (b) Repair the machine and bring it to the best state 1 at a cost  $R$ .

We assume that the machine, once repaired, is guaranteed to stay in state 1 for one period. In subsequent periods, it may deteriorate to states  $j > 1$  according to the transition probabilities  $p_{1j}$ .

Thus the objective here is to decide on the level of deterioration (state) at which it is worth paying the cost of machine repair, thereby obtaining the benefit of smaller future operating costs. Note that the decision should also be affected by the period we are in. For example, we would be less inclined to repair the machine when there are few periods left.

The system evolution for this problem can be described by the graphs of Fig. 1.1.3. These graphs depict the transition probabilities between various pairs of states for each value of the control and are known as *transition probability graphs* or simply *transition graphs*. Note that there is a different graph for each control; in the present case there are two controls (repair or not repair).



**Figure 1.1.3** Machine replacement example. Transition probability graphs for each of the two possible controls (repair or not repair). At each stage and state  $i$ , the cost of repairing is  $R + g(1)$ , and the cost of not repairing is  $g(i)$ . The terminal cost is 0.

### Example 1.1.4 (Control of a Queue)

Consider a queueing system with room for  $n$  customers operating over  $N$  time periods. We assume that service of a customer can start (end) only at the beginning (end) of the period and that the system can serve only one customer at a time. The probability  $p_m$  of  $m$  customer arrivals during a period is given, and the numbers of arrivals in two different periods are independent. Customers finding the system full depart without attempting to enter later. The system offers two kinds of service, *fast* and *slow*, with cost per period  $c_f$  and  $c_s$ , respectively. Service can be switched between fast and slow at the beginning of each period. With fast (slow) service, a customer in service at the beginning of a period will terminate service at the end of the period with probability  $q_f$  (respectively,  $q_s$ ) independently of the number of periods the customer has been in service and the number of customers in the system ( $q_f > q_s$ ). There is a cost  $r(i)$  for each period for which there are  $i$  customers in the system. There is also a terminal cost  $R(i)$  for  $i$  customers left in the system at the end of the last period.

The problem is to choose, at each period, the type of service as a function of the number of customers in the system so as to minimize the expected total cost over  $N$  periods. One expects that when there is a large number of customers  $i$  in queue, it is better to use the fast service, and the question is to find the values of  $i$  for which this is true.

Here it is appropriate to take as state the number  $i$  of customers in the system at the start of a period and as control the type of service provided. Then, the cost per period is  $r(i)$  plus  $c_f$  or  $c_s$  depending on whether fast or slow service is provided. We derive the transition probabilities of the system.

When the system is empty at the start of the period, the probability that the next state is  $j$  is independent of the type of service provided. It equals the given probability of  $j$  customer arrivals when  $j < n$ ,

$$p_{0j}(u_f) = p_{0j}(u_s) = p_j, \quad j = 0, 1, \dots, n-1,$$

and it equals the probability of  $n$  or more customer arrivals when  $j = n$ ,

$$p_{0n}(u_f) = p_{0n}(u_s) = \sum_{m=n}^{\infty} p_m.$$

When there is at least one customer in the system ( $i > 0$ ), we have

$$p_{ij}(u_f) = 0, \quad \text{if } j < i-1,$$

$$p_{ij}(u_f) = q_f p_0, \quad \text{if } j = i-1,$$

$$\begin{aligned} p_{ij}(u_f) &= P\{j-i+1 \text{ arrivals, service completed}\} \\ &\quad + P\{j-i \text{ arrivals, service not completed}\} \\ &= q_f p_{j-i+1} + (1-q_f) p_{j-i}, \quad \text{if } i-1 < j < n-1, \end{aligned}$$

$$p_{i(n-1)}(u_f) = q_f \sum_{m=n-i}^{\infty} p_m + (1-q_f) p_{n-1-i},$$

$$p_{in}(u_f) = (1-q_f) \sum_{m=n-i}^{\infty} p_m.$$

The transition probabilities when slow service is provided are also given by these formulas with  $u_f$  and  $q_f$  replaced by  $u_s$  and  $q_s$ , respectively.

### Example 1.1.5 (Optimizing a Chess Match Strategy)

A player is about to play a two-game chess match with an opponent, and wants to maximize his winning chances. Each game can have one of two outcomes:

- (a) A win by one of the players (1 point for the winner and 0 for the loser).
- (b) A draw (1/2 point for each of the two players).

If the score is tied at 1-1 at the end of the two games, the match goes into sudden-death mode, whereby the players continue to play until the first time one of them wins a game (and the match). The player has two playing styles and he can choose one of the two at will in each game, independently of the style he chose in previous games.

- (1) *Timid play* with which he draws with probability  $p_d > 0$ , and he loses with probability  $(1-p_d)$ .
- (2) *Bold play* with which he wins with probability  $p_w$ , and he loses with probability  $(1-p_w)$ .

Thus, in a given game, timid play never wins, while bold play never draws. The player wants to find a style selection strategy that maximizes his probability of winning the match. Note that once the match gets into sudden death, the player should play bold, since with timid play he can at best prolong the sudden death play, while running the risk of losing. Therefore, there are only two decisions for the player to make, the selection of the playing strategy in the first two games. Thus, we can model the problem as one with two stages, and with states the possible scores at the start of each of the first two stages (games), as shown in Fig. 1.1.4. The initial state is the initial score 0-0. The transition probabilities for each of the two different controls (playing styles) are also shown in Fig. 1.1.4. There is a cost at the terminal states: a cost of -1 at the winning scores 2-0 and 1.5-0.5, a cost of 0 at the losing scores 0-2 and 0.5-1.5, and a cost of  $-p_w$  at the tied score 1-1 (since the probability of winning in sudden death is  $p_w$ ). Note that to maximize the probability  $P$  of winning the match, we must minimize  $-P$ .

This problem has an interesting feature. One would think that if  $p_w < 1/2$ , the player would have a less than 50-50 chance of winning the match, even with optimal play, since his probability of losing is greater than his probability of winning any one game, regardless of his playing style. This is not so, however, because the player can adapt his playing style to the current score, but his opponent does not have that option. In other words, the player can use a closed-loop strategy, and it will be seen later that with optimal play, as determined by the dynamic programming algorithm, he has a better than

50-50 chance of winning the match provided  $p_d$  is higher than a threshold value  $\bar{p}$ , which, depending on the value of  $p_d$ , may satisfy  $\bar{p} < 1/2$ .

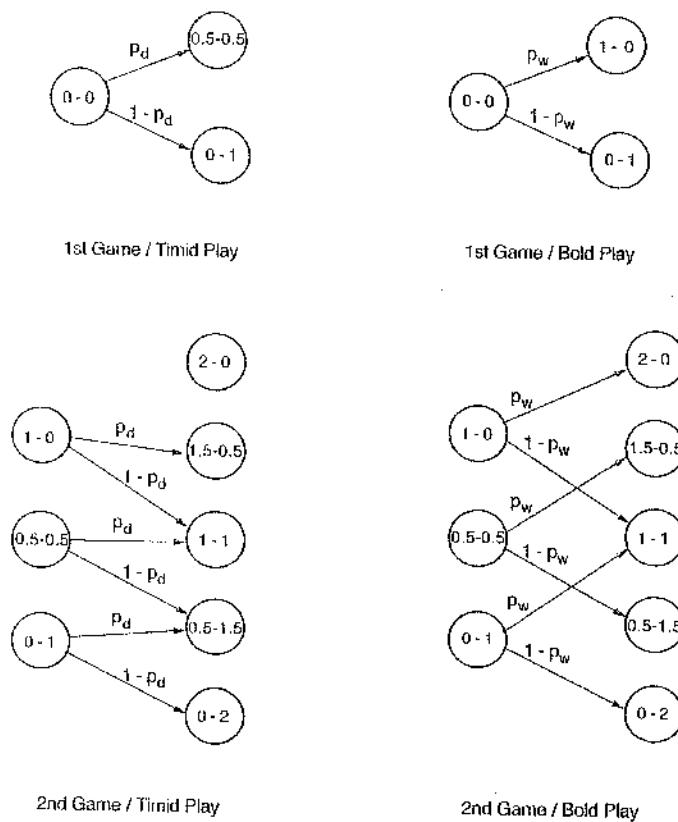


Figure 1.1.4 Chess match example. Transition probability graphs for each of the two possible controls (timid or bold play). Note here that the state space is not the same at each stage. The terminal cost is -1 at the winning final scores 2-0 and 1.5-0.5, 0 at the losing final scores 0-2 and 0.5-1.5, and  $-p_w$  at the tied score 1-1.

based on dynamic programming in the first six chapters, and we will extend our analysis to versions of this problem involving an infinite number of stages in the last chapter and in Vol. II of this work.

The basic problem is very general. In particular, we will not require that the state, control, or random parameter take a finite number of values or belong to a space of  $n$ -dimensional vectors. A surprising aspect of dynamic programming is that its applicability depends very little on the nature of the state, control, and random parameter spaces. For this reason it is convenient to proceed without any assumptions on the structure of these spaces; indeed such assumptions would become a serious impediment later.

### Basic Problem

We are given a discrete-time dynamic system

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1,$$

where the state  $x_k$  is an element of a space  $S_k$ , the control  $u_k$  is an element of a space  $C_k$ , and the random "disturbance"  $w_k$  is an element of a space  $D_k$ .

The control  $u_k$  is constrained to take values in a given nonempty subset  $U_k(x_k) \subset C_k$ , which depends on the current state  $x_k$ ; that is,  $u_k \in U_k(x_k)$  for all  $x_k \in S_k$  and  $k$ .

The random disturbance  $w_k$  is characterized by a probability distribution  $P_k(\cdot | x_k, u_k)$  that may depend explicitly on  $x_k$  and  $u_k$  but not on values of prior disturbances  $w_{k-1}, \dots, w_0$ .

We consider the class of policies (also called control laws) that consist of a sequence of functions

$$\pi = \{\mu_0, \dots, \mu_{N-1}\},$$

where  $\mu_k$  maps states  $x_k$  into controls  $u_k = \mu_k(x_k)$  and is such that  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k \in S_k$ . Such policies will be called *admissible*.

Given an initial state  $x_0$  and an admissible policy  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$ , the states  $x_k$  and disturbances  $w_k$  are random variables with distributions defined through the system equation

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \quad k = 0, 1, \dots, N-1. \quad (1.1)$$

Thus, for given functions  $g_k$ ,  $k = 0, 1, \dots, N$ , the expected cost of  $\pi$  starting at  $x_0$  is

$$J_\pi(x_0) = E \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

## 1.2 THE BASIC PROBLEM

We now formulate a general problem of decision under stochastic uncertainty over a finite number of stages. This problem, which we call *basic*, is central in this book. We will discuss solution methods for this problem

where the expectation is taken over the random variables  $w_k$  and  $x_k$ . An optimal policy  $\pi^*$  is one that minimizes this cost; that is,

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_{\pi}(x_0),$$

where  $\Pi$  is the set of all admissible policies.

Note that the optimal policy  $\pi^*$  is associated with a fixed initial state  $x_0$ . However, an interesting aspect of the basic problem and of dynamic programming is that it is typically possible to find a policy  $\pi^*$  that is simultaneously optimal for all initial states.

The optimal cost depends on  $x_0$  and is denoted by  $J^*(x_0)$ ; that is,

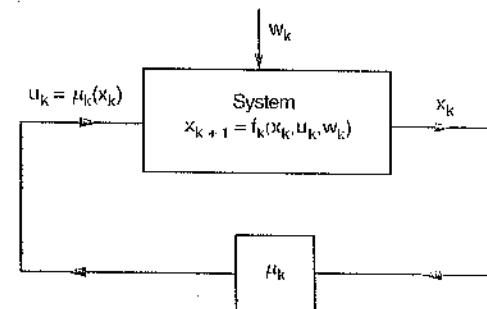
$$J^*(x_0) = \min_{\pi \in \Pi} J_{\pi}(x_0).$$

It is useful to view  $J^*$  as a function that assigns to each initial state  $x_0$  the optimal cost  $J^*(x_0)$  and call it the *optimal cost function* or *optimal value function*.†

### The Role and Value of Information

We noted earlier the distinction between open-loop minimization, where we select all controls  $u_0, \dots, u_{N-1}$  at once at time 0, and closed-loop minimization, where we select a policy  $\{\mu_0, \dots, \mu_{N-1}\}$  that applies the control  $\mu_k(x_k)$  at time  $k$  with knowledge of the current state  $x_k$  (see Fig. 1.2.1). With closed-loop policies, it is possible to achieve lower cost, essentially by taking advantage of the extra information (the value of the current state). The reduction in cost may be called the *value of the information* and can be significant indeed. If the information is not available, the controller cannot adapt appropriately to unexpected values of the state, and as a result the cost can be adversely affected. For example, in the inventory control example of the preceding section, the information that becomes available at the beginning of each period  $k$  is the inventory stock  $x_k$ . Clearly, this information is very important to the inventory manager, who will want to adjust the amount  $u_k$  to be purchased depending on whether the current stock  $x_k$  is running high or low.

† For the benefit of the mathematically oriented reader we note that in the preceding equation, "min" denotes the greatest lower bound (or infimum) of the set of numbers  $\{J_{\pi}(x_0) \mid \pi \in \Pi\}$ . A notation more in line with normal mathematical usage would be to write  $J^*(x_0) = \inf_{\pi \in \Pi} J_{\pi}(x_0)$ . However (as discussed in Appendix B), we find it convenient to use "min" in place of "inf" even when the infimum is not attained. It is less distracting, and it will not lead to any confusion.



**Figure 1.2.1** Information gathering in the basic problem. At each time  $k$  the controller observes the current state  $x_k$  and applies a control  $u_k = \mu_k(x_k)$  that depends on that state.

### Example 1.2.1

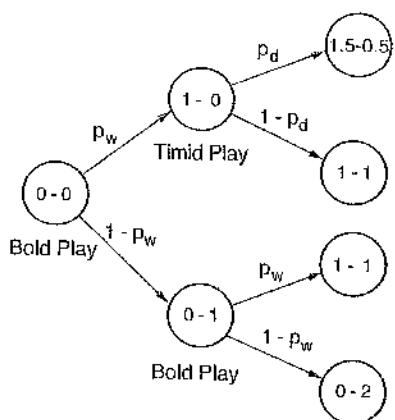
To illustrate the benefits of the proper use of information, let us consider the chess match example of the preceding section. There, a player can select timid play (probabilities  $p_d$  and  $1 - p_d$  for a draw and a loss, respectively) or bold play (probabilities  $p_w$  and  $1 - p_w$  for a win and a loss, respectively) in each of the two games of the match. Suppose the player chooses a policy of playing timid if and only if he is ahead in the score, as illustrated in Fig. 1.2.2; we will see in the next section that this policy is optimal, assuming  $p_d > p_w$ . Then after the first game (in which he plays bold), the score is 1-0 with probability  $p_w$  and 0-1 with probability  $1 - p_w$ . In the second game, he plays timid in the former case and bold in the latter case. Thus after two games, the probability of a match win is  $p_w p_d$ , the probability of a match loss is  $(1 - p_w)^2$ , and the probability of a tied score is  $p_w(1 - p_d) + (1 - p_w)p_d$ , in which case he has a probability  $p_w$  of winning the subsequent sudden death game. Thus the probability of winning the match with the given strategy is

$$p_w p_d + p_w(p_w(1 - p_d) + (1 - p_w)p_d),$$

which, with some rearrangement, gives

$$\text{Probability of a match win} = p_w^2(2 - p_w) + p_w(1 - p_w)p_d. \quad (1.2)$$

Suppose now that  $p_w < 1/2$ . Then the player has a greater probability of losing than winning any one game, regardless of the type of play he uses. From this we can infer that no open-loop strategy can give the player a greater than 50-50 chance of winning the match. Yet from Eq. (1.2) it can be seen that with the closed-loop strategy of playing timid if and only if the player is ahead in the score, the chance of a match win can be greater than 50-50, provided that  $p_w$  is close enough to  $1/2$  and  $p_d$  is close enough to 1. As an example, for  $p_w = 0.45$  and  $p_d = 0.9$ , Eq. (1.2) gives a match win probability of roughly 0.53.



**Figure 1.2.3** Illustration of the policy used in Example 1.2.1 to obtain a greater than 50-50 chance of winning the chess match and associated transition probabilities. The player chooses a policy of playing timid if and only if he is ahead in the score.

To calculate the value of information, let us consider the four open-loop policies, whereby we decide on the type of play to be used without waiting to see the result of the first game. These are:

- (1) Play timid in both games; this has a probability  $p_d^2 p_w$  of winning the match.
- (2) Play bold in both games; this has a probability  $p_w^2 + 2p_w(1 - p_w) = p_w^2(3 - 2p_w)$  of winning the match.
- (3) Play bold in the first game and timid in the second game; this has a probability  $p_w p_d + p_w^2(1 - p_d)$  of winning the match.
- (4) Play timid in the first game and bold in the second game; this also has a probability  $p_w p_d + p_w^2(1 - p_d)$  of winning the match.

The first policy is always dominated by the others, and the optimal open-loop probability of winning the match is

$$\begin{aligned} \text{Open-loop probability of win} &= \max(p_w^2(3 - 2p_w), p_w p_d + p_w^2(1 - p_d)) \\ &= p_w^2 + p_w(1 - p_w) \max(2p_w, p_d). \end{aligned} \quad (1.3)$$

Thus if  $p_d > 2p_w$ , we see that the optimal open-loop policy is to play timid in one of the two games and play bold in the other, and otherwise it is optimal to play bold in both games. For  $p_w = 0.45$  and  $p_d = 0.9$ , Eq. (1.3) gives an optimal open-loop match win probability of roughly 0.425. Thus, the value of the information (the outcome of the first game) is the difference of the optimal closed-loop and open-loop values, which is approximately  $0.53 - 0.425 = 0.105$ .

More generally, by subtracting Eqs. (1.2) and (1.3), we see that

$$\begin{aligned} \text{Value of information} &= p_w^2(2 - p_w) + p_w(1 - p_w)p_d \\ &- p_w^2 + p_w(1 - p_w) \max(2p_w, p_d) \\ &= p_w(1 - p_w) \min(p_w, p_d - p_w). \end{aligned}$$

It should be noted, however, that whereas availability of the state information cannot hurt, it may not result in an advantage either. For instance, in deterministic problems, where no random disturbances are present, one can predict the future states given the initial state and the sequence of controls. Thus, optimization over all sequences  $\{u_0, u_1, \dots, u_{N-1}\}$  of controls leads to the same optimal cost as optimization over all admissible policies. The same can be true even in some stochastic problems (see for example Exercise 1.13). This brings up a related issue. Assuming no information is forgotten, the controller actually knows the prior states and controls  $x_0, u_0, \dots, x_{k-1}, u_{k-1}$  as well as the current state  $x_k$ . Therefore, the question arises whether policies that use the entire system history can be superior to policies that use just the current state. The answer turns out to be negative although the proof is technically complicated (see [BeS78]). The intuitive reason is that, for a given time  $k$  and state  $x_k$ , all future expected costs depend explicitly just on  $x_k$  and not on prior history.

### Encoding Risk in the Cost Function

As mentioned above, an important characteristic of stochastic problems is the possibility of using information with advantage. Another distinguishing characteristic is the need to take into account *risk* in the problem formulation. For example, in a typical investment problem one is not only interested in the expected profit of the investment decision, but also in its variance: given a choice between two investments with nearly equal expected profit and markedly different variance, most investors would prefer the investment with smaller variance. This indicates that expected value of cost or reward need not be the most appropriate yardstick for expressing a decision maker's preference between decisions.

As a more dramatic example of the need to take risk into account when formulating optimization problems under uncertainty, consider the so-called St. Petersburg paradox. Here, a person is offered the opportunity of paying  $x$  dollars in exchange for participation in the following game: a fair coin is flipped sequentially and the person is paid  $2^k$  dollars, where  $k$  is the number of times heads have come up before tails come up for the first time. The decision that the person must make is whether to accept or reject participation in the game. Now if he accepts, his expected profit

from the game is

$$\sum_{k=0}^{\infty} \frac{1}{2^{k+1}} \cdot 2^k - x = \infty,$$

so if his acceptance criterion is based on maximization of expected profit, he is willing to pay any amount  $x$  to enter the game. This, however, is in strong disagreement with observed behavior, due to the risk element involved in entering the game, and shows that a different formulation of the problem is needed. The formulation of problems of decision under uncertainty so that risk is properly taken into account is a deep subject with an interesting theory. An introduction to this theory is given in Appendix G. It is shown in particular that minimization of expected cost is appropriate under reasonable assumptions, provided the cost function is suitably chosen so that it properly encodes the risk preferences of the decision maker.

### 1.3 THE DYNAMIC PROGRAMMING ALGORITHM

The dynamic programming (DP) technique rests on a very simple idea, the *principle of optimality*. The name is due to Bellman, who contributed a great deal to the popularization of DP and to its transformation into a systematic tool. Roughly, the principle of optimality states the following rather obvious fact.

#### Principle of Optimality

Let  $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$  be an optimal policy for the basic problem, and assume that when using  $\pi^*$ , a given state  $x_i$  occurs at time  $i$  with positive probability. Consider the subproblem whereby we are at  $x_i$  at time  $i$  and wish to minimize the "cost-to-go" from time  $i$  to time  $N$

$$E \left\{ g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}.$$

Then the truncated policy  $\{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{N-1}^*\}$  is optimal for this subproblem.

The intuitive justification of the principle of optimality is very simple. If the truncated policy  $\{\mu_i^*, \mu_{i+1}^*, \dots, \mu_{N-1}^*\}$  were not optimal as stated, we would be able to reduce the cost further by switching to an optimal policy for the subproblem once we reach  $x_i$ . For an auto travel analogy, suppose that the fastest route from Los Angeles to Boston passes through Chicago. The principle of optimality translates to the obvious fact that the Chicago to Boston portion of the route is also the fastest route for a trip that starts from Chicago and ends in Boston.

The principle of optimality suggests that an optimal policy can be constructed in piecemeal fashion, first constructing an optimal policy for the "tail subproblem" involving the last stage, then extending the optimal policy to the "tail subproblem" involving the last two stages, and continuing in this manner until an optimal policy for the entire problem is constructed. The DP algorithm is based on this idea: it proceeds sequentially, by solving all the tail subproblems of a given time length, using the solution of the tail subproblems of shorter time length. We introduce the algorithm with two examples, one deterministic and one stochastic.

#### The DP Algorithm for a Deterministic Scheduling Example

Let us consider the scheduling example of the preceding section, and let us apply the principle of optimality to calculate the optimal schedule. We have to schedule optimally the four operations A, B, C, and D. The transition and setup costs are shown in Fig. 1.3.1 next to the corresponding arcs.

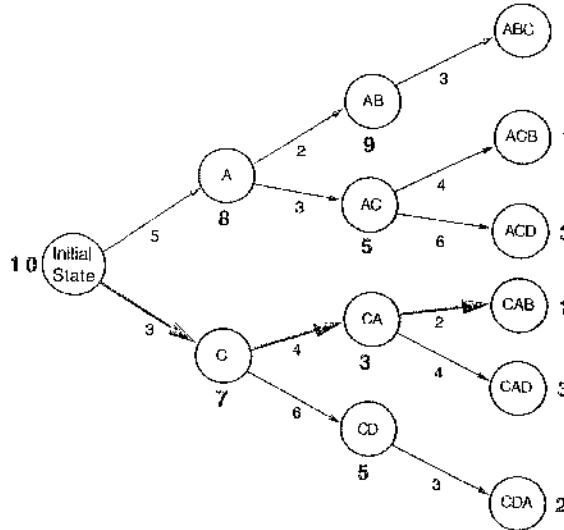
According to the principle of optimality, the "tail" portion of an optimal schedule must be optimal. For example, suppose that the optimal schedule is CABD. Then, having scheduled first C and then A, it must be optimal to complete the schedule with BD rather than with DB. With this in mind, we solve all possible tail subproblems of length two, then all tail subproblems of length three, and finally the original problem that has length four (the subproblems of length one are of course trivial because there is only one operation that is as yet unscheduled). As we will see shortly, the tail subproblems of length  $k+1$  are easily solved once we have solved the tail subproblems of length  $k$ , and this is the essence of the DP technique.

*Tail Subproblems of Length 2:* These subproblems are the ones that involve two unscheduled operations and correspond to the states AB, AC, CA, and CD (see Fig. 1.3.1)

*State AB:* Here it is only possible to schedule operation C as the next operation, so the optimal cost of this subproblem is 9 (the cost of scheduling C after B, which is 3, plus the cost of scheduling D after C, which is 6).

*State AC:* Here the possibilities are to (a) schedule operation B and then D, which has cost 5, or (b) schedule operation D and then B, which has cost 9. The first possibility is optimal, and the corresponding cost of the tail subproblem is 5, as shown next to node AC in Fig. 1.3.1.

*State CA:* Here the possibilities are to (a) schedule operation B and then D, which has cost 3, or (b) schedule operation D and then B, which has cost 7. The first possibility is optimal, and the correspond-



**Figure 1.3.1** Transition graph of the deterministic scheduling problem, with the cost of each decision shown next to the corresponding arc. Next to each node/state we show the cost to optimally complete the schedule starting from that state. This is the optimal cost of the corresponding tail subproblem (cf. the principle of optimality). The optimal cost for the original problem is equal to 10, as shown next to the initial state. The optimal schedule corresponds to the thick-line arcs.

ing cost of the tail subproblem is 3, as shown next to node CA in Fig. 1.3.1.

**State CD:** Here it is only possible to schedule operation A as the next operation, so the optimal cost of this subproblem is 5.

**Tail Subproblems of Length 3:** These subproblems can now be solved using the optimal costs of the subproblems of length 2.

**State A:** Here the possibilities are to (a) schedule next operation B (cost 2) and then solve optimally the corresponding subproblem of length 2 (cost 9, as computed earlier), a total cost of 11, or (b) schedule next operation C (cost 3) and then solve optimally the corresponding subproblem of length 2 (cost 5, as computed earlier), a total cost of 8. The second possibility is optimal, and the corresponding cost of the tail subproblem is 8, as shown next to node A in Fig. 1.3.1.

**State C:** Here the possibilities are to (a) schedule next operation A (cost 4) and then solve optimally the corresponding subproblem of length 2 (cost 3, as computed earlier), a total cost of 7, or (b) schedule next operation D (cost 6) and then solve optimally the corresponding

subproblem of length 2 (cost 5, as computed earlier), a total cost of 11. The first possibility is optimal, and the corresponding cost of the tail subproblem is 7, as shown next to node A in Fig. 1.3.1.

**Original Problem of Length 4:** The possibilities here are (a) start with operation A (cost 5) and then solve optimally the corresponding subproblem of length 3 (cost 8, as computed earlier), a total cost of 13, or (b) start with operation C (cost 3) and then solve optimally the corresponding subproblem of length 3 (cost 7, as computed earlier), a total cost of 10. The second possibility is optimal, and the corresponding optimal cost is 10, as shown next to the initial state node in Fig. 1.3.1.

Note that having computed the optimal cost of the original problem through the solution of all the tail subproblems, we can construct the optimal schedule by starting at the initial node and proceeding forward, each time choosing the operation that starts the optimal schedule for the corresponding tail subproblem. In this way, by inspection of the graph and the computational results of Fig. 1.3.1, we determine that CABD is the optimal schedule.

### The DP Algorithm for the Inventory Control Example

Consider the inventory control example of the previous section. Similar to the solution of the preceding deterministic scheduling problem, we calculate sequentially the optimal costs of all the tail subproblems, going from shorter to longer problems. The only difference is that the optimal costs are computed as expected values, since the problem here is stochastic.

**Tail Subproblems of Length 1:** Assume that at the beginning of period  $N - 1$  the stock is  $x_{N-1}$ . Clearly, no matter what happened in the past, the inventory manager should order the amount of inventory that minimizes over  $u_{N-1} \geq 0$  the sum of the ordering cost and the expected terminal holding/shortage cost. Thus, he should minimize over  $u_{N-1}$  the sum  $cu_{N-1} + E\{R(x_N)\}$ , which can be written as

$$cu_{N-1} + \min_{u_{N-1}} E\{R(x_{N-1} + u_{N-1} - w_{N-1})\}.$$

Adding the holding/shortage cost of period  $N - 1$ , we see that the optimal cost for the last period (plus the terminal cost) is given by

$$J_{N-1}(x_{N-1}) = r(x_{N-1})$$

$$+ \min_{u_{N-1} \geq 0} \left[ cu_{N-1} + E\{R(x_{N-1} + u_{N-1} - w_{N-1})\} \right].$$

Naturally,  $J_{N-1}$  is a function of the stock  $x_{N-1}$ . It is calculated either analytically or numerically (in which case a table is used for computer

storage of the function  $J_{N-1}$ ). In the process of calculating  $J_{N-1}$ , we obtain the optimal inventory policy  $\mu_{N-1}^*(x_{N-1})$  for the last period:  $\mu_{N-1}^*(x_{N-1})$  is the value of  $u_{N-1}$  that minimizes the right-hand side of the preceding equation for a given value of  $x_{N-1}$ .

*Tail Subproblems of Length 2:* Assume that at the beginning of period  $N - 2$  the stock is  $x_{N-2}$ . It is clear that the inventory manager should order the amount of inventory that minimizes not just the expected cost of period  $N - 2$  but rather the

$$\begin{aligned} & (\text{expected cost of period } N - 2) + (\text{expected cost of period } N - 1, \\ & \quad \text{given that an optimal policy will be used at period } N - 1), \end{aligned}$$

which is equal to

$$r(x_{N-2}) + cu_{N-2} + E\{J_{N-1}(x_{N-1})\}.$$

Using the system equation  $x_{N-1} = x_{N-2} + u_{N-2} - w_{N-2}$ , the last term is also written as  $J_{N-1}(x_{N-2} + u_{N-2} - w_{N-2})$ .

Thus the optimal cost for the last two periods given that we are at state  $x_{N-2}$ , denoted  $J_{N-2}(x_{N-2})$ , is given by

$$\begin{aligned} J_{N-2}(x_{N-2}) &= r(x_{N-2}) \\ &+ \min_{u_{N-2} \geq 0} \left[ cu_{N-2} + E_{w_{N-2}} \{ J_{N-1}(x_{N-2} + u_{N-2} - w_{N-2}) \} \right] \end{aligned}$$

Again  $J_{N-2}(x_{N-2})$  is calculated for every  $x_{N-2}$ . At the same time, the optimal policy  $\mu_{N-2}^*(x_{N-2})$  is also computed.

*Tail Subproblems of Length  $N - k$ :* Similarly, we have that at period  $k$ , when the stock is  $x_k$ , the inventory manager should order  $u_k$  to minimize

$$\begin{aligned} & (\text{expected cost of period } k) + (\text{expected cost of periods } k + 1, \dots, N - 1, \\ & \quad \text{given that an optimal policy will be used for these periods}). \end{aligned}$$

By denoting by  $J_k(x_k)$  the optimal cost, we have

$$J_k(x_k) = r(x_k) + \min_{u_k \geq 0} \left[ cu_k + E_{w_k} \{ J_{k+1}(x_k + u_k - w_k) \} \right], \quad (1.4)$$

which is actually the dynamic programming equation for this problem.

The functions  $J_k(x_k)$  denote the optimal expected cost for the tail subproblem that starts at period  $k$  with initial inventory  $x_k$ . These functions are computed recursively backward in time, starting at period  $N - 1$  and ending at period 0. The value  $J_0(x_0)$  is the optimal expected cost when the initial stock at time 0 is  $x_0$ . During the calculations, the optimal

policy is simultaneously computed from the minimization in the right-hand side of Eq. (1.4).

The example illustrates the main advantage offered by DP. While the original inventory problem requires an optimization over the set of policies, the DP algorithm of Eq. (1.4) decomposes this problem into a sequence of minimizations carried out over the set of controls. Each of these minimizations is much simpler than the original problem.

### The DP Algorithm

We now state the DP algorithm for the basic problem and show its optimality by translating into mathematical terms the heuristic argument given above for the inventory example.

**Proposition 1.3.1:** For every initial state  $x_0$ , the optimal cost  $J^*(x_0)$  of the basic problem is equal to  $J_0(x_0)$ , given by the last step of the following algorithm, which proceeds backward in time from period  $N - 1$  to period 0:

$$J_N(x_N) = g_N(x_N), \quad (1.5)$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}, \quad k = 0, 1, \dots, N - 1, \quad (1.6)$$

where the expectation is taken with respect to the probability distribution of  $w_k$ , which depends on  $x_k$  and  $u_k$ . Furthermore, if  $u_k^* = \mu_k^*(x_k)$  minimizes the right side of Eq. (1.6) for each  $x_k$  and  $k$ , the policy  $\pi^* = \{\mu_0^*, \dots, \mu_{N-1}^*\}$  is optimal.

**Proof:** † For any admissible policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$  and each  $k = 0, 1, \dots, N - 1$ , denote  $\pi^k = \{\mu_k, \mu_{k+1}, \dots, \mu_{N-1}\}$ . For  $k = 0, 1, \dots, N - 1$ , let  $J_k^*(x_k)$  be the optimal cost for the  $(N - k)$ -stage problem that starts at state  $x_k$  and time  $k$ , and ends at time  $N$ ,

$$J_k^*(x_k) = \min_{\pi^k} E_{w_k, \dots, w_{N-1}} \left\{ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\}.$$

† Our proof is somewhat informal and assumes that the functions  $J_k$  are well-defined and finite. For a strictly rigorous proof, some technical mathematical issues must be addressed; see Section 1.5. These issues do not arise if the disturbance  $w_k$  takes a finite or countable number of values and the expected values of all terms in the expression of the cost function (1.1) are well-defined and finite for every admissible policy  $\pi$ .

For  $k = N$ , we define  $J_N^*(x_N) = g_N(x_N)$ . We will show by induction that the functions  $J_k^*$  are equal to the functions  $J_k$  generated by the DP algorithm, so that for  $k = 0$ , we will obtain the desired result.

Indeed, we have by definition  $J_N^* = J_N = g_N$ . Assume that for some  $k$  and all  $x_{k+1}$ , we have  $J_{k+1}^*(x_{k+1}) = J_{k+1}(x_{k+1})$ . Then, since  $\pi^k := (\mu_k, \pi^{k+1})$ , we have for all  $x_k$

$$\begin{aligned} J_k^*(x_k) &= \min_{(\mu_k, \pi^{k+1})} E_{w_k, \dots, w_{N-1}} \left\{ g_k(x_k, \mu_k(x_k), w_k) \right. \\ &\quad \left. + g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\} \\ &= \min_{\mu_k} E_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) \right. \\ &\quad \left. + \min_{\pi^{k+1}} \left[ E_{w_{k+1}, \dots, w_{N-1}} \left\{ g_N(x_N) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\} \right] \right\} \\ &= \min_{\mu_k} E_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}^*(f_k(x_k, \mu_k(x_k), w_k)) \right\} \\ &= \min_{\mu_k} E_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right\} \\ &= \min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} \\ &= J_k(x_k), \end{aligned}$$

completing the induction. In the second equation above, we moved the minimum over  $\pi^{k+1}$  inside the braced expression, using a principle of optimality argument: “the tail portion of an optimal policy is optimal for the tail subproblem” (a more rigorous justification of this step is given in Section 1.5). In the third equation, we used the definition of  $J_{k+1}^*$ , and in the fourth equation we used the induction hypothesis. In the fifth equation, we converted the minimization over  $\mu_k$  to a minimization over  $u_k$ , using the fact that for any function  $F$  of  $x$  and  $u$ , we have

$$\min_{\mu \in M} F(x, \mu(x)) = \min_{u \in U(x)} F(x, u),$$

where  $M$  is the set of all functions  $\mu(x)$  such that  $\mu(x) \in U(x)$  for all  $x$ . Q.E.D.

The argument of the preceding proof provides an interpretation of  $J_k(x_k)$  as the optimal cost for an  $(N - k)$ -stage problem starting at state  $x_k$  and time  $k$ , and ending at time  $N$ . We consequently call  $J_k(x_k)$  the *cost-to-go* at state  $x_k$  and time  $k$ , and refer to  $J_k$  as the *cost-to-go function* at time  $k$ .

Ideally, we would like to use the DP algorithm to obtain closed-form expressions for  $J_k$  or an optimal policy. In this book, we will discuss a large number of models that admit analytical solution by DP. Even if such models rely on oversimplified assumptions, they are often very useful. They may provide valuable insights about the structure of the optimal solution of more complex models, and they may form the basis for suboptimal control schemes. Furthermore, the broad collection of analytically solvable models provides helpful guidelines for modeling: when faced with a new problem it is worth trying to pattern its model after one of the principal analytically tractable models.

Unfortunately, in many practical cases an analytical solution is not possible, and one has to resort to numerical execution of the DP algorithm. This may be quite time-consuming since the minimization in the DP Eq. (1.6) must be carried out for each value of  $x_k$ . The state space must be discretized in some way if it is not already a finite set. The computational requirements are proportional to the number of possible values of  $x_k$ , so for complex problems the computational burden may be excessive. Nonetheless, DP is the only general approach for sequential optimization under uncertainty, and even when it is computationally prohibitive, it can serve as the basis for more practical suboptimal approaches, which will be discussed in Chapter 6.

The following examples illustrate some of the analytical and computational aspects of DP.

### Example 1.3.1

A certain material is passed through a sequence of two ovens (see Fig. 1.3.2). Denote

$x_0$ : initial temperature of the material,

$x_k$ ,  $k = 1, 2$ : temperature of the material at the exit of oven  $k$ ,

$u_{k-1}$ ,  $k = 1, 2$ : prevailing temperature in oven  $k$ .

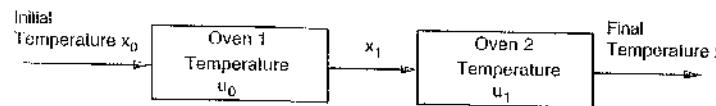
We assume a model of the form

$$x_{k+1} = (1 - a)x_k + au_k, \quad k = 0, 1,$$

where  $a$  is a known scalar from the interval  $(0, 1)$ . The objective is to get the final temperature  $x_2$  close to a given target  $T$ , while expending relatively little energy. This is expressed by a cost function of the form

$$r(x_2 - T)^2 + u_0^2 + u_1^2,$$

where  $r > 0$  is a given scalar. We assume no constraints on  $u_k$ . (In reality, there are constraints, but if we can solve the unconstrained problem and verify that the solution satisfies the constraints, everything will be fine.) The problem is deterministic; that is, there is no stochastic uncertainty. However,



**Figure 1.3.2** Problem of Example 1.3.1. The temperature of the material evolves according to  $x_{k+1} = (1 - a)x_k + au_k$ , where  $a$  is some scalar with  $0 < a < 1$ .

such problems can be placed within the basic framework by introducing a fictitious disturbance taking a unique value with probability one.

We have  $N = 2$  and a terminal cost  $g_2(x_2) = r(x_2 - T)^2$ , so the initial condition for the DP algorithm is [cf. Eq. (1.5)]

$$J_2(x_2) = r(x_2 - T)^2.$$

For the next-to-last stage, we have [cf. Eq. (1.6)]

$$\begin{aligned} J_1(x_1) &= \min_{u_1} [u_1^2 + J_2(x_2)] \\ &= \min_{u_1} [u_1^2 + J_2((1 - a)x_1 + au_1 - T)^2]. \end{aligned}$$

Substituting the previous form of  $J_2$ , we obtain

$$J_1(x_1) = \min_{u_1} \left[ u_1^2 + r((1 - a)x_1 + au_1 - T)^2 \right]. \quad (1.7)$$

This minimization will be done by setting to zero the derivative with respect to  $u_1$ . This yields

$$0 = 2u_1 + 2ra((1 - a)x_1 + au_1 - T),$$

and by collecting terms and solving for  $u_1$ , we obtain the optimal temperature for the last oven:

$$\mu_1^*(x_1) = \frac{ra(T - (1 - a)x_1)}{1 + ra^2}.$$

Note that this is not a single control but rather a control function, a rule that tells us the optimal oven temperature  $u_1 = \mu_1^*(x_1)$  for each possible state  $x_1$ .

By substituting the optimal  $u_1$  in the expression (1.7) for  $J_1$ , we obtain

$$\begin{aligned} J_1(x_1) &= \frac{r^2 a^2 ((1 - a)x_1 - T)^2}{(1 + ra^2)^2} + r \left( (1 - a)x_1 + \frac{ra^2(T - (1 - a)x_1)}{1 + ra^2} - T \right)^2 \\ &= \frac{r^2 a^2 ((1 - a)x_1 - T)^2}{(1 + ra^2)^2} + r \left( \frac{ra^2}{1 + ra^2} - 1 \right)^2 ((1 - a)x_1 - T)^2 \\ &= \frac{r((1 - a)x_1 - T)^2}{1 + ra^2}. \end{aligned}$$

### Sec. 1.3 The Dynamic Programming Algorithm

We now go back one stage. We have [cf. Eq. (1.6)]

$$J_0(x_0) = \min_{u_0} [u_0^2 + J_1(x_1)] = \min_{u_0} [u_0^2 + J_1((1 - a)x_0 + au_0 - T)],$$

and by substituting the expression already obtained for  $J_1$ , we have

$$J_0(x_0) = \min_{u_0} \left[ u_0^2 + \frac{r((1 - a)^2 x_0 + (1 - a)a u_0 - T)^2}{1 + ra^2} \right].$$

We minimize with respect to  $u_0$  by setting the corresponding derivative to zero. We obtain

$$0 = 2u_0 + \frac{2r(1 - a)a((1 - a)^2 x_0 + (1 - a)a u_0 - T)}{1 + ra^2}.$$

This yields, after some calculation, the optimal temperature of the first oven:

$$\mu_0^*(x_0) = \frac{r(1 - a)a(T - (1 - a)^2 x_0)}{1 + ra^2(1 + (1 - a)^2)}.$$

The optimal cost is obtained by substituting this expression in the formula for  $J_0$ . This leads to a straightforward but lengthy calculation, which in the end yields the rather simple formula

$$J_0(x_0) = \frac{r((1 - a)^2 x_0 - T)^2}{1 + ra^2(1 + (1 - a)^2)}.$$

This completes the solution of the problem.

One noteworthy feature in the preceding example is the facility with which we obtained an analytical solution. A little thought while tracing the steps of the algorithm will convince the reader that what simplifies the solution is the quadratic nature of the cost and the linearity of the system equation. In Section 4.1 we will see that, generally, when the system is linear and the cost is quadratic, the optimal policy and cost-to-go function are given by closed-form expressions, regardless of the number of stages  $N$ .

Another noteworthy feature of the example is that the optimal policy remains unaffected when a zero-mean stochastic disturbance is added in the system equation. To see this, assume that the material's temperature evolves according to

$$x_{k+1} = (1 - a)x_k + au_k + w_k, \quad k = 0, 1,$$

where  $w_0, w_1$  are independent random variables with given distribution, zero mean

$$E\{w_0\} = E\{w_1\} = 0,$$

and finite variance. Then the equation for  $J_1$  [cf. Eq. (1.6)] becomes

$$\begin{aligned} J_1(x_1) &= \min_{u_1} E \left\{ u_1^2 + r((1-a)x_1 + au_1 + w_1 - T)^2 \right\} \\ &= \min_{u_1} \left[ u_1^2 + r((1-a)x_1 + au_1 - T)^2 \right. \\ &\quad \left. + 2rE\{w_1\}((1-a)x_1 + au_1 - T) + rE\{w_1^2\} \right]. \end{aligned}$$

Since  $E\{w_1\} = 0$ , we obtain

$$J_1(x_1) = \min_{u_1} \left[ u_1^2 + r((1-a)x_1 + au_1 - T)^2 \right] + rE\{w_1^2\}.$$

Comparing this equation with Eq. (1.7), we see that the presence of  $w_1$  has resulted in an additional inconsequential term,  $rE\{w_1^2\}$ . Therefore, the optimal policy for the last stage remains unaffected by the presence of  $w_1$ , while  $J_1(x_1)$  is increased by the constant term  $rE\{w_1^2\}$ . It can be seen that a similar situation also holds for the first stage. In particular, the optimal cost is given by the same expression as before except for an additive constant that depends on  $E\{w_0^2\}$  and  $E\{w_1^2\}$ .

If the optimal policy is unaffected when the disturbances are replaced by their means, we say that *certainty equivalence* holds. We will derive certainty equivalence results for several types of problems involving a linear system and a quadratic cost (see Sections 4.1, 5.2, and 5.3).

### Example 1.3.2

To illustrate the computational aspects of DP, consider an inventory control problem that is slightly different from the one of Sections 1.1 and 1.2. In particular, we assume that inventory  $u_k$  and the demand  $w_k$  are nonnegative integers, and that the excess demand ( $w_k - x_k - u_k$ ) is lost. As a result, the stock equation takes the form

$$x_{k+1} = \max(0, x_k + u_k - w_k).$$

We also assume that there is an upper bound of 2 units on the stock that can be stored, i.e. there is a constraint  $x_k + u_k \leq 2$ . The holding/storage cost for the  $k$ th period is given by

$$(x_k + u_k - w_k)^2,$$

implying a penalty both for excess inventory and for unmet demand at the end of the  $k$ th period. The ordering cost is 1 per unit stock ordered. Thus the cost per period is

$$g_k(x_k, u_k, w_k) = u_k + (x_k + u_k - w_k)^2.$$

The terminal cost is assumed to be 0,

$$g_N(x_N) = 0.$$

The planning horizon  $N$  is 3 periods, and the initial stock  $x_0$  is 0. The demand  $w_k$  has the same probability distribution for all periods, given by

$$p(w_k = 0) = 0.4, \quad p(w_k = 1) = 0.7, \quad p(w_k = 2) = 0.2.$$

The system can also be represented in terms of the transition probabilities  $p_{ij}(u)$  between the three possible states, for the different values of the control (see Fig. 1.3.3).

The starting equation for the DP algorithm is

$$J_3(x_3) = 0,$$

since the terminal state cost is 0 [cf. Eq. (1.5)]. The algorithm takes the form [cf. Eq. (1.6)]

$$J_k(x_k) = \min_{\substack{0 \leq u_k \leq 2-x_k \\ u_k = 0, 1, 2}} E \left\{ u_k + (x_k + u_k - w_k)^2 + J_{k+1}(\max(0, x_k + u_k - w_k)) \right\},$$

where  $k = 0, 1, 2$ , and  $x_k, u_k, w_k$  can take the values 0, 1, and 2.

**Period 2:** We compute  $J_2(x_2)$  for each of the three possible states. We have

$$\begin{aligned} J_2(0) &= \min_{u_2=0,1,2} E \left\{ u_2 + (u_2 - w_2)^2 \right\} \\ &= \min_{u_2=0,1,2} [u_2 + 0.1(u_2)^2 + 0.7(u_2 - 1)^2 + 0.2(u_2 - 2)^2]. \end{aligned}$$

We calculate the expectation of the right side for each of the three possible values of  $u_2$ :

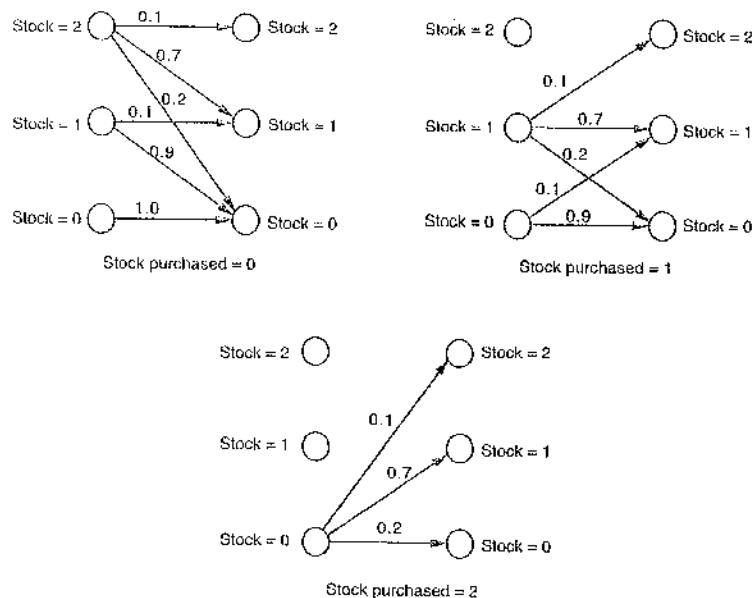
$$\begin{aligned} u_2 = 0 : E\{\cdot\} &= 0.7 \cdot 1 + 0.2 \cdot 4 = 1.5, \\ u_2 = 1 : E\{\cdot\} &= 1 + 0.1 \cdot 1 + 0.2 \cdot 1 = 1.3, \\ u_2 = 2 : E\{\cdot\} &= 2 + 0.1 \cdot 4 + 0.7 \cdot 1 = 3.1. \end{aligned}$$

Hence we have, by selecting the minimizing  $u_2$ ,

$$J_2(0) = 1.3, \quad u_2^*(0) = 1.$$

For  $x_2 = 1$ , we have

$$\begin{aligned} J_2(1) &= \min_{u_2=0,1} E \left\{ u_2 + (1 + u_2 - w_2)^2 \right\} \\ &= \min_{u_2=0,1} [u_2 + 0.1(1 + u_2)^2 + 0.7(u_2)^2 + 0.2(u_2 - 1)^2]. \end{aligned}$$



Stock	Stage 0 Cost-to-go	Stage 0 Optimal stock to purchase	Stage 1 Cost-to-go	Stage 1 Optimal stock to purchase	Stage 2 Cost-to-go	Stage 2 Optimal stock to purchase
0	3.7	1	2.5	1	1.3	1
1	2.7	0	1.5	0	0.3	0
2	2.818	0	1.68	0	1.1	0

Figure 1.3.3 System and DP results for Example 1.3.2. The transition probability diagrams for the different values of stock purchased (control) are shown. The numbers next to the arcs are the transition probabilities. The control  $u = 1$  is not available at state 2 because of the limitation  $x_k + u_k \leq 2$ . Similarly, the control  $u = 2$  is not available at states 1 and 2. The results of the DP algorithm are given in the table.

The expected value in the right side is

$$u_2 = 0 : E\{\cdot\} = 0.1 \cdot 1 + 0.2 \cdot 1 = 0.3,$$

$$u_2 = 1 : E\{\cdot\} = 1 + 0.1 \cdot 4 + 0.7 \cdot 1 = 2.1.$$

Hence

$$J_2(1) = 0.3, \quad \mu_2^*(1) = 0.$$

For  $x_2 = 2$ , the only admissible control is  $u_2 = 0$ , so we have

$$J_2(2) = E_{w_2} \{ (2 - w_2)^2 \} = 0.1 \cdot 4 + 0.7 \cdot 1 = 1.1,$$

$$J_2(2) = 1.1, \quad \mu_2^*(2) = 0.$$

**Period 1:** Again we compute  $J_1(x_1)$  for each of the three possible states  $x_1 = 0, 1, 2$ , using the values  $J_2(0)$ ,  $J_2(1)$ ,  $J_2(2)$  obtained in the previous period. For  $x_1 = 0$ , we have

$$J_1(0) = \min_{u_1=0,1,2} E_{w_1} \left\{ u_1 + (u_1 - w_1)^2 + J_2(\max(0, u_1 - w_1)) \right\},$$

$$u_1 = 0 : E\{\cdot\} = 0.1 \cdot J_2(0) + 0.7(1 + J_2(0)) + 0.2(4 + J_2(0)) = 2.8,$$

$$u_1 = 1 : E\{\cdot\} = 1 + 0.1(1 + J_2(1)) + 0.7 \cdot J_2(0) + 0.2(1 + J_2(0)) = 2.5,$$

$$u_1 = 2 : E\{\cdot\} = 2 + 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0) = 3.68,$$

$$J_1(0) = 2.5, \quad \mu_1^*(0) = 1.$$

For  $x_1 = 1$ , we have

$$J_1(1) = \min_{u_1=0,1} E_{w_1} \left\{ u_1 + (1 + u_1 - w_1)^2 + J_2(\max(0, 1 + u_1 - w_1)) \right\},$$

$$u_1 = 0 : E\{\cdot\} = 0.1(1 + J_2(1)) + 0.7 \cdot J_2(0) + 0.2(1 + J_2(0)) = 1.5,$$

$$u_1 = 1 : E\{\cdot\} = 1 + 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0) = 2.68,$$

$$J_1(1) = 1.5, \quad \mu_1^*(1) = 0.$$

For  $x_1 = 2$ , the only admissible control is  $u_1 = 0$ , so we have

$$J_1(2) = E_{w_1} \left\{ (2 - w_1)^2 + J_2(\max(0, 2 - w_1)) \right\}$$

$$= 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0)$$

$$= 1.68,$$

$$J_1(2) = 1.68, \quad \mu_1^*(2) = 0.$$

**Period 0:** Here we need to compute only  $J_0(0)$  since the initial state is known to be 0. We have

$$J_0(0) = \min_{u_0=0,1,2} E_{w_0} \left\{ u_0 + (u_0 - w_0)^2 + J_1(\max(0, u_0 - w_0)) \right\},$$

$$u_0 = 0 : E\{\cdot\} = 0.1 \cdot J_1(0) + 0.7(1 + J_1(0)) + 0.2(4 + J_1(0)) = 4.0,$$

$$u_0 = 1 : E\{\cdot\} = 1 + 0.1(1 + J_1(1)) + 0.7 \cdot J_1(0) + 0.2(1 + J_1(0)) = 3.7,$$

$$u_0 = 2 : E\{\cdot\} = 2 + 0.1(4 + J_1(2)) + 0.7(1 + J_1(1)) + 0.2 \cdot J_1(0) = 4.818,$$

$$J_0(0) = 3.7, \quad \mu_0^*(0) = 1.$$

If the initial state were not known a priori, we would have to compute in a similar manner  $J_0(1)$  and  $J_0(2)$ , as well as the minimizing  $\mu_0$ . The reader may verify (Exercise 1.2) that these calculations yield

$$J_0(1) = 2.7, \quad \mu_0^*(1) = 0,$$

$$J_0(2) = 2.818, \quad \mu_0^*(2) = 0.$$

Thus the optimal ordering policy for each period is to order one unit if the current stock is zero and order nothing otherwise. The results of the DP algorithm are given in tabular form in Fig. 1.3.3.

### Example 1.3.3 (Optimizing a Chess Match Strategy)

Consider the chess match example of Section 1.1. There, a player can select timid play (probabilities  $p_d$  and  $1 - p_d$  for a draw or loss, respectively) or bold play (probabilities  $p_w$  and  $1 - p_w$  for a win or loss, respectively) in each game of the match. We want to formulate a DP algorithm for finding the policy that maximizes the player's probability of winning the match. Note that here we are dealing with a maximization problem. We can convert the problem to a minimization problem by changing the sign of the cost function, but a simpler alternative, which we will generally adopt, is to replace the minimization in the DP algorithm with maximization.

Let us consider the general case of an  $N$ -game match, and let the state be the *net score*, that is, the difference between the points of the player minus the points of the opponent (so a state of 0 corresponds to an even score). The optimal cost-to-go function at the start of the  $k$ th game is given by the dynamic programming recursion

$$J_k(x_k) = \max [p_d J_{k+1}(x_k) + (1 - p_d) J_{k+1}(x_k - 1), \\ p_w J_{k+1}(x_k + 1) + (1 - p_w) J_{k+1}(x_k - 1)]. \quad (1.8)$$

The maximum above is taken over the two possible decisions:

- (a) Timid play, which keeps the score at  $x_k$  with probability  $p_d$ , and changes  $x_k$  to  $x_k - 1$  with probability  $1 - p_d$ .
- (b) Bold play, which changes  $x_k$  to  $x_k + 1$  or to  $x_k - 1$  with probabilities  $p_w$  or  $(1 - p_w)$ , respectively.

It is optimal to play bold when

$$p_w J_{k+1}(x_k + 1) + (1 - p_w) J_{k+1}(x_k - 1) \geq p_d J_{k+1}(x_k) + (1 - p_d) J_{k+1}(x_k - 1)$$

or equivalently, if

$$\frac{p_w}{p_d} \geq \frac{J_{k+1}(x_k) - J_{k+1}(x_k - 1)}{J_{k+1}(x_k + 1) - J_{k+1}(x_k - 1)}. \quad (1.9)$$

The dynamic programming recursion is started with

$$J_N(x_N) = \begin{cases} 1 & \text{if } x_N > 0, \\ p_w & \text{if } x_N = 0, \\ 0 & \text{if } x_N < 0. \end{cases} \quad (1.10)$$

In this equation, we have  $J_N(0) = p_w$  because when the score is even after  $N$  games ( $x_N = 0$ ), it is optimal to play bold in the first game of sudden death.

By executing the DP algorithm (1.8) starting with the terminal condition (1.10), and using the criterion (1.9) for optimality of bold play, we find the following, assuming that  $p_d > p_w$ :

$$J_{N-1}(x_{N-1}) = 1 \text{ for } x_{N-1} > 1; \quad \text{optimal play: either}$$

$$J_{N-1}(1) = \max[p_d + (1 - p_d)p_w, p_w + (1 - p_w)p_w] \\ = p_d + (1 - p_d)p_w; \quad \text{optimal play: timid}$$

$$J_{N-1}(0) = p_w; \quad \text{optimal play: bold}$$

$$J_{N-1}(-1) = p_w^2; \quad \text{optimal play: bold}$$

$$J_{N-1}(x_{N-1}) = 0 \text{ for } x_{N-1} < -1; \quad \text{optimal play: either.}$$

Also, given  $J_{N-1}(x_{N-1})$ , and Eqs. (1.8) and (1.9) we obtain

$$J_{N-2}(0) = \max [p_d p_w + (1 - p_d)p_w^2, p_w(p_d + (1 - p_d)p_w) + (1 - p_w)p_w^2] \\ = p_w(p_w + (p_w + p_d)(1 - p_w))$$

and that if the score is even with 2 games remaining, it is optimal to play bold. Thus for a 2-game match, the optimal policy for both periods is to play timid if and only if the player is ahead in the score. The region of pairs  $(p_w, p_d)$  for which the player has a better than 50-50 chance to win a 2-game match is

$$R_2 = \left\{ (p_w, p_d) \mid J_0(0) = p_w(p_w + (p_w + p_d)(1 - p_w)) > 1/2 \right\},$$

and, as noted in the preceding section, it includes points where  $p_w < 1/2$ .

### Example 1.3.4 (Finite-State Systems)

We mentioned earlier (cf. the examples in Section 1.1) that systems with a finite number of states can be represented either in terms of a discrete-time system equation or in terms of the probabilities of transition between the states. Let us work out the DP algorithm corresponding to the latter case. We assume for the sake of the following discussion that the problem is stationary (i.e., the transition probabilities, the cost per stage, and the control constraint sets do not change from one stage to the next). Then, if

$$p_{ij}(u) = P\{x_{k+1} = j \mid x_k = i, u_k = u\}$$

are the transition probabilities, we can alternatively represent the system by the system equation (cf. the discussion of the previous section)

$$x_{k+1} = w_k,$$

where the probability distribution of the disturbance  $w_k$  is

$$P\{w_k = j \mid x_k = i, u_k = u\} = p_{ij}(u).$$

Using this system equation and denoting by  $g(i, u)$  the expected cost per stage at state  $i$  when control  $u$  is applied, the DP algorithm can be rewritten as

$$J_k(i) = \min_{u \in U(i)} [g(i, u) + E\{J_{k+1}(w_k)\}]$$

or equivalently (in view of the distribution of  $w_k$  given previously)

$$J_k(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_j p_{ij}(u) J_{k+1}(j) \right].$$

As an illustration, in the machine replacement example of Section 1.1, this algorithm takes the form

$$\begin{aligned} J_N(i) &= 0, \quad i = 1, \dots, n, \\ J_k(i) &= \min \left[ R + g(1) + J_{k+1}(1), g(i) + \sum_{j=1}^n p_{ij} J_{k+1}(j) \right]. \end{aligned}$$

The two expressions in the above minimization correspond to the two available decisions (replace or not replace the machine).

In the queuing example of Section 1.1, the DP algorithm takes the form

$$\begin{aligned} J_N(i) &= R(i), \quad i = 0, 1, \dots, n, \\ J_k(i) &= \min \left[ r(i) + c_f + \sum_{j=0}^n p_{ij}(u_f) J_{k+1}(j), r(i) + c_s + \sum_{j=0}^n p_{ij}(u_s) J_{k+1}(j) \right]. \end{aligned}$$

The two expressions in the above minimization correspond to the two possible decisions (fast and slow service).

Note that if there are  $n$  states at each stage, and  $U(i)$  contains as many as  $m$  controls, the minimization in the right-hand side of the DP algorithm requires, for each  $(i, k)$ , as many as a constant multiple of  $mn$  operations. Since there are  $nN$  state-time pairs, the total number of operations for the DP algorithm is as large as a constant multiple of  $mn^2N$  operations. By contrast, the number of all policies is exponential in  $nN$  (it is as large as  $m^{nN}$ ), so a brute force approach which enumerates all policies and compares their cost, requires an exponential number of operations in  $nN$ .

## 1.4 STATE AUGMENTATION AND OTHER REFORMULATIONS

We now discuss how to deal with situations where some of the assumptions of the basic problem are violated. Generally, in such cases the problem can be reformulated into the basic problem format. This process is called *state augmentation* because it typically involves the enlargement of the state space. The general guideline in state augmentation is to *include in the enlarged state at time  $k$  all the information that is known to the controller at time  $k$  and can be used with advantage in selecting  $u_k$* . Unfortunately, state augmentation often comes at a price: the reformulated problem may have very complex state and/or control spaces. We provide some examples.

### Time Lags

In many applications the system state  $x_{k+1}$  depends not only on the preceding state  $x_k$  and control  $u_k$  but also on earlier states and controls. In other words, states and controls influence future states with some time lag. Such situations can be handled by state augmentation; the state is expanded to include an appropriate number of earlier states and controls.

For simplicity, assume that there is at most a single period time lag in the state and control; that is, the system equation has the form

$$x_{k+1} = f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), \quad k = 1, 2, \dots, N-1, \quad (1.11)$$

$$x_1 = f_0(x_0, u_0, w_0).$$

Time lags of more than one period can be handled similarly.

If we introduce additional state variables  $y_k$  and  $s_k$ , and we make the identifications  $y_k = x_{k-1}$ ,  $s_k = u_{k-1}$ , the system equation (1.11) yields

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, y_k, u_k, s_k, w_k) \\ x_k \\ u_k \end{pmatrix}. \quad (1.12)$$

By defining  $\tilde{x}_k = (x_k, y_k, s_k)$  as the new state, we have

$$\tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, u_k, w_k),$$

where the system function  $\tilde{f}_k$  is defined from Eq. (1.12). By using the preceding equation as the system equation and by expressing the cost function in terms of the new state, the problem is reduced to the basic problem without time lags. Naturally, the control  $u_k$  should now depend on the new state  $\tilde{x}_k$ , or equivalently a policy should consist of functions  $\mu_k$  of the current state  $x_k$ , as well as the preceding state  $x_{k-1}$  and the preceding control  $u_{k-1}$ .

When the DP algorithm for the reformulated problem is translated in terms of the variables of the original problem, it takes the form

$$J_N(x_N) = g_N(x_N),$$

$$\begin{aligned} J_{N-1}(x_{N-1}, x_{N-2}, u_{N-2}) \\ = \min_{u_{N-1} \in U_{N-1}(x_{N-1})} E_{w_{N-1}} \left\{ g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \right. \\ \left. + J_N(f_{N-1}(x_{N-1}, x_{N-2}, u_{N-1}, u_{N-2}, w_{N-1})) \right\}, \end{aligned}$$

$$\begin{aligned} J_k(x_k, x_{k-1}, u_{k-1}) = \min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) \right. \\ \left. + J_{k+1}(f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), x_k, u_k) \right\}, \quad k = 1, \dots, N-2, \end{aligned}$$

$$J_0(x_0) = \min_{u_0 \in U_0(x_0)} E_{w_0} \left\{ g_0(x_0, u_0, w_0) + J_1(f_0(x_0, u_0, w_0), x_0, u_0) \right\}.$$

Similar reformulations are possible when time lags appear in the cost; for example, in the case where the cost has the form

$$E \left\{ g_N(x_N, x_{N-1}) + g_0(x_0, u_0, w_0) + \sum_{k=1}^{N-1} g_k(x_k, x_{k-1}, u_k, w_k) \right\}.$$

The extreme case of time lags in the cost arises in the nonadditive form

$$E \{ g_N(x_N, x_{N-1}, \dots, x_0, u_{N-1}, \dots, u_0, w_{N-1}, \dots, w_0) \}.$$

Then, the problem can be reduced to the basic problem format, by taking as augmented state

$$\tilde{x}_k = (x_k, x_{k-1}, \dots, x_0, u_{k-1}, \dots, u_0, w_{k-1}, \dots, w_0)$$

and  $E\{g_N(\tilde{x}_N)\}$  as reformulated cost. Policies consist of functions  $\mu_k$  of the present and past states  $x_k, \dots, x_0$ , the past controls  $u_{k-1}, \dots, u_0$ , and the past disturbances  $w_{k-1}, \dots, w_0$ . Naturally, we must assume that the past disturbances are known to the controller. Otherwise, we are faced with a problem where the state is imprecisely known to the controller. Such problems are known as problems with imperfect state information and will be discussed in Chapter 5.

### Correlated Disturbances

We turn now to the case where the disturbances  $w_k$  are correlated over time. A common situation that can be handled efficiently by state augmentation arises when the process  $w_0, \dots, w_{N-1}$  can be represented as the output of a linear system driven by independent random variables. As an example, suppose that by using statistical methods, we determine that the evolution of  $w_k$  can be modeled by an equation of the form

$$w_k = \lambda w_{k-1} + \xi_k,$$

where  $\lambda$  is a given scalar and  $\{\xi_k\}$  is a sequence of independent random vectors with given distribution. Then we can introduce an additional state variable

$$y_k = w_{k-1}$$

and obtain a new system equation

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, u_k, \lambda y_k + \xi_k) \\ \lambda y_k + \xi_k \end{pmatrix},$$

where the new state is the pair  $\tilde{x}_k = (x_k, y_k)$  and the new disturbance is the vector  $\xi_k$ .

More generally, suppose that  $w_k$  can be modeled by

$$w_k = C_k y_{k+1},$$

where

$$y_{k+1} = A_k y_k + \xi_k, \quad k = 0, \dots, N-1,$$

$A_k, C_k$  are known matrices of appropriate dimension, and  $\xi_k$  are independent random vectors with given distribution (see Fig. 1.4.1). By viewing  $y_k$  as an additional state variable, we obtain the new system equation

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \\ A_k y_k + \xi_k \end{pmatrix}.$$

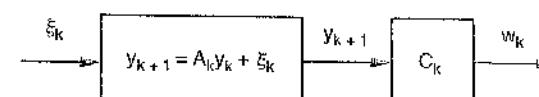


Figure 1.4.1 Representing correlated disturbances as the output of a linear system driven by independent random vectors.

Note that in order to have perfect state information, the controller must be able to observe  $y_k$ . Unfortunately, this is true only in the minority of practical cases; for example when  $C_k$  is the identity matrix and  $w_{k-1}$  is observed before  $u_k$  is applied. In the case of perfect state information, the DP algorithm takes the form

$$\begin{aligned} J_N(x_N, y_N) &= g_N(x_N), \\ J_k(x_k, y_k) &= \min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \right. \\ &\quad \left. + J_{k+1}(f_k(x_k, u_k, C_k(A_k y_k + \xi_k)), A_k y_k + \xi_k) \right\}. \end{aligned}$$

### Forecasts

Finally, consider the case where at time  $k$  the controller has access to a forecast  $y_k$  that results in a reassessment of the probability distribution of  $w_k$  and possibly of future disturbances. For example,  $y_k$  may be an exact prediction of  $w_k$  or an exact prediction that the probability distribution of  $w_k$  is a specific one out of a finite collection of distributions. Forecasts of interest in practice are, for example, probabilistic predictions on the state of the weather, the interest rate for money, and the demand for inventory.

Generally, forecasts can be handled by state augmentation although the reformulation into the basic problem format may be quite complex. We will treat here only a simple special case.

Assume that at the beginning of each period  $k$ , the controller receives an accurate prediction that the next disturbance  $w_k$  will be selected according to a particular probability distribution out of a given collection of distributions  $\{Q_1, \dots, Q_m\}$ ; that is, if the forecast is  $i$ , then  $w_k$  is selected according to  $Q_i$ . The a priori probability that the forecast will be  $i$  is denoted by  $p_i$  and is given.

For instance, suppose that in our earlier inventory example the demand  $w_k$  is determined according to one of three distributions  $Q_1$ ,  $Q_2$ , and  $Q_3$ , corresponding to "small," "medium," and "large" demand. Each of the three types of demand occurs with a given probability at each time period, independently of the values of demand at previous time periods. However, the inventory manager, prior to ordering  $u_k$ , gets to know through a forecast the type of demand that will occur. (Note that it is the probability distribution of demand that becomes known through the forecast, not the demand itself.)

The forecasting process can be represented by means of the equation

$$y_{k+1} = \xi_k,$$

where  $y_{k+1}$  can take the values  $1, \dots, m$ , corresponding to the  $m$  possible forecasts, and  $\xi_k$  is a random variable taking the value  $i$  with probability

$p_i$ . The interpretation here is that when  $\xi_k$  takes the value  $i$ , then  $w_{k+1}$  will occur according to the distribution  $Q_i$ .

By combining the system equation with the forecast equation  $y_{k+1} = \xi_k$ , we obtain an augmented system given by

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, u_k, w_k) \\ \xi_k \end{pmatrix}.$$

The new state is

$$\tilde{x}_k = (x_k, y_k),$$

and because the forecast  $y_k$  is known at time  $k$ , perfect state information prevails. The new disturbance is

$$\bar{w}_k = (w_k, \xi_k),$$

and its probability distribution is determined by the distributions  $Q_i$  and the probabilities  $p_i$ , and depends explicitly on  $\tilde{x}_k$  (via  $y_k$ ) but not on the prior disturbances.

Thus, by suitable reformulation of the cost, the problem can be cast into the basic problem format. Note that the control applied depends on both the current state and the current forecast. The DP algorithm takes the form

$$\begin{aligned} J_N(x_N, y_N) &= g_N(x_N), \\ J_k(x_k, y_k) &= \min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) \right. \\ &\quad \left. + \sum_{i=1}^m p_i J_{k+1}(f_k(x_k, u_k, w_k), i) \mid y_k \right\}, \end{aligned} \quad (1.13)$$

where  $y_k$  may take the values  $1, \dots, m$ , and the expectation over  $w_k$  is taken with respect to the distribution  $Q_{y_k}$ .

It should be clear that the preceding formulation admits several extensions. One example is the case where forecasts can be influenced by the control action and involve several future disturbances. However, the price for these extensions is increased complexity of the corresponding DP algorithm.

### Simplification for Uncontrollable State Components

When augmenting the state of a given system one often ends up with composite states, consisting of several components. It turns out that if some of these components cannot be affected by the choice of control, the DP algorithm can be simplified considerably, as we will now describe.

Let the state of the system be a composite  $(x_k, y_k)$  of two components  $x_k$  and  $y_k$ . The evolution of the main component,  $x_k$ , is affected by the control  $u_k$  according to the equation

$$x_{k+1} = f_k(x_k, y_k, u_k, w_k),$$

where the probability distribution  $P_k(w_k | x_k, y_k, u_k)$  is given. The evolution of the other component,  $y_k$ , is governed by a given conditional distribution  $P_k(y_k | x_k)$  and cannot be affected by the control, except indirectly through  $x_k$ . One is tempted to view  $y_k$  as a disturbance, but there is a difference:  $y_k$  is observed by the controller before applying  $u_k$ , while  $w_k$  occurs after  $u_k$  is applied, and indeed  $w_k$  may probabilistically depend on  $u_k$ .

We will formulate a DP algorithm that is executed over the controllable component of the state, with the dependence on the uncontrollable component being “averaged out.” In particular, let  $J_k(x_k, y_k)$  denote the optimal cost-to-go at stage  $k$  and state  $(x_k, y_k)$ , and define

$$\hat{J}_k(x_k) = E_{y_k} \{ J_k(x_k, y_k) | x_k \}.$$

We will derive a DP algorithm that generates  $\hat{J}_k(x_k)$ .

Indeed, we have

$$\begin{aligned} \hat{J}_k(x_k) &= E_{y_k} \{ J_k(x_k, y_k) | x_k \} \\ &= E_{y_k} \left\{ \min_{u_k \in U_k(x_k, y_k)} E_{w_k, x_{k+1}, y_{k+1}} \{ g_k(x_k, y_k, u_k, w_k) \right. \\ &\quad \left. + J_{k+1}(x_{k+1}, y_{k+1}) | x_k, y_k, u_k \} | x_k \right\} \\ &= E_{y_k} \left\{ \min_{u_k \in U_k(x_k, y_k)} E_{w_k, x_{k+1}} \{ g_k(x_k, y_k, u_k, w_k) \right. \\ &\quad \left. + E_{y_{k+1}} \{ J_{k+1}(x_{k+1}, y_{k+1}) | x_{k+1} \} | x_k, y_k, u_k \} | x_k \right\}, \end{aligned}$$

and finally

$$\begin{aligned} \hat{J}_k(x_k) &= E_{y_k} \left\{ \min_{u_k \in U_k(x_k, y_k)} E_{w_k} \{ g_k(x_k, y_k, u_k, w_k) \right. \\ &\quad \left. + \hat{J}_{k+1}(f_k(x_k, y_k, u_k, w_k)) \} | x_k \right\}. \end{aligned} \quad (1.14)$$

The advantage of this equivalent DP algorithm is that it is executed over a significantly reduced state space. For example, if  $x_k$  takes  $n$  possible values and  $y_k$  takes  $m$  possible values, then DP is executed over  $n$  states instead of  $nm$  states. Note, however, that the minimization in the right-hand side of the preceding equation yields an optimal control law as a function of the full state  $(x_k, y_k)$ .

As an example, consider the augmented state resulting from the incorporation of forecasts, as described earlier. Then, the forecast  $y_k$  represents

an uncontrolled state component, so that the DP algorithm can be simplified as in Eq. (1.14). In particular, by defining

$$\hat{J}_k(x_k) = \sum_{i=1}^m p_i J_k(x_k, i), \quad k = 0, 1, \dots, N-1,$$

and

$$\hat{J}_N(x_N) = g_N(x_N),$$

we have, using Eq. (1.13),

$$\begin{aligned} \hat{J}_k(x_k) &= \sum_{i=1}^m p_i \min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) \right. \\ &\quad \left. + \hat{J}_{k+1}(f_k(x_k, u_k, w_k)) | y_k = i \right\}, \end{aligned}$$

which is executed over the space of  $x_k$  rather than  $x_k$  and  $y_k$ .

Uncontrolled state components often occur in arrival systems, such as queueing, where action must be taken in response to a random event (such as a customer arrival) that cannot be influenced by the choice of control. Then the state of the arrival system must be augmented to include the random event, but the DP algorithm can be executed over a smaller space, as per Eq. (1.14). Here is another example of similar type.

#### Example 1.4.1: (Tetris)

Tetris is a popular video game played on a two-dimensional grid. Each square in the grid can be full or empty, making up a “wall of bricks” with “holes” and a “jagged top”. The squares fill up as blocks of different shapes fall from the top of the grid and are added to the top of the wall. As a given block falls, the player can move horizontally and rotate the block in all possible ways, subject to the constraints imposed by the sides of the grid and the top of the wall. The falling blocks are generated independently according to some probability distribution, defined over a finite set of standard shapes. The game starts with an empty grid and ends when a square in the top row becomes full and the top of the wall reaches the top of the grid. When a row of full squares is created, this row is removed, the bricks lying above this row move one row downward, and the player scores a point. The player’s objective is to maximize the score attained (total number of rows removed) within  $N$  steps or up to termination of the game, whichever occurs first.

We can model the problem of finding an optimal tetris playing strategy as a stochastic DP problem. The control, denoted by  $u$ , is the horizontal positioning and rotation applied to the falling block. The state consists of two components:

- (1) The board position, i.e., a binary description of the full/empty status of each square, denoted by  $x$ .

- (2) The shape of the current falling block, denoted by  $y$ .

There is also an additional termination state which is cost-free. Once the state reaches the termination state, it stays there with no change in cost.

The shape  $y$  is generated according to a probability distribution  $p(y)$ , independently of the control, so it can be viewed as an uncontrollable state component. The DP algorithm (1.14) is executed over the space of  $x$  and has the intuitive form

$$\hat{J}_k(x) = \sum_y p(y) \max_u [g(x, y, u) + \hat{J}_{k+1}(f(x, y, u))], \quad \text{for all } x,$$

where  $g(x, y, u)$  and  $f(x, y, u)$  are the number of points scored (rows removed), and the board position (or termination state) when the state is  $(x, y)$  and control  $u$  is applied, respectively. Note, however, that despite the simplification in the DP algorithm achieved by eliminating the uncontrollable portion of the state, the number of states  $x$  is enormous, and the problem can only be addressed by suboptimal methods, which will be discussed in Chapter 6 and in Vol. II.

## 1.5 SOME MATHEMATICAL ISSUES

Let us now discuss some technical issues relating to the basic problem formulation and the validity of the DP algorithm. The reader who is not mathematically inclined need not be concerned about these issues and can skip this section without loss of continuity.

Once an admissible policy  $\{\mu_0, \dots, \mu_{N-1}\}$  is adopted, the following sequence of events is envisioned at the typical stage  $k$ :

1. The controller observes  $x_k$  and applies  $u_k = \mu_k(x_k)$ .
2. The disturbance  $w_k$  is generated according to the given distribution  $P_k(\cdot | x_k, \mu_k(x_k))$ .
3. The cost  $g_k(x_k, \mu_k(x_k), w_k)$  is incurred and added to previous costs.
4. The next state  $x_{k+1}$  is generated according to the system equation

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k).$$

If this is the last stage ( $k = N - 1$ ), the terminal cost  $g_N(x_N)$  is added to previous costs and the process terminates. Otherwise,  $k$  is incremented, and the same sequence of events is repeated at the next stage.

For each stage, the above process is well-defined and is couched in precise probabilistic terms. Matters are, however, complicated by the need to

view the cost as a well-defined random variable with well-defined expected value. The framework of probability theory requires that for each policy we define an underlying probability space, that is, a set  $\Omega$ , a collection of events in  $\Omega$ , and a probability measure on these events. In addition, the cost must be a well-defined random variable on this space in the sense of Appendix C (a measurable function from the probability space into the real line in the terminology of measure-theoretic probability theory). For this to be true, additional (measurability) assumptions on the functions  $f_k$ ,  $g_k$ , and  $\mu_k$  may be required, and it may be necessary to introduce additional structure on the spaces  $S_k$ ,  $C_k$ , and  $D_k$ . Furthermore, these assumptions may restrict the class of admissible policies, since the functions  $\mu_k$  may be constrained to satisfy additional (measurability) requirements.

Thus, unless these additional assumptions and structure are specified, the basic problem is formulated inadequately from a mathematical point of view. Unfortunately, a rigorous formulation for general state, control, and disturbance spaces is well beyond the mathematical framework of this introductory book and will not be undertaken here. Nonetheless, it turns out that these difficulties are mainly technical and do not substantially affect the basic results to be obtained. For this reason, we find it convenient to proceed with informal derivations and arguments; this is consistent with most of the literature on the subject.

We would like to stress, however, that under at least one frequently satisfied assumption, the mathematical difficulties mentioned above disappear. In particular, let us assume that the disturbance spaces  $D_k$  are all countable and the expected values of all terms in the cost are finite for every admissible policy (this is true in particular if the spaces  $D_k$  are finite sets). Then, for every admissible policy, the expected values of all the cost terms can be written as (possibly infinite) sums involving the probabilities of the elements of the spaces  $D_k$ .

Alternatively, one may write the cost as

$$J_\pi(x_0) = E_{x_1, \dots, x_N} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} \tilde{g}_k(x_k, \mu_k(x_k)) \right\}, \quad (1.15)$$

where

$$\tilde{g}_k(x_k, \mu_k(x_k)) = E_{w_k} \left\{ g_k(x_k, \mu_k(x_k), w_k) \mid x_k, \mu_k(x_k) \right\},$$

with the preceding expectation taken with respect to the distribution  $P_k(\cdot | x_k, \mu_k(x_k))$  defined on the countable set  $D_k$ . Then one may take as the basic probability space the Cartesian product of the spaces  $\tilde{S}_k$ ,  $k = 1, \dots, N$ , given for all  $k$  by

$$\tilde{S}_{k+1} = \{x_{k+1} \in S_{k+1} \mid x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), x_k \in \tilde{S}_k, w_k \in D_k\},$$

where  $\tilde{S}_0 = \{x_0\}$ . The set  $\tilde{S}_k$  is the subset of all states that can be reached at time  $k$  when the policy  $\{\mu_0, \dots, \mu_{N-1}\}$  is used. Because the disturbance spaces  $D_k$  are countable, the sets  $\tilde{S}_k$  are also countable (this is true since the union of any countable collection of countable sets is a countable set). The system equation  $x_{k+1} = f_k(x_k, \mu_k(x_k), w_k)$ , the probability distributions  $P_k(\cdot | x_k, \mu_k(x_k))$ , the initial state  $x_0$ , and the policy  $\{\mu_0, \dots, \mu_{N-1}\}$  define a probability distribution on the countable set  $\tilde{S}_1 \times \dots \times \tilde{S}_N$ , and the expected value in the cost expression (1.15) is defined with respect to this latter distribution.

Let us now give a more detailed proof of the validity of the DP algorithm (Prop. 1.3.1). We assume that the disturbance  $w_k$  takes a finite or countable number of values and the expected values of all terms in the expression of the cost function are finite for every admissible policy  $\pi$ . Furthermore, the functions  $J_k(x_k)$  generated by the DP algorithm are finite for all states  $x_k$  and times  $k$ . We do not need to assume that the minimum over  $u_k$  in the definition of  $J_k(x_k)$  is attained by some  $u_k \in U(x_k)$ .

For any admissible policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$  and each  $k = 0, 1, \dots, N-1$ , denote  $\pi^k = \{\mu_k, \mu_{k+1}, \dots, \mu_{N-1}\}$ . For  $k = 0, 1, \dots, N-1$ , let  $J_k^*(x_k)$  be the optimal cost for the  $(N-k)$ -stage problem that starts at state  $x_k$  and time  $k$ , and ends at time  $N$ ; that is,

$$J_k^*(x_k) = \min_{\pi^k} E \left\{ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right\}.$$

For  $k = N$ , we define  $J_N^*(x_N) = g_N(x_N)$ . We will show by induction that the functions  $J_k^*$  are equal to the functions  $J_k$  generated by the DP algorithm, so that for  $k = 0$ , we will obtain the desired result.

For any  $\epsilon > 0$ , and for all  $k$  and  $x_k$ , let  $\mu_k^\epsilon(x_k)$  attain the minimum in the equation

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}, \quad k = 0, 1, \dots, N-1, \quad (1.16)$$

within  $\epsilon$ ; that is, for all  $x_k$  and  $k$ , we have  $\mu_k^\epsilon(x_k) \in U_k(x_k)$  and

$$E \left\{ g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^\epsilon(x_k), w_k)) \right\} \leq J_k(x_k) + \epsilon. \quad (1.17)$$

Let  $J_k^\epsilon(x_k)$  be the expected cost starting at state  $x_k$  at time  $k$ , and using the policy  $\{\mu_k^\epsilon, \mu_{k+1}^\epsilon, \dots, \mu_{N-1}^\epsilon\}$ . We will show that for all  $x_k$  and  $k$ , we have

$$J_k(x_k) \leq J_k^\epsilon(x_k) \leq J_k(x_k) + (N - k)\epsilon, \quad (1.18)$$

$$J_k^*(x_k) \leq J_k^\epsilon(x_k) \leq J_k^*(x_k) + (N - k)\epsilon, \quad (1.19)$$

$$J_k(x_k) = J_k^*(x_k). \quad (1.20)$$

It is seen using Eq. (1.17) that the inequalities (1.18) and (1.19) hold for  $k = N-1$ . By taking  $\epsilon \rightarrow 0$  in Eqs. (1.18) and (1.19), it is also seen that  $J_{N-1} = J_{N-1}^*$ . Assume that Eqs. (1.18)-(1.20) hold for index  $k+1$ . We will show that they also hold for index  $k$ .

Indeed, we have

$$\begin{aligned} J_k^\epsilon(x_k) &= E \left\{ g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}^\epsilon(f_k(x_k, \mu_k^\epsilon(x_k), w_k)) \right\} \\ &\leq E \left\{ g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^\epsilon(x_k), w_k)) \right\} + (N - k - 1)\epsilon \\ &\leq J_k(x_k) + \epsilon + (N - k - 1)\epsilon \\ &= J_k(x_k) + (N - k)\epsilon, \end{aligned}$$

where the first equation holds by the definition of  $J_k^\epsilon$ , the first inequality holds by the induction hypothesis, and the second inequality holds Eq. (1.17). We also have

$$\begin{aligned} J_k^*(x_k) &= E \left\{ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}^*(f_k(x_k, \mu_k^*(x_k), w_k)) \right\} \\ &\geq E \left\{ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right\} \\ &\geq \min_{u_k \in U(x_k)} E \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} \\ &= J_k(x_k), \end{aligned}$$

where the first inequality holds by the induction hypothesis. Combining the preceding two relations, we see that Eq. (1.18) holds for index  $k$ .

For every policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , we have

$$\begin{aligned} J_k^\epsilon(x_k) &= E \left\{ g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}^\epsilon(f_k(x_k, \mu_k^\epsilon(x_k), w_k)) \right\} \\ &\leq E \left\{ g_k(x_k, \mu_k^\epsilon(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^\epsilon(x_k), w_k)) \right\} + (N - k - 1)\epsilon \\ &\leq \min_{u_k \in U(x_k)} E \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} + (N - k)\epsilon \\ &\leq \min_{u_k \in U(x_k)} E \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} + (N - k)\epsilon \\ &\leq E \left\{ g_k(x_k, \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k(x_k), w_k)) \right\} + (N - k)\epsilon \\ &= J_{\pi^k}(x_k) + (N - k)\epsilon, \end{aligned}$$

where the first inequality holds by the induction hypothesis, and the second inequality holds by Eq. (1.17). Taking the minimum over  $\pi^k$  in the preceding relation, we obtain for all  $x_k$

$$J_k^\epsilon(x_k) \leq J_k^*(x_k) + (N - k)\epsilon.$$

We also have by the definition of  $J_k^*$ , for all  $x_k$ ,

$$J_k^*(x_k) \leq J_k^c(x_k).$$

Combining the preceding two relations, we see that Eq. (1.19) holds for index  $k$ . Finally, Eq. (1.20) follows from Eqs. (1.18) and (1.19), by taking  $\epsilon \rightarrow 0$ , and the induction is complete.

Note that by using  $c = 0$  in the relation

$$J_0^c(x_k) \leq J_0^*(x_k) + Nc,$$

[cf. Eq. (1.19)], we see that a policy that attains the minimum for all  $x_k$  and  $k$  in Eq. (1.16) is optimal.

In conclusion, the basic problem has been formulated rigorously, and the DP algorithm has been proved rigorously only when the disturbance spaces  $D_0, \dots, D_{N-1}$  are countable sets, and the expected values of all the cost expressions associated with the problem and the DP algorithm are finite. In the absence of these assumptions, the reader should interpret subsequent results and conclusions as essentially correct but mathematically imprecise statements. In fact, when discussing infinite horizon problems (where the need for precision is greater), we will make the countability assumption explicit.

We note, however, that the advanced reader will have little difficulty in establishing most of our subsequent results concerning specific finite horizon applications, even if the countability assumption is not satisfied. This can be done by using the DP algorithm as a verification theorem. In particular, if one can find within a subset of policies  $\tilde{\Pi}$  (such as those satisfying certain measurability restrictions) a policy that attains the minimum in the DP algorithm, then this policy can be readily shown to be optimal within  $\tilde{\Pi}$ . This result is developed in Exercise 1.12, and can be used by the mathematically oriented reader to establish rigorously many of our subsequent results concerning specific applications. For example, in linear-quadratic problems (Section 4.1) one determines from the DP algorithm a policy in closed form, which is linear in the current state. When  $w_k$  can take uncountably many values, it is necessary that admissible policies consist of Borel measurable functions  $\mu_k$ . Since the linear policy obtained from the DP algorithm belongs to this class, the result of Exercise 1.12 guarantees that this policy is optimal. For a rigorous mathematical treatment of DP that resolves the associated measurability issues and supplements the present text, see the book by Bertsekas and Shreve [BeS78].

## 1.6 DYNAMIC PROGRAMMING AND MINIMAX CONTROL

The problem of optimal control of uncertain systems has traditionally been treated in a stochastic framework, whereby all uncertain quantities are described by probability distributions, and the expected value of the cost is

minimized. However, in many practical situations a stochastic description of the uncertainty may not be available, and one may have information with less detailed structure, such as bounds on the magnitude of the uncertain quantities. In other words, one may know a set within which the uncertain quantities are known to lie, but may not know the corresponding probability distribution. Under these circumstances one may use a minimax approach, whereby the worst possible values of the uncertain quantities within the given set are assumed to occur.

The minimax approach for decision making under uncertainty is described in Appendix G and is contrasted with the expected cost approach, which we have been following so far. In its simplest form, the corresponding decision problem is described by a triplet  $(\Pi, W, J)$ , where  $\Pi$  is the set of policies under consideration,  $W$  is the set in which the uncertain quantities are known to belong, and  $J : \Pi \times W \rightarrow (-\infty, +\infty]$  is a given cost function. The objective is to

$$\text{minimize } \max_{w \in W} J(\pi, w)$$

over all  $\pi \in \Pi$ .

It is possible to formulate a minimax counterpart to the basic problem with perfect state information. This problem is a special case of the abstract minimax problem above, as discussed more fully in Appendix G. Generally, it is unusual for even the simplest special cases of this problem to admit a closed-form solution. However, a computational solution using DP is possible, and our purpose in this section is to describe the corresponding algorithm.

In the framework of the basic problem, consider the case where the disturbances  $w_0, w_1, \dots, w_{N-1}$  do not have a probabilistic description but rather are known to belong to corresponding given sets  $W_k(x_k, u_k) \subset D_k$ ,  $k = 0, 1, \dots, N-1$ , which may depend on the current state  $x_k$  and control  $u_k$ . Consider the problem of finding a policy  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$  with  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k$  and  $k$ , which minimizes the cost function

$$J_\pi(x_0) = \max_{\substack{w_k \in W_k(x_k, \mu_k(x_k)) \\ k=0,1,\dots,N-1}} \left[ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right].$$

The DP algorithm for this problem takes the following form, which resembles the one corresponding to the stochastic basic problem (maximization is used in place of expectation):

$$J_N(x_N) = g_N(x_N), \quad (1.21)$$

$$J_k(x_k) = \min_{u_k \in U(x_k)} \max_{w_k \in W_k(x_k, u_k)} \left[ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right]. \quad (1.22)$$

This algorithm can be explained by using a principle of optimality type of argument. In particular, we consider the tail subproblem whereby we are at state  $x_k$  at time  $k$ , and we wish to minimize the “cost-to-go”

$$\max_{\substack{w_i \in W_i(x_i, \mu_i(x_i)) \\ i=k, k+1, \dots, N-1}} \left[ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, \mu_i(x_i), w_i) \right],$$

and we argue that if  $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$  is an optimal policy for the minimax problem, then the truncated policy  $\{\mu_k^*, \mu_{k+1}^*, \dots, \mu_{N-1}^*\}$  is optimal for the tail subproblem. The optimal cost of this subproblem is  $J_k(x_k)$ , as given by the DP algorithm (1.21)-(1.22). The algorithm expresses the intuitively clear fact that when at state  $x_k$  at time  $k$ , then regardless of what happened in the past, we should choose  $u_k$  that minimizes the worst/maximum value over  $w_k$  of the sum of the current stage cost plus the optimal cost of the tail subproblem that starts from the next state.

We will now give a mathematical proof that the DP algorithm (1.21)-(1.22) is valid, and that the optimal cost is equal to  $J_0(x_0)$ . For this it is necessary to assume that  $J_k(x_k) > -\infty$  for all  $x_k$  and  $k$ . This is analogous to the assumption we made in the preceding section for the validity of the DP algorithm under stochastic disturbances, i.e., that the values  $J_k(x_k)$  generated by the DP algorithm are finite for all states  $x_k$  and stages  $k$ . The following lemma provides the key argument.

**Lemma 1.6.1:** Let  $f : W \rightarrow X$  be a function, and  $M$  be the set of all functions  $\mu : X \rightarrow U$ , where  $W$ ,  $X$ , and  $U$  are some sets. Then for any functions  $G_0 : W \rightarrow (-\infty, \infty]$  and  $G_1 : X \times U \rightarrow (-\infty, \infty]$  such that

$$\min_{u \in U} G_1(f(w), u) > -\infty, \quad \text{for all } w \in W,$$

we have

$$\min_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] = \max_{w \in W} [G_0(w) + \min_{u \in U} G_1(f(w), u)].$$

**Proof:** We have for all  $\mu \in M$

$$\max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] \geq \max_{w \in W} [G_0(w) + \min_{u \in U} G_1(f(w), u)]$$

and by taking the minimum over  $\mu \in M$ , we obtain

$$\min_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] \geq \max_{w \in W} [G_0(w) + \min_{u \in U} G_1(f(w), u)]. \quad (1.23)$$

To show the reverse inequality, for any  $\epsilon > 0$ , let  $\mu_\epsilon \in M$  be such that  $G_1(f(w), \mu_\epsilon(f(w))) \leq \min_{u \in U} G_1(f(w), u) + \epsilon$ , for all  $w \in W$ .

[Such a  $\mu_\epsilon$  exists because of the assumption  $\min_{u \in U} G_1(f(w), u) > -\infty$ .] Then

$$\begin{aligned} \min_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] \\ \leq \max_{w \in W} [G_0(w) + G_1(f(w), \mu_\epsilon(f(w)))] \\ \leq \max_{w \in W} [G_0(w) + \min_{u \in U} G_1(f(w), u)] + \epsilon. \end{aligned}$$

Since  $\epsilon > 0$  can be taken arbitrarily small, we obtain the reverse to Eq. (1.23), and the desired result follows. Q.E.D.

To see how the conclusion of the lemma can fail without the condition  $\min_{u \in U} G_1(f(w), u) > -\infty$  for all  $w$ , let  $u$  be a scalar, let  $w = (w_1, w_2)$  be a two-dimensional vector, and let there be no constraints on  $u$  and  $w$  ( $U = \mathbb{R}$ ,  $W = \mathbb{R} \times \mathbb{R}$ , where  $\mathbb{R}$  is the real line). Let also

$$G_0(w) = w_1, \quad f(w) = w_2, \quad G_1(f(w), u) = f(w) + u.$$

Then, for all  $\mu \in M$  we have,

$$\max_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] = \max_{w_1 \in \mathbb{R}, w_2 \in \mathbb{R}} [w_1 + w_2 + \mu(w_2)] = \infty,$$

so that

$$\min_{\mu \in M} \max_{w \in W} [G_0(w) + G_1(f(w), \mu(f(w)))] = \infty.$$

On the other hand,

$$\max_{w \in W} \left[ G_0(w) + \min_{u \in U} G_1(f(w), u) \right] = \max_{w_1 \in \mathbb{R}, w_2 \in \mathbb{R}} \left[ w_1 + \min_{u \in \mathbb{R}} [w_2 + u] \right] = -\infty,$$

since  $\min_{u \in \mathbb{R}} [w_2 + u] = -\infty$  for all  $w_2$ .

We now turn to proving the DP algorithm (1.21)-(1.22). The proof is similar to the one for the DP algorithm for stochastic problems. The optimal cost  $J^*(x_0)$  of the problem is given by

$$\begin{aligned} J^*(x_0) &= \min_{\mu_0} \cdots \min_{\mu_{N-1}} \max_{w_0 \in W[x_0, \mu_0(x_0)]} \cdots \max_{w_{N-1} \in W[x_{N-1}, \mu_{N-1}(x_{N-1})]} \\ &\quad \left[ \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) \right] \\ &= \min_{\mu_0} \cdots \min_{\mu_{N-2}} \left[ \min_{\mu_{N-1}} \max_{w_0 \in W[x_0, \mu_0(x_0)]} \cdots \max_{w_{N-2} \in W[x_{N-2}, \mu_{N-2}(x_{N-2})]} \right. \\ &\quad \left. \left[ \sum_{k=0}^{N-2} g_k(x_k, \mu_k(x_k), w_k) + \max_{w_{N-1} \in W[x_{N-1}, \mu_{N-1}(x_{N-1})]} \right. \right. \\ &\quad \left. \left. \left[ g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) + J_N(x_N) \right] \right] \right]. \end{aligned}$$

We can interchange the minimum over  $\mu_{N-1}$  and the maximum over  $w_0, \dots, w_{N-2}$  by applying Lemma 1.6.1 with the identifications

$$w = (w_0, w_1, \dots, w_{N-2}), \quad u = u_{N-1}, \quad f(w) = x_{N-1},$$

$$G_0(w) = \begin{cases} \sum_{k=0}^{N-2} g_k(x_k, \mu_k(x_k), w_k) & \text{if } w_k \in W_k(x_k, \mu_k(x_k)) \text{ for all } k, \\ \infty & \text{otherwise,} \end{cases}$$

$$G_1(f(w), u) = \begin{cases} \hat{G}_1(f(w), u) & \text{if } u \in U_{N-1}(f(w)), \\ \infty & \text{otherwise,} \end{cases}$$

where

$$\hat{G}_1(f(w), u) = \max_{w_{N-1} \in W_{N-1}(f(w), u)} \left[ g_{N-1}(f(w), u, w_{N-1}) + J_N(f(w), u, w_{N-1})) \right],$$

to obtain

$$\begin{aligned} J^*(x_0) &= \min_{\mu_0} \cdots \min_{\mu_{N-2}} \max_{w_0 \in W[x_0, \mu_0(x_0)]} \cdots \max_{w_{N-2} \in W[x_{N-2}, \mu_{N-2}(x_{N-2})]} \\ &\quad \left[ \sum_{k=0}^{N-2} g_k(x_k, \mu_k(x_k), w_k) + J_{N-1}(x_{N-1}) \right]. \end{aligned} \quad (1.24)$$

The required condition  $\min_{u \in U} G_1(f(w), u) > -\infty$  for all  $w$  (required for application of Lemma 1.6.1) is implied by the assumption  $J_{N-1}(x_{N-1}) > -\infty$  for all  $x_{N-1}$ . Now, by working with the expression for  $J^*(x_0)$  in Eq. (1.24), and by similarly continuing backwards, with  $N-1$  in place of  $N$ , etc., after  $N$  steps we obtain  $J^*(x_0) = J_0(x_0)$ , which is the desired relation. The line of argument just given also shows that an optimal policy for the minimax problem can be constructed by minimizing in the right-hand side of the DP Eq. (1.22), similar to the case of the DP algorithm for the stochastic basic problem.

Unfortunately, as mentioned earlier, there are hardly any interesting examples of an analytical, closed-form solution of the DP algorithm (1.21)-(1.22). A computational solution, requires roughly comparable effort to the one of the stochastic DP algorithm. Instead of the expectation operation, one must carry out a maximization operation for each  $x_k$  and  $k$ .

Minimax control problems will be revisited in Chapter 4 in the context of reachability of target sets and target tubes (Section 4.6.2), and in Chapter 6 in the context of competitive games and computer chess (Section 6.3), and model predictive control (Section 6.5).

## 1.7 NOTES, SOURCES, AND EXERCISES

Dynamic programming is a simple mathematical technique that has been used for many years by engineers, mathematicians, and social scientists in a variety of contexts. It was Bellman, however, who realized in the early fifties that DP could be developed (in conjunction with the then appearing digital computer) into a systematic tool for optimization. In his influential books [Bel57], [BeD62], Bellman demonstrated the broad scope of DP and helped streamline its theory.

Following Bellman's works, there was much research on DP. In particular, the mathematical and algorithmic aspects of infinite horizon problems were extensively investigated, extensions to continuous-time problems were formulated and analyzed, and the mathematical issues discussed in Section 1.5, relating to the formulation of DP problems, were addressed. In addition, DP was used in a very broad variety of applications, ranging from many branches of engineering to statistics, economics, finance, and some of the social sciences. Samples of these applications will be given in subsequent chapters.

## EXERCISES

### 1.1

Consider the system

$$x_{k+1} = x_k + u_k + w_k, \quad k = 0, 1, 2, 3,$$

with initial state  $x_0 = 5$ , and the cost function

$$\sum_{k=0}^3 (x_k^2 + u_k^2).$$

Apply the DP algorithm for the following three cases:

- The control constraint set  $U_k(x_k)$  is  $\{u \mid 0 \leq u_k \leq 5, u : \text{integer}\}$  for all  $x_k$  and  $k$ , and the disturbance  $w_k$  is equal to zero for all  $k$ .
- The control constraint and the disturbance  $w_k$  are as in part (a), but there is in addition a constraint  $x_4 = 5$  on the final state. Hint: For this problem you need to define a state space for  $x_4$  that consists of just the value  $x_4 = 5$ , and also to redefine  $U_3(x_3)$ . Alternatively, you may use a terminal cost  $g_4(x_4)$  equal to a very large number for  $x_4 \neq 5$ .

- (c) The control constraint is as in part (a) and the disturbance  $w_k$  takes the values  $-1$  and  $1$  with equal probability  $1/2$  for all  $x_k$  and  $u_k$ , except if  $x_k + u_k$  is equal to  $0$  or  $5$ , in which case  $w_k = 0$  with probability  $1$ .

## 1.2

Carry out the calculations needed to verify that  $J_0(1) = 2.67$  and  $J_0(2) = 2.608$  in Example 1.3.2.

## 1.3

Suppose we have a machine that is either running or is broken down. If it runs throughout one week, it makes a gross profit of \$100. If it fails during the week, gross profit is zero. If it is running at the start of the week and we perform preventive maintenance, the probability that it will fail during the week is  $0.4$ . If we do not perform such maintenance, the probability of failure is  $0.7$ . However, maintenance will cost \$20. When the machine is broken down at the start of the week, it may either be repaired at a cost of \$40, in which case it will fail during the week with a probability of  $0.4$ , or it may be replaced at a cost of \$150 by a new machine that is guaranteed to run through its first week of operation. Find the optimal repair, replacement, and maintenance policy that maximizes total profit over four weeks, assuming a new machine at the start of the first week.

## 1.4

A game of the blackjack variety is played by two players as follows: Both players throw a die. The first player, knowing his opponent's result, may stop or may throw the die again and add the result to the result of his previous throw. He then may stop or throw again and add the result of the new throw to the sum of his previous throws. He may repeat this process as many times as he wishes. If his sum exceeds seven (i.e., he busts), he loses the game. If he stops before exceeding seven, the second player takes over and throws the die successively until the sum of his throws is four or higher. If the sum of the second player is over seven, he loses the game. Otherwise the player with the larger sum wins, and in case of a tie the second player wins. The problem is to determine a stopping strategy for the first player that maximizes his probability of winning for each possible initial throw of the second player. Formulate the problem in terms of DP and find an optimal stopping strategy for the case where the second player's initial throw is three. Hint: Take  $N = 6$  and a state space consisting of the following 15 states:

$x^1$ : busted

$x^{1+}$ : already stopped at sum  $i$  ( $1 \leq i \leq 7$ ),

$x^{8+i}$ : current sum is  $i$  but the player has not yet stopped ( $1 \leq i \leq 7$ ).

The optimal strategy is to throw until the sum is four or higher.

## 1.5 (Computer Assignment)

In the classical game of blackjack the player draws cards knowing only one card of the dealer. The player loses upon reaching a sum of cards exceeding 21. If the player stops before exceeding 21, the dealer draws cards until reaching 17 or higher. The dealer loses upon reaching a sum exceeding 21 or stopping at a lower sum than the player's. If player and dealer end up with an equal sum no one wins. In all other cases the dealer wins. An ace for the player may be counted as a 1 or an 11 as the player chooses. An ace for the dealer is counted as an 11 if this results in a sum from 17 to 21 and as a 1 otherwise. Jacks, queens, and kings count as 10 for both dealer and player. We assume an infinite card deck so the probability of a particular card showing up is independent of earlier cards.

- For every possible initial dealer card, calculate the probability that the dealer will reach a sum of 17, 18, 19, 20, 21, or over 21.
- Calculate the optimal choice of the player (draw or stop) for each of the possible combinations of dealer's card and player's sum of 12 to 20. Assume that the player's cards do not include an ace.
- Repeat part (b) for the case where the player's cards include an ace.

## 1.6 (Discounted Cost per Stage)

In the framework of the basic problem, consider the case where the cost is of the form

$$\min_{w_k} \left\{ \alpha^N g_N(x_N) + \sum_{k=0}^{N-1} \alpha^k g_k(x_k, u_k, w_k) \right\},$$

where  $\alpha$  is a discount factor with  $0 < \alpha < 1$ . Show that an alternate form of the DP algorithm is given by

$$V_N(x_N) = g_N(x_N),$$

$$V_k(x_k) = \min_{u_k \in U_k(x_k)} \min_{w_k} \left\{ g_k(x_k, u_k, w_k) + \alpha V_{k+1}(f_k(x_k, u_k, w_k)) \right\}.$$

## 1.7 (Exponential Cost Function)

In the framework of the basic problem, consider the case where the cost is of the form

$$\min_{w_k} \left\{ \exp \left( g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right) \right\}.$$

- Show that the optimal cost and an optimal policy can be obtained from the DP-like algorithm

$$J_N(x_N) = \exp(g_N(x_N)),$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E \left\{ J_{k+1}(f_k(x_k, u_k, w_k)) \exp(g_k(x_k, u_k, w_k)) \right\}.$$

- (b) Define the functions  $V_k(x_k) = \ln J_k(x_k)$ . Assume also that  $g_k$  is a function of  $x_k$  and  $u_k$  only (and not of  $w_k$ ). Show that the above algorithm can be rewritten as

$$V_N(x_N) = g_N(x_N),$$

$$V_k(x_k) = \min_{u_k \in U_k(x_k)} \left\{ g_k(x_k, u_k) + \ln E \left\{ \exp(V_{k+1}(f_k(x_k, u_k, w_k))) \right\} \right\}.$$

*Note:* The exponential cost function is an example of a *risk-sensitive cost function* that can be used to encode a preference for policies with a small variance of the cost  $g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$ . The associated problems have a lot of interesting properties, which are discussed in several sources, e.g., Whittle [Whi90], Fernandez-Gaucherand and Markus [FeM94], James, Baras, and Elliott [JBE94], Basar and Bernhard [BaB95].

## 1.8 (Terminating Process)

In the framework of the basic problem, consider the case where the system evolution terminates at time  $i$  when a given value  $\bar{w}_i$  of the disturbance at time  $i$  occurs, or when a termination decision  $u_i$  is made by the controller. If termination occurs at time  $i$ , the resulting cost is

$$T \cdot 1 \sum_{k=0}^i g_k(x_k, u_k, w_k),$$

where  $T$  is a termination cost. If the process has not terminated up to the final time  $N$ , the resulting cost is  $g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$ . Reformulate the problem into the framework of the basic problem. Hint: Augment the state space with a special termination state.

## 1.9 (Multiplicative Cost)

In the framework of the basic problem, consider the case where the cost has the multiplicative form

$$\min_{u_k \in U_k(x_k)} \left\{ g_N(x_N) \cdot g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \cdots g_0(x_0, u_0, w_0) \right\}.$$

Develop a DP-like algorithm for this problem assuming that  $g_k(x_k, u_k, w_k) \geq 0$  for all  $x_k, u_k, w_k$ , and  $k$ .

## 1.10

Assume that we have a vessel whose maximum weight capacity is  $z$  and whose cargo is to consist of different quantities of  $N$  different items. Let  $v_i$  denote the value of the  $i$ th type of item,  $w_i$  the weight of  $i$ th type of item, and  $x_i$  the number of items of type  $i$  that are loaded in the vessel. The problem is to find the most valuable cargo, i.e., to maximize  $\sum_{i=1}^N x_i v_i$  subject to the constraints  $\sum_{i=1}^N x_i w_i \leq z$  and  $x_i = 0, 1, 2, \dots$ . Formulate this problem in terms of DP.

## 1.11

Consider a device consisting of  $N$  stages connected in series, where each stage consists of a particular component. The components are subject to failure, and to increase the reliability of the device duplicate components are provided. For  $j = 1, 2, \dots, N$ , let  $(1 + m_j)$  be the number of components for the  $j$ th stage, let  $p_j(m_j)$  be the probability of successful operation of the  $j$ th stage when  $(1 + m_j)$  components are used, and let  $c_j$  denote the cost of a single component at the  $j$ th stage. Formulate in terms of DP the problem of finding the number of components at each stage that maximize the reliability of the device expressed by

$$p_1(m_1) \cdot p_2(m_2) \cdots p_N(m_N),$$

subject to the cost constraint  $\sum_{j=1}^N c_j m_j \leq A$ , where  $A > 0$  is given.

## 1.12 (Minimization over a Subset of Policies)

This problem is primarily of theoretical interest (see the end of Section 1.5). Consider a variation of the basic problem whereby we seek

$$\min_{\pi \in \tilde{\Pi}} J_\pi(x_0),$$

where  $\tilde{\Pi}$  is some given subset of the set of sequences  $\{\mu_0, \mu_1, \dots, \mu_{N-1}\}$  of functions  $\mu_k : S_k \rightarrow C_k$  with  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k \in S_k$ . Assume that

$$\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$$

belongs to  $\tilde{\Pi}$  and attains the minimum in the DP algorithm; that is, for all  $k = 0, 1, \dots, N-1$  and  $x_k \in S_k$ ,

$$\begin{aligned} J_k(x_k) &= \min_{w_k} \left\{ g_k(x_k, \mu_k^*(x_k), w_k) + J_{k+1}(f_k(x_k, \mu_k^*(x_k), w_k)) \right\} \\ &= \min_{u_k \in U_k(x_k)} \min_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}, \end{aligned}$$

with  $J_N(x_N) = g_N(x_N)$ . Assume further that the functions  $J_k$  are real-valued and that the preceding expectations are well-defined and finite. Show that  $\pi^*$  is optimal within  $\tilde{\Pi}$  and that the corresponding optimal cost is equal to  $J_0(x_0)$ .

### 1.13 (Semilinear Systems)

Consider a problem involving the system

$$x_{k+1} = A_k x_k + f_k(u_k) + w_k,$$

where  $x_k \in \mathbb{R}^n$ ,  $f_k$  are given functions, and  $A_k$  and  $w_k$  are random  $n \times n$  matrices and  $n$ -vectors, respectively, with given probability distributions that do not depend on  $x_k$ ,  $u_k$  or prior values of  $A_k$  and  $w_k$ . Assume that the cost is of the form

$$E_{A_k, w_k} \left\{ c_N' x_N + \sum_{k=0}^{N-1} (c_k' x_k + g_k(\mu_k(x_k))) \right\},$$

where  $c_k$  are given vectors and  $g_k$  are given functions. Show that if the optimal cost for this problem is finite and the control constraint sets  $U_k(x_k)$  are independent of  $x_k$ , then the cost-to-go functions of the DP algorithm are affine (linear plus constant). Assuming that there is at least one optimal policy, show that there exists an optimal policy that consists of constant functions  $\mu_k^*$ ; that is,  $\mu_k^*(x_k) = \text{constant}$  for all  $x_k \in \mathbb{R}^n$ .

### 1.14

A farmer annually producing  $x_k$  units of a certain crop stores  $(1 - u_k)x_k$  units of his production, where  $0 \leq u_k \leq 1$ , and invests the remaining  $u_k x_k$  units, thus increasing the next year's production to a level  $x_{k+1}$  given by

$$x_{k+1} = x_k + w_k u_k x_k, \quad k = 0, 1, \dots, N-1.$$

The scalars  $w_k$  are independent random variables with identical probability distributions that do not depend either on  $x_k$  or  $u_k$ . Furthermore,  $E\{w_k\} = \bar{w} > 0$ . The problem is to find the optimal investment policy that maximizes the total expected product stored over  $N$  years

$$E_{w_k} \left\{ x_N + \sum_{k=0}^{N-1} (1 - u_k)x_k \right\}.$$

Show the optimality of the following policy that consists of constant functions:

- (a) If  $\bar{w} > 1$ ,  $\mu_0^*(x_0) = \dots = \mu_{N-1}^*(x_{N-1}) = 1$ .
- (b) If  $0 < \bar{w} < 1/N$ ,  $\mu_0^*(x_0) = \dots = \mu_{N-1}^*(x_{N-1}) = 0$ .
- (c) If  $1/N \leq \bar{w} \leq 1$ ,

$$\mu_0^*(x_0) = \dots = \mu_{N-\bar{k}-1}^*(x_{N-\bar{k}-1}) = 1,$$

$$\mu_{N-\bar{k}}^*(x_{N-\bar{k}}) = \dots = \mu_{N-1}^*(x_{N-1}) = 0,$$

where  $\bar{k}$  is such that  $1/(\bar{k} + 1) < \bar{w} \leq 1/\bar{k}$ .

### 1.15

Let  $x_k$  denote the number of educators in a certain country at time  $k$  and let  $y_k$  denote the number of research scientists at time  $k$ . New scientists (potential educators or research scientists) are produced during the  $k$ th period by educators at a rate  $\gamma_k$  per educator, while educators and research scientists leave the field due to death, retirement, and transfer at a rate  $\delta_k$ . The scalars  $\gamma_k$ ,  $k = 0, 1, \dots, N-1$ , are independent identically distributed random variables taking values within a closed and bounded interval of positive numbers. Similarly  $\delta_k$ ,  $k = 0, 1, \dots, N-1$ , are independent identically distributed and take values in an interval  $[\delta, \delta']$  with  $0 < \delta \leq \delta' < 1$ . By means of incentives, a science policy maker can determine the proportion  $u_k$  of new scientists produced at time  $k$  who become educators. Thus, the number of research scientists and educators evolves according to the equations

$$x_{k+1} = (1 - \delta_k)x_k + u_k \gamma_k x_k,$$

$$y_{k+1} = (1 - \delta_k)y_k + (1 - u_k)\gamma_k x_k.$$

The initial numbers  $x_0, y_0$  are known, and it is required to find a policy

$$\{\mu_0^*(x_0, y_0), \dots, \mu_{N-1}^*(x_{N-1}, y_{N-1})\}$$

with

$$0 < \alpha \leq \mu_k^*(x_k, y_k) \leq \beta < 1, \quad \text{for all } x_k, y_k, \text{ and } k,$$

which maximizes  $E_{\gamma_k, \delta_k}\{y_N\}$  (i.e., the expected final number of research scientists after  $N$  periods). The scalars  $\alpha$  and  $\beta$  are given.

- (a) Show that the cost-to-go functions  $J_k(x_k, y_k)$  are linear; that is, for some scalars  $\xi_k, \zeta_k$ ,

$$J_k(x_k, y_k) = \xi_k x_k + \zeta_k y_k.$$

- (b) Derive an optimal policy  $\{\mu_0^*, \dots, \mu_{N-1}^*\}$  under the assumption

$$E\{\gamma_k\} > E\{\delta_k\}$$

and show that this optimal policy can consist of constant functions.

- (c) Assume that the proportion of new scientists who become educators at time  $k$  is  $u_k + \epsilon_k$  (rather than  $u_k$ ), where  $\epsilon_k$  are identically distributed independent random variables that are also independent of  $\gamma_k, \delta_k$  and take values in the interval  $[-\alpha, 1 - \beta]$ . Derive the form of the cost-to-go functions and the optimal policy.

### 1.16

Given a sequence of matrix multiplications

$$M_1 M_2 \cdots M_k M_{k+1} \cdots M_N,$$

where each  $M_k$  is a matrix of dimension  $n_k \times n_{k+1}$ , the order in which multiplications are carried out can make a difference. For example, if  $n_1 = 1$ ,  $n_2 = 10$ ,  $n_3 = 1$ , and  $n_4 = 10$ , the calculation  $((M_1 M_2) M_3)$  requires 20 scalar multiplications, but the calculation  $(M_1 (M_2 M_3))$  requires 200 scalar multiplications (multiplying an  $m \times n$  matrix with an  $n \times k$  matrix requires  $mnk$  scalar multiplications).

- (a) Derive a DP algorithm for finding the optimal multiplication order [any order is allowed, including orders that involve multiple partial products each consisting of two or more adjacent matrices, e.g.,  $((M_1 M_2)(M_3 M_4))$ ]. Solve the problem for  $N = 3$ ,  $n_1 = 2$ ,  $n_2 = 10$ ,  $n_3 = 5$ , and  $n_4 = 1$ .
- (b) Derive a DP algorithm for finding the optimal multiplication order within the class of orders where at each step, we maintain only one partial product that consists only of adjacent matrices, e.g.,  $((M_1 (M_2 M_3)) M_4)$ .

### 1.17

The paragraphing problem deals with breaking up a sequence of  $N$  words of given lengths into lines of length  $A$ . Let  $w_1, \dots, w_N$  be the words and let  $L_1, \dots, L_N$  be their lengths. In a simple version of the problem, words are separated by blanks whose ideal width is  $b$ , but blanks can stretch or shrink if necessary, so that a line  $w_i, w_{i+1}, \dots, w_{i+k}$  has length exactly  $A$ . The cost associated with the line is  $(k+1)|b' - b|$ , where  $b' = (A - L_i - \dots - L_{i+k})/(k+1)$  is the actual average width of the blanks, except if we have the last line ( $N = i+k$ ), in which case the cost is zero when  $b' \geq b$ . Formulate a DP algorithm for finding the minimum cost separation. *Hint:* Consider the subproblems of optimally separating  $w_i, \dots, w_N$  for  $i = 1, \dots, N$ .

### 1.18 [Shr81] [www](#)

A decision maker must continually choose between two activities over a time interval  $[0, T]$ . Choosing activity  $i$  at time  $t$ , where  $i = 1, 2$ , earns reward at a rate  $g_i(t)$ , and every switch between the two activities costs  $c > 0$ . Thus, for example, the reward for starting with activity 1, switching to 2 at time  $t_1$ , and switching back to 1 at time  $t_2 > t_1$  earns total reward

$$\int_0^{t_1} g_1(t) dt + \int_{t_1}^{t_2} g_2(t) dt + \int_{t_2}^T g_1(t) dt - 2c.$$

We want to find a set of switching times that maximize the total reward. Assume that the function  $g_1(t) - g_2(t)$  changes sign a finite number of times in the interval  $[0, T]$ . Formulate the problem as a finite horizon problem and write the corresponding DP algorithm.

### 1.19 (Games)

- (a) Consider a smaller version of a popular puzzle game. Three square tiles numbered 1, 2, and 3 are placed in a  $2 \times 2$  grid with one space left empty. The two tiles adjacent to the empty space can be moved into that space, thereby creating new configurations. Use a DP argument to answer the question whether it is possible to generate a given configuration starting from any other configuration.
- (b) From a pile of eleven matchsticks, two players take turns removing one or four sticks. The player who removes the last stick wins. Use a DP argument to show that there is a winning strategy for the player who plays first.

### 1.20 (The Counterfeit Coin Problem)

We are given six coins, one of which is counterfeit and is known to have different weight than the rest. Construct a strategy to find the counterfeit coin using a two-pan scale in a minimum average number of tries. *Hint:* There are two initial decisions that make sense: (1) test two of the coins against two others, and (2) test one of the coins against one other.

### 1.21 (Regular Polygon Theorem) [www](#)

According to a famous theorem (attributed to the ancient Greek geometer Zenodorus), of all  $N$ -side polygons inscribed in a given circle, those that are regular (all sides are equal) have maximal area.

- (a) Prove the theorem by applying DP to a suitable problem involving sequential placement of  $N$  points in the circle.
- (b) Use DP to solve the problem of placing a given number of points on a subarc of the circle, so as to maximize the area of the polygon whose vertices are these points, the endpoints of the subarc, and the center of the circle.

### 1.22 (Inscribed Polygon of Maximal Perimeter)

Consider the problem of inscribing an  $N$ -side polygon in a given circle, so that the polygon has maximal perimeter.

- (a) Formulate the problem as a DP problem involving sequential placement of  $N$  points in the circle.
- (b) Use DP to show that the optimal polygon is regular (all sides are equal).

### 1.23 (Monotonicity Property of DP)

An evident, yet very important property of the DP algorithm is that if the terminal cost  $g_N$  is changed to a uniformly larger cost  $\bar{g}_N$  [i.e.,  $g_N(x_N) \leq \bar{g}_N(x_N)$  for all  $x_N$ ], then the last stage cost-to-go  $J_{N-1}(x_{N-1})$  will be uniformly increased. More generally, given two functions  $J_{k+1}$  and  $\bar{J}_{k+1}$  with  $J_{k+1}(x_{k+1}) \leq \bar{J}_{k+1}(x_{k+1})$  for all  $x_{k+1}$ , we have, for all  $x_k$  and  $u_k \in U_k(x_k)$ ,

$$\begin{aligned} & E_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\} \\ & \leq E_{w_k} \left\{ g_k(x_k, u_k, w_k) + \bar{J}_{k+1}(f_k(x_k, u_k, w_k)) \right\}. \end{aligned}$$

Suppose now that in the basic problem the system and cost are time invariant; that is,  $S_k \equiv S$ ,  $C_k \equiv C$ ,  $D_k \equiv D$ ,  $f_k \equiv f$ ,  $U_k \equiv U$ ,  $P_k \equiv P$ , and  $g_k \equiv g$  for some  $S$ ,  $C$ ,  $D$ ,  $f$ ,  $U$ ,  $P$ , and  $g$ . Show that if in the DP algorithm we have  $J_{N-1}(x) \leq J_N(x)$  for all  $x \in S$ , then

$$J_k(x) \leq J_{k+1}(x), \quad \text{for all } x \in S \text{ and } k.$$

Similarly, if we have  $J_{N-1}(x) \geq J_N(x)$  for all  $x \in S$ , then

$$J_k(x) \geq J_{k+1}(x), \quad \text{for all } x \in S \text{ and } k.$$

### 1.24 (Traveling Repairman Problem)

A repairman must service  $n$  sites, which are located along a line and are sequentially numbered  $1, 2, \dots, n$ . The repairman starts at a given site  $s$  with  $1 < s < n$ , and is constrained to service only sites that are adjacent to the ones serviced so far, i.e., if he has already serviced sites  $i, i+1, \dots, j$ , then he may service next only site  $i-1$  (assuming  $1 < i$ ) or site  $j+1$  (assuming  $j < n$ ). There is a waiting cost  $c_i$  for each time period that site  $i$  has remained unserviced and there is a travel cost  $t_{ij}$  for servicing site  $j$  immediately after servicing site  $i$ . Formulate a DP algorithm for finding a minimum cost service schedule.

### 1.25 [www](#)

An unscrupulous innkeeper charges a different rate for a room as the day progresses, depending on whether he has many or few vacancies. His objective is to maximize his expected total income during the day. Let  $x$  be the number of empty rooms at the start of the day, and let  $y$  be the number of customers that will ask for a room in the course of the day. We assume (somewhat unrealistically) that the innkeeper knows  $y$  with certainty, and upon arrival of a customer, quotes one of  $m$  prices  $r_i$ ,  $i = 1, \dots, m$ , where  $0 < r_1 \leq r_2 \leq \dots \leq r_m$ . A quote of a rate  $r_i$  is accepted with probability  $p_i$  and is rejected with probability  $1 - p_i$ , in which case the customer departs, never to return during that day.

- (a) Formulate this as a problem with  $y$  stages and show that the maximal expected income, as a function of  $x$  and  $y$ , satisfies the recursion

$$J(x, y) = \max_{i=1, \dots, m} \left[ p_i (r_i + J(x-1, y-1)) + (1-p_i) J(x, y-1) \right],$$

for all  $x \geq 1$  and  $y \geq 1$ , with initial conditions

$$J(x, 0) = J(0, y) = 0, \quad \text{for all } x \text{ and } y.$$

Assuming that the product  $p_i r_i$  is monotonically nondecreasing with  $i$ , and that  $p_i$  is monotonically nonincreasing with  $i$ , show that the innkeeper should always charge the highest rate  $r_m$ .

- (b) Consider a variant of the problem where each arriving customer, with probability  $p_i$ , offers a price  $r_i$  for a room, which the innkeeper may accept or reject, in which case the customer departs, never to return during that day. Show that an appropriate DP algorithm is

$$J(x, y) = \sum_{i=1}^m p_i \max[r_i + J(x-1, y-1), J(x, y-1)],$$

with initial conditions

$$J(x, 0) = J(0, y) = 0, \quad \text{for all } x \text{ and } y.$$

Show also that for given  $x$  and  $y$  it is optimal to accept a customer's offer if it is larger than some threshold  $\bar{r}(x, y)$ . Hint: This part is related to DP for uncontrollable state components (cf. Section 1.4).

### 1.26 (Investing in a Stock) [www](#)

An investor observes at the beginning of each period  $k$  the price  $x_k$  of a stock and decides whether to buy 1 unit, sell 1 unit, or do nothing. There is a transaction cost  $c$  for buying or selling. The stock price can take one of  $n$  different values  $v^1, \dots, v^n$  and the transition probabilities  $p_{ij}^k = P\{x_{k+1} = v^j | x_k = v^i\}$  are known. The investor wants to maximize the total worth of his stock at a fixed final period  $N$  minus his investment costs from period 0 to period  $N-1$  (revenue from a sale is viewed as negative cost). We assume that the function

$$P_k(x) = E\{x_N | x_k = x\} - x$$

is monotonically nonincreasing as a function of  $x$ ; that is, the expected profit from a purchase is a nonincreasing function of the purchase price.

- (a) Assume that the investor starts with  $N$  or more units of stock and an unlimited amount of cash, so that a purchase or sale decision is possible at each period regardless of the past decisions and the current price. For every period  $k$ , let  $\underline{x}_k$  be the largest value of  $x \in \{v^1, \dots, v^n\}$  such that

$P_k(x) > c$ , and let  $\bar{x}_k$  be the smallest value of  $x \in \{v^1, \dots, v^n\}$  such that  $P_k(x) < -c$ . Show that it is optimal to buy if  $x_k \leq \underline{x}_k$ , sell if  $\bar{x}_k \leq x_k$ , and do nothing otherwise. Hint: Formulate the problem as one of maximizing

$$E \left\{ \sum_{k=0}^{N-1} (u_k P_k(x_k) - c |u_k|) \right\},$$

where  $u_k \in \{-1, 0, 1\}$ .

- (b) Formulate an efficient DP algorithm for the case where the investor starts with less than  $N$  units of stock and an unlimited amount of cash. Show that it is still optimal to buy if  $x_k \leq \underline{x}_k$  and it is still not optimal to sell if  $x_k < \bar{x}_k$ . Could it be optimal to buy at any prices  $x_k$  greater than  $\bar{x}_k$ ?
- (c) Consider the situation where the investor initially has  $N$  or more units of stock and there is a constraint that for any time  $k$  the number of purchase decisions up to  $k$  should not exceed the number of sale decisions up to  $k$  by more than a given fixed number  $m$  (this models approximately the situation where the investor has a limited initial amount of cash). Formulate an efficient DP algorithm for this case. Show that it is still optimal to sell if  $\bar{x}_k \leq x_k$  and it is still not optimal to buy if  $\underline{x}_k < x_k$ .
- (d) Consider the situation where there are restrictions on both the initial amount of stock as in part (b), and the number of purchase decisions as in part (c). Derive a DP algorithm for this problem.
- (e) How would the analysis of (a)-(d) be affected if cash is invested at a given fixed interest rate?

## 2

# Deterministic Systems and the Shortest Path Problem

## Contents

2.1. Finite-State Systems and Shortest Paths . . . . .	p. 64
2.2. Some Shortest Path Applications . . . . .	p. 68
2.2.1. Critical Path Analysis . . . . .	p. 68
2.2.2. Hidden Markov Models and the Viterbi Algorithm . . . . .	p. 70
2.3. Shortest Path Algorithms . . . . .	p. 77
2.3.1. Label Correcting Methods . . . . .	p. 78
2.3.2. Label Correcting Variations - A* Algorithm . . . . .	p. 87
2.3.3. Branch-and-Bound . . . . .	p. 88
2.3.4. Constrained and Multiobjective Problems . . . . .	p. 91
2.4. Notes, Sources, and Exercises . . . . .	p. 97

In this chapter, we focus on deterministic problems, that is, problems where each disturbance  $w_k$  can take only one value. Such problems arise in many important contexts and they also arise in cases where the problem is really stochastic but, as an approximation, the disturbance is fixed at some typical value; see Chapter 6.

An important property of deterministic problems is that, in contrast with stochastic problems, *using feedback results in no advantage in terms of cost reduction*. In other words, minimizing the cost over admissible policies  $\{\mu_0, \dots, \mu_{N-1}\}$  results in the same optimal cost as minimizing over sequences of control vectors  $\{u_0, \dots, u_{N-1}\}$ . This is true because given a policy  $\{\mu_0, \dots, \mu_{N-1}\}$  and the initial state  $x_0$ , the future states are perfectly predictable through the equation

$$x_{k+1} = f_k(x_k, \mu_k(x_k)), \quad k = 0, 1, \dots, N-1,$$

and the corresponding controls are perfectly predictable through the equation

$$u_k = \mu_k(x_k), \quad k = 0, 1, \dots, N-1.$$

Thus, the cost achieved by an admissible policy  $\{\mu_0, \dots, \mu_{N-1}\}$  for a deterministic problem is also achieved by the control sequence  $\{u_0, \dots, u_{N-1}\}$  defined above. As a result, we may restrict attention to sequences of controls without loss of optimality.

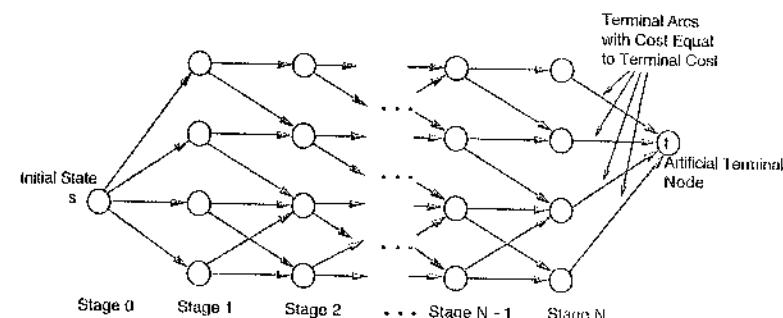
The difference just discussed between deterministic and stochastic problems often has important computational implications. In particular, in a deterministic problem with a “continuous space” character (states and controls are Euclidean vectors), optimal control sequences may be found by deterministic variational techniques to be discussed in Chapter 3, and by widely used iterative optimal control algorithms such as steepest descent, conjugate gradient, and Newton’s method (see e.g., nonlinear programming texts such as Bertsekas [Ber99] or Luenberger [Lue84]). These algorithms, when applicable, are usually more efficient than DP. On the other hand, DP has a wider scope of applicability since it can handle difficult constraint sets such as integer or discrete sets. Furthermore, DP leads to a globally optimal solution as opposed to variational techniques, for which this cannot be guaranteed in general.

In this chapter, we consider deterministic problems with a discrete character for which variational optimal control techniques are inapplicable, so that specialized forms of DP are the principal solution methods.

## 2.1 FINITE-STATE SYSTEMS AND SHORTEST PATHS

Consider a deterministic problem where the state space  $S_k$  is a finite set for each  $k$ . Then at any state  $x_k$ , a control  $u_k$  can be associated with a

transition from the state  $x_k$  to the state  $f_k(x_k, u_k)$ , at a cost  $g_k(x_k, u_k)$ . Thus a finite-state deterministic problem can be equivalently represented by a graph such as the one of Fig. 2.1.1, where the arcs correspond to transitions between states at successive stages and each arc has an associated cost. To handle the final stage, an artificial terminal node  $t$  has been added. Each state  $x_N$  at stage  $N$  is connected to the terminal node  $t$  with an arc having cost  $g_N(x_N)$ . Control sequences correspond to paths originating at the initial state (node  $s$  at stage 0) and terminating at one of the nodes corresponding to the final stage  $N$ . If we view the cost of an arc as its length, we see that *a deterministic finite-state problem is equivalent to finding a minimum-length (or shortest) path from the initial node  $s$  of the graph to the terminal node  $t$* . Here, by a path we mean a sequence of arcs of the form  $(j_1, j_2), (j_2, j_3), \dots, (j_{k-1}, j_k)$ , and by the length of a path we mean the sum of the lengths of its arcs.



**Figure 2.1.1** Transition graph for a deterministic finite-state system. Nodes correspond to states. An arc with start and end nodes  $x_k$  and  $x_{k+1}$ , respectively, corresponds to a transition of the form  $x_{k+1} = f_k(x_k, u_k)$ . We view the cost  $g_k(x_k, u_k)$  of the transition as the length of this arc. The problem is equivalent to finding a shortest path from the initial node  $s$  to the terminal node  $t$ .

Let us denote

$$a_{ij}^k = \text{Cost of transition at stage } k \text{ from state } i \in S_k \text{ to state } j \in S_{k+1},$$

$$a_{ii}^N = \text{Terminal cost of state } i \in S_N \text{ [which is } g_N(i)],$$

where we adopt the convention  $a_{ij}^k = \infty$  if there is no control that moves the state from  $i$  to  $j$  at stage  $k$ . The DP algorithm takes the form

$$J_N(i) = a_{it}^N, \quad i \in S_N, \quad (2.1)$$

$$J_k(i) = \min_{j \in S_{k+1}} [a_{ij}^k + J_{k+1}(j)], \quad i \in S_k, \quad k = 0, 1, \dots, N-1. \quad (2.2)$$

The optimal cost is  $J_0(s)$  and is equal to the length of the shortest path from  $s$  to  $t$ .

### A Forward DP Algorithm

The preceding algorithm proceeds *backward* in time. It is possible to derive an equivalent algorithm that proceeds *forward* in time by means of the following simple observation. An optimal path from  $s$  to  $t$  is also an optimal path from  $t$  to  $s$  in a “reverse” shortest path problem where the direction of each arc is reversed and its length is left unchanged. The DP algorithm corresponding to this “reverse” problem starts from the states  $x_1 \in S_1$  of stage 1, proceeds to states  $x_2 \in S_2$  of stage 2, and continues all the way to states  $x_N \in S_N$  of stage  $N$ . It is given by

$$\bar{J}_N(j) = a_{sj}^0, \quad j \in S_1, \quad (2.3)$$

$$\bar{J}_k(j) = \min_{i \in S_{N-k}} [a_{ij}^{N-k} + \bar{J}_{k+1}(i)], \quad j \in S_{N-k+1}, \quad k = 1, 2, \dots, N-1. \quad (2.4)$$

The optimal cost is

$$\bar{J}_0(t) = \min_{i \in S_N} [a_{it}^N + \bar{J}_1(i)].$$

The backward algorithm (2.1)-(2.2) and the forward algorithm (2.3)-(2.4) yield the same result in the sense that

$$J_0(s) = \bar{J}_0(t),$$

and an optimal control sequence (or shortest path) obtained from any one of the two is optimal for the original problem. We may view  $\bar{J}_k(j)$  in Eq. (2.4) as an *optimal cost-to-arrive* to state  $j$  from the initial state  $s$ . This should be contrasted with  $J_k(i)$  in Eq. (2.2), which represents the optimal cost-to-go from state  $i$  to the terminal state  $t$ .

An important use of the forward DP algorithm arises in real-time applications where the stage  $k$  problem data are unknown prior to stage  $k$ , and are revealed to the controller just before stage  $k$  begins. An example will be given in connection with the state estimation of Hidden Markov Models in Section 2.2.2. Note that to derive the forward DP algorithm, we used the shortest path formulation, which is available only for deterministic problems. Indeed, for stochastic problems, there is no analog of the forward DP algorithm.

In conclusion, a *deterministic finite-state problem is equivalent to a special type of shortest path problem and can be solved by either the ordinary (backward) DP algorithm or by an alternative forward DP algorithm*. It is also interesting to note that *any shortest path problem can be posed as a deterministic finite-state DP problem*, as we now show.

### Converting a Shortest Path Problem to a Deterministic Finite-State Problem

Let  $\{1, 2, \dots, N, t\}$  be the set of nodes of a graph, and let  $a_{ij}$  be the cost of moving from node  $i$  to node  $j$  (also referred to as the *length* of the arc joining  $i$  and  $j$ ). Node  $t$  is a special node, which we call the *destination*. We allow the possibility  $a_{ij} = \infty$  to account for the case where there is no arc joining nodes  $i$  and  $j$ . We want to find a shortest path from each node  $i$  to node  $t$ , i.e., a sequence of moves that minimizes total cost to get to  $t$  from each of the nodes  $1, 2, \dots, N$ .

For the problem to have a solution, it is necessary to make an assumption relating to cycles, i.e., paths of the form  $(i, j_1), (j_1, j_2), \dots, (j_k, i)$  that start and end at the same node. We must exclude the possibility that a cycle has negative total length. Otherwise, it would be possible to decrease the length of some paths to arbitrarily small values simply by adding more and more negative-length cycles. We thus assume that *all cycles have nonnegative length*. With this assumption, it is clear that an optimal path need not take more than  $N$  moves, so we may limit the number of moves to  $N$ . We formulate the problem as one where *we require exactly  $N$  moves but allow degenerate moves from a node  $i$  to itself with cost  $a_{ii} = 0$* . We denote for  $i = 1, \dots, N$ ,  $k = 0, \dots, N-1$ ,

$$J_k(i) = \text{optimal cost of getting from } i \text{ to } t \text{ in } N-k \text{ moves.}$$

Then the cost of the optimal path from  $i$  to  $t$  is  $J_0(i)$ .

It is possible to formulate this problem within the framework of the basic problem and subsequently apply the DP algorithm. For simplicity, however, we write directly the DP equation, which takes the intuitively clear form

optimal cost from  $i$  to  $t$  in  $N-k$  moves

$$= \min_{j=1, \dots, N} [a_{ij} + (\text{optimal cost from } j \text{ to } t \text{ in } N-k-1 \text{ moves})],$$

or

$$J_k(i) = \min_{j=1, \dots, N} [a_{ij} + J_{k+1}(j)], \quad k = 0, 1, \dots, N-2,$$

with

$$J_{N-1}(i) = a_{it}, \quad i = 1, 2, \dots, N.$$

The optimal policy when at node  $i$  after  $k$  moves is to move to a node  $j^*$  that minimizes  $a_{ij} + J_{k+1}(j)$  over all  $j = 1, \dots, N$ . If the optimal path obtained from the algorithm contains degenerate moves from a node to itself, this simply means that the path involves in reality less than  $N$  moves.

Note that if for some  $k > 0$ , we have  $J_k(i) = J_{k+1}(i)$  for all  $i$ , then subsequent DP iterations will not change the values of the cost-to-go

$\{J_{k-m}(i) = J_k(i)$  for all  $m > 0$  and  $i\}$ , so the algorithm can be terminated with  $J_k(i)$  being the shortest distance from  $i$  to  $t$ , for all  $i$ .

To demonstrate the algorithm, consider the problem shown in Fig. 2.1.2(a) where the costs  $a_{ij}$  with  $i \neq j$  are shown along the connecting line segments (we assume  $a_{ij} = a_{ji}$ ). Figure 2.1.2(b) shows the cost-to-go  $J_k(i)$  at each  $i$  and  $k$  together with the optimal paths.

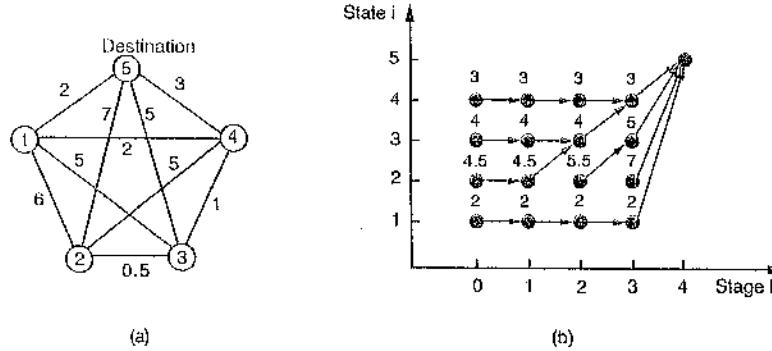


Figure 2.1.2 (a) Shortest path problem data. The destination is node 5. Arc lengths are equal in both directions and are shown along the line segments connecting nodes. (b) Costs-to-go generated by the DP algorithm. The number along stage  $k$  and state  $i$  is  $J_k(i)$ . Arrows indicate the optimal moves at each stage and node. The optimal paths are

$$1 \rightarrow 5, \quad 2 \rightarrow 3 \rightarrow 4 \rightarrow 5, \quad 3 \rightarrow 4 \rightarrow 5, \quad 4 \rightarrow 5.$$

## 2.2 SOME SHORTEST PATH APPLICATIONS

The shortest path problem appears in many diverse contexts. We provide some examples.

### 2.2.1 Critical Path Analysis

Consider the planning of a project involving several activities, some of which must be completed before others can begin. The duration of each activity is known in advance. We want to find the time required to complete the project, as well as the *critical* activities, those that even if slightly delayed will result in a corresponding delay of completion of the overall project.

The problem can be represented by a graph with nodes  $1, \dots, N$  such as the one shown in Fig. 2.2.1. Here nodes represent completion of some

phase of the project. An arc  $(i, j)$  represents an activity that starts once phase  $i$  is completed and has known duration  $t_{ij} > 0$ . A phase (node)  $j$  is completed when all activities or arcs  $(i, j)$  that are incoming to  $j$  are completed. The special nodes 1 and  $N$  represent the start and end of the project. Node 1 has no incoming arcs, while node  $N$  has no outgoing arcs. Furthermore, there is at least one path from node 1 to every other node.

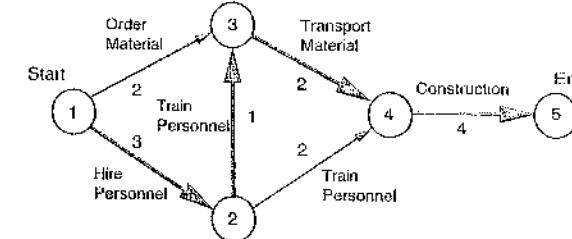


Figure 2.2.1 Graph of an activity network. Arcs represent activities and are labeled by the corresponding duration. Nodes represent completion of some phase of the project. A phase is completed if all activities associated with incoming arcs at the corresponding node are completed. The project is completed when all phases are completed. The project duration time is the length of the longest path from node 1 to node 5, which is shown with thick line.

An important characteristic of an activity network is that it is *acyclic*; that is, it has no cycles. This is inherent in the problem formulation and the interpretation of nodes as phase completions.

For any path  $p = \{(1, j_1), (j_1, j_2), \dots, (j_k, i)\}$  from node 1 to a node  $i$ , let  $D_p$  be the duration of the path defined as the sum of durations of its activities; that is,

$$D_p = t_{1j_1} + t_{j_1j_2} + \dots + t_{j_ki}.$$

Then the time  $T_i$  required to complete phase  $i$  is

$$T_i = \max_{\substack{\text{paths } p \\ \text{from 1 to } i}} D_p.$$

Thus to find  $T_i$ , we should find the *longest* path from 1 to  $i$ . This problem may also be viewed as a shortest path problem with the length of each arc  $(i, j)$  being  $-t_{ij}$ . In particular, finding the duration of the project is equivalent to finding the shortest path from 1 to  $N$ . This path is also called a *critical* path. It can be seen that a delay by a given amount in the completion of one of the activities on the critical path will delay the completion of the overall project by the same amount. Note that because the network is acyclic, there can be only a finite number of paths from 1 to any  $i$ , so that at least one of these paths corresponds to the maximal path duration  $T_i$ .

Let us denote by  $S_1$  the set of phases that do not depend on completion of any other phase, and more generally, for  $k = 1, 2, \dots$ , let  $S_k$  be the set

$$S_k = \{i \mid \text{all paths from } 1 \text{ to } i \text{ have } k \text{ arcs or less}\},$$

with  $S_0 = \{1\}$ . The sets  $S_k$  can be viewed as the state spaces for the equivalent DP problem. Using maximization in place of minimization while changing the sign of the arc lengths, the DP algorithm can be written as

$$T_i = \max_{\substack{(j,i) \text{ such that} \\ j \in S_{k-1}}} [t_{ji} + T_j], \quad \text{for all } i \in S_k \text{ with } i \notin S_{k-1}.$$

Note that this is a forward algorithm; that is, it starts at the origin 1 and proceeds towards the destination  $N$ . An alternative backward algorithm, which starts at  $N$  and proceeds towards 1, is also possible, as discussed in the preceding section.

As an example, for the activity network of Fig. 2.2.1, we have

$$S_0 = \{1\}, \quad S_1 = \{1, 2\}, \quad S_2 = \{1, 2, 3\},$$

$$S_3 = \{1, 2, 3, 4\}, \quad S_4 = \{1, 2, 3, 4, 5\}.$$

A calculation using the preceding formula yields

$$T_1 = 0, \quad T_2 = 3, \quad T_3 = 4, \quad T_4 = 6, \quad T_5 = 10.$$

The critical path is  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ .

### 2.2.2 Hidden Markov Models and the Viterbi Algorithm

Consider a Markov chain with a finite number of states and given state transition probabilities  $p_{ij}$ . Suppose that when a transition occurs, the states corresponding to the transition are unknown (or “hidden”) to us, but instead we obtain an observation that relates to that transition. Given a sequence of observations, we want to estimate in some optimal sense the sequence of corresponding transitions. We are given the probability  $r(z; i, j)$  of an observation taking value  $z$  when the state transition is from  $i$  to  $j$ . We assume independent observations; that is, an observation depends only on its corresponding transition and not on other transitions. We are also given the probability  $\pi_i$  that the initial state takes value  $i$ . The probabilities  $p_{ij}$  and  $r(z; i, j)$  are assumed to be independent of time for notational convenience. The methodology to be described admits a straightforward extension to the case of time-varying system and observation probabilities.

Markov chains whose state transitions are imperfectly observed according to the probabilistic mechanism just described are called *Hidden Markov Models (HMMs for short)* or *partially observable Markov chains*.

In Chapter 5 we will discuss the control of such Markov chains in the context of stochastic optimal control problems with imperfect state information. In the present section, we will focus on the problem of estimating the state sequence given a corresponding observation sequence. This is an important problem that arises in a broad variety of practical contexts.

We use a “most likely state” estimation criterion, whereby given the observation sequence  $Z_N = \{z_1, z_2, \dots, z_N\}$ , we adopt as our estimate the state transition sequence  $\hat{X}_N = \{\hat{x}_0, \hat{x}_1, \dots, \hat{x}_N\}$  that maximizes over all  $X_N = \{x_0, x_1, \dots, x_N\}$  the conditional probability  $p(X_N \mid Z_N)$ . We will show that  $\hat{X}_N$  can be found by solving a special type of shortest path problem that involves an acyclic graph.

We have

$$p(X_N \mid Z_N) = \frac{p(X_N, Z_N)}{p(Z_N)},$$

where  $p(X_N, Z_N)$  and  $p(Z_N)$  are the unconditional probabilities of occurrence of  $(X_N, Z_N)$  and  $Z_N$ , respectively. Since  $p(Z_N)$  is a positive constant once  $Z_N$  is known, we can maximize  $p(X_N, Z_N)$  in place of  $p(X_N \mid Z_N)$ . The probability  $p(X_N, Z_N)$  can be written as

$$\begin{aligned} p(X_N, Z_N) &= p(x_0, x_1, \dots, x_N, z_1, z_2, \dots, z_N) \\ &= \pi_{x_0} p(x_1, \dots, x_N, z_1, z_2, \dots, z_N \mid x_0) \\ &= \pi_{x_0} p(x_1, z_1 \mid x_0) p(x_2, \dots, x_N, z_2, \dots, z_N \mid x_0, x_1, z_1) \\ &= \pi_{x_0} p_{x_0 x_1} r(z_1; x_0, x_1) p(x_2, \dots, x_N, z_2, \dots, z_N \mid x_0, x_1, z_1). \end{aligned}$$

This calculation can be continued by writing

$$\begin{aligned} p(x_2, \dots, x_N, z_2, \dots, z_N \mid x_0, x_1, z_1) &= p(x_2, z_2 \mid x_0, x_1, z_1) p(x_3, \dots, x_N, z_3, \dots, z_N \mid x_0, x_1, z_1, x_2, z_2) \\ &= p_{x_1 x_2} r(z_2; x_1, x_2) p(x_3, \dots, x_N, z_3, \dots, z_N \mid x_0, x_1, z_1, x_2, z_2), \end{aligned}$$

where for the last equation, we used the independence of the observations, i.e.,  $p(z_2 \mid x_0, x_1, x_2, z_1) = r(z_2; x_1, x_2)$ . Combining the above two relations,

$$\begin{aligned} p(X_N, Z_N) &= \pi_{x_0} p_{x_0 x_1} r(z_1; x_0, x_1) p_{x_1 x_2} r(z_2; x_1, x_2) \\ &\quad \cdot p(x_3, \dots, x_N, z_3, \dots, z_N \mid x_0, x_1, z_1, x_2, z_2), \end{aligned}$$

and continuing in the same manner, we obtain

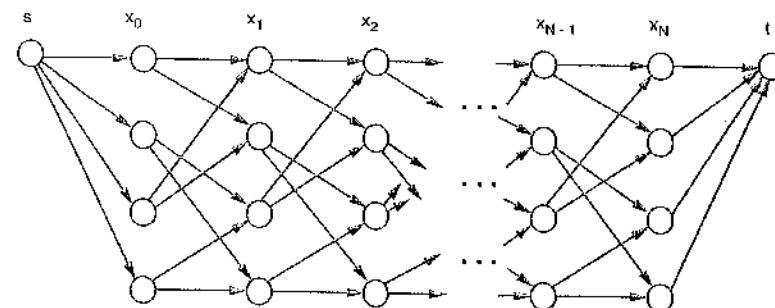
$$p(X_N, Z_N) = \pi_{x_0} \prod_{k=1}^N p_{x_{k-1} x_k} r(z_k; x_{k-1}, x_k). \quad (2.5)$$

We now show how the maximization of the above expression can be viewed as a shortest path problem. In particular, we construct a graph of state-time pairs, called the *trellis diagram*, by concatenating  $N+1$  copies

of the state space, and by preceding and following them with dummy nodes  $s$  and  $t$ , respectively, as shown in Fig. 2.2.2. The nodes of the  $k$ th copy correspond to the states  $x_{k-1}$  at time  $k-1$ . An arc connects a node  $x_{k-1}$  of the  $k$ th copy with a node  $x_k$  of the  $(k+1)$ st copy if the corresponding transition probability  $p_{x_{k-1}x_k}$  is positive. Since maximizing a positive cost function is equivalent to maximizing its logarithm, we see from Eq. (2.5) that, given the observation sequence  $Z_N = \{z_1, z_2, \dots, z_N\}$ , the problem of maximizing  $p(X_N, Z_N)$  is equivalent to the problem

$$\begin{aligned} & \text{minimize } -\ln(\pi_{x_0}) - \sum_{k=1}^N \ln(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k)) \\ & \text{over all possible sequences } \{x_0, x_1, \dots, x_N\}. \end{aligned}$$

By assigning to an arc  $(s, x_0)$  the length  $-\ln(\pi_{x_0})$ , to an arc  $(x_N, t)$  the length 0, and to an arc  $(x_{k-1}, x_k)$  the length  $-\ln(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k))$ , we see that the above minimization problem is equivalent to the problem of finding the shortest path from  $s$  to  $t$  in the trellis diagram. This shortest path defines the estimated state sequence  $\{\hat{x}_0, \hat{x}_1, \dots, \hat{x}_N\}$ .



**Figure 2.2.2** State estimation of an HMM viewed as a problem of finding a shortest path from  $s$  to  $t$ . Length of arcs from  $s$  to states  $x_0$  is  $-\ln(\pi_{x_0})$ , and length of arcs from states  $x_N$  to  $t$  is zero. Length of an arc from a state  $x_{k-1}$  to  $x_k$  is  $-\ln(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k))$ , where  $z_k$  is the  $k$ th observation.

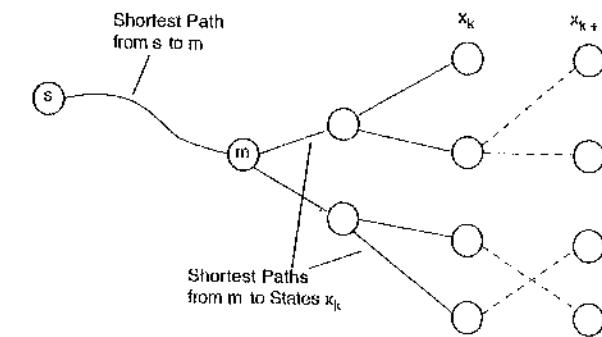
In practice, the shortest path is most conveniently constructed sequentially by forward DP, that is, by first calculating the shortest distance from  $s$  to each node  $x_1$ , then using these distances to calculate the shortest distance from  $s$  to each node  $x_2$ , etc. In particular, suppose that we have computed the shortest distances  $D_k(x_k)$  from  $s$  to all states  $x_k$  on the basis of the observation sequence  $Z_k$ , and suppose that the new observation  $z_{k+1}$  is obtained. Then the shortest distances  $D_{k+1}(x_{k+1})$  from  $s$  to any state

$x_{k+1}$  can be computed by the DP recursion

$$D_{k+1}(x_{k+1}) = \min_{\substack{\text{all } x_k \text{ such that} \\ p_{x_k x_{k+1}} > 0}} [D_k(x_k) - \ln(p_{x_k x_{k+1}} r(z_{k+1}; x_k, x_{k+1}))].$$

The initial condition is  $D_0(x_0) = -\ln(\pi_{x_0})$ . The final estimated state sequence  $\hat{X}_N$  corresponds to the shortest path from  $s$  to the final state  $\hat{x}_N$  that minimizes  $D_N(x_N)$  over the finite set of possible states  $x_N$ . An advantage of this procedure is that it can be executed in real time, as soon as each new observation is obtained.

There are a number of practical schemes for estimating a portion of the state sequence without waiting to receive the entire observation sequence  $Z_N$ , and this is useful if  $Z_N$  is a long sequence. For example, one can check fairly easily whether for some  $k$ , all shortest paths from  $s$  to states  $x_k$  pass through a single node in the subgraph of states  $x_0, \dots, x_{k-1}$ . If so, it can be seen from Fig. 2.2.3 that the shortest path from  $s$  to that node will not be affected by reception of additional observations, and therefore the subsequence of state estimates up to that node can be determined without waiting for the remaining observations.



**Figure 2.2.3** Estimating a portion of the state sequence prior to receiving the entire observation sequence. Suppose that the shortest paths from  $s$  to all states  $x_k$  pass through a single node  $m$ . If an additional observation is received, the shortest paths from  $s$  to all states  $x_{k+1}$  will continue to pass through  $m$ . Therefore, the portion of the state sequence up to node  $m$  can be safely estimated because additional observations will not change the initial portion of the shortest paths from  $s$  up to  $m$ .

The shortest path-based estimation procedure just described is known as the *Viterbi algorithm*, and finds numerous applications in a variety of contexts. An example is *speech recognition*, where the basic goal is to transcribe a spoken word sequence in terms of elementary speech units called *phonemes*. One possibility is to associate the states of the HMM with

phonemes, and given a sequence of recorded phonemes  $Z_N = \{z_1, \dots, z_N\}$ , to find a phonemic sequence  $\hat{X}_N = \{\hat{x}_1, \dots, \hat{x}_N\}$  that maximizes over all  $X_N = \{x_1, \dots, x_N\}$  the conditional probability  $p(X_N | Z_N)$ . The probabilities  $r(z_k; x_{k-1}, x_k)$  and  $p_{x_{k-1}x_k}$  can be experimentally obtained, if necessary by specialized “training” for each speaker that uses the speech recognition system. The Viterbi algorithm can then be used to find the most likely phonemic sequence. There are also other HMMs used for word and sentence recognition, where only phonemic sequences that constitute words from a given dictionary are considered. We refer the reader to Rabiner [Rab89] and Picone [Pic90] for a general review of HMMs applied to speech recognition and for further references to related work. It is also possible to use similar models for computerized recognition of handwriting.

The Viterbi algorithm was originally developed as a scheme for decoding data after transmission over a noisy communication channel. The following example describes this context in some detail.

### Example 2.2.1 (Convolutional Coding and Decoding)

When binary data are transmitted over a noisy communication channel, it is often essential to use coding as a means of enhancing reliability of communication. A common type of coding method, called *convolutional coding*, converts a source-generated binary data sequence

$$\{w_1, w_2, \dots\}, \quad w_k \in \{0, 1\}, \quad k = 1, 2, \dots,$$

into a coded sequence  $\{y_1, y_2, \dots\}$ , where each  $y_k$  is an  $n$ -dimensional vector with binary coordinates, called *codeword*,

$$y_k = \begin{pmatrix} y_k^1 \\ \vdots \\ y_k^n \end{pmatrix}, \quad y_k^i \in \{0, 1\}, \quad i = 1, \dots, n, \quad k = 1, 2, \dots$$

The sequence  $\{y_1, y_2, \dots\}$  is then transmitted over a noisy channel and gets transformed into a sequence  $\{z_1, z_2, \dots\}$ , which is then decoded to yield the decoded data sequence  $\{\hat{w}_1, \hat{w}_2, \dots\}$ ; see Fig. 2.2.4. The objective is to design the encoder/decoder scheme so that the decoded sequence is as close to the original as possible.

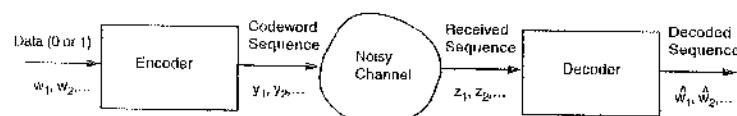


Figure 2.2.4 Encoder/decoder scheme.

The problem just discussed is central in information theory and can be approached in several different ways. In a particularly popular and effective technique called *convolutional coding*, the vectors  $y_k$  are related to  $w_k$  via equations of the form

$$y_k = Cx_{k-1} + dw_k, \quad k = 1, 2, \dots, \quad (2.6)$$

$$x_k = Ax_{k-1} + bw_k, \quad k = 1, 2, \dots, \quad x_0 : \text{given}, \quad (2.7)$$

where  $x_k$  is an  $m$ -dimensional vector with binary coordinates, which we view as state, and  $C$ ,  $d$ ,  $A$ , and  $b$  are  $n \times m$ ,  $n \times 1$ ,  $m \times m$ , and  $m \times 1$  matrices, respectively, with binary coordinates. The products and the sums involved in the expressions  $Cx_{k-1} + dw_k$  and  $Ax_{k-1} + bw_k$  are calculated using modulo 2 arithmetic.

As an example, let  $m = 2$ ,  $n = 3$ , and

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad d = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Then the evolution of the system (2.6)-(2.7) can be represented by the diagram shown in Fig. 2.2.5. Given the initial  $x_0$ , this diagram can be used to generate the codeword sequence  $\{y_1, y_2, \dots\}$  corresponding to a data sequence  $\{w_1, w_2, \dots\}$ . For example, when the initial state is  $x_0 = 00$ , the data sequence

$$\{w_1, w_2, w_3, w_4\} = \{1, 0, 0, 1\}$$

generates the state sequence

$$\{x_0, x_1, x_2, x_3, x_4\} = \{00, 01, 11, 10, 00\},$$

and the codeword sequence

$$\{y_1, y_2, y_3, y_4\} = \{111, 011, 111, 011\}.$$

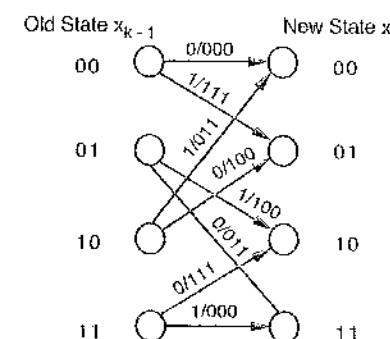


Figure 2.2.5 State transition diagram for convolutional coding. The binary number pair on each arc is the data/codeword pair  $w_k/y_k$  for the corresponding transition. So for example, when  $x_{k-1} = 01$ , a zero data bit ( $w_k = 0$ ) effects a transition to  $x_k = 11$  and generates the codeword 011.

Assume now that the characteristics of the noisy transmission channel are such that a codeword  $y$  is actually received as  $z$  with known probability  $p(z|y)$ , where  $z$  is any  $n$ -bit binary number. We assume independent errors so that

$$p(Z_N | Y_N) = \prod_{k=1}^N p(z_k | y_k), \quad (2.8)$$

where  $Z_N = \{z_1, \dots, z_N\}$  is the received sequence and  $Y_N = \{y_1, \dots, y_N\}$  is the transmitted sequence. By associating the codewords  $y$  with state transitions, we formulate a maximum likelihood estimation problem, whereby we want to find a sequence  $\hat{Y}_N = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N\}$  such that

$$p(Z_N | \hat{Y}_N) = \max_{Y_N} p(Z_N | Y_N).$$

The constraint on  $Y_N$  is that it must be a feasible codeword sequence (i.e., it must correspond to some initial state and data sequence, or equivalently, to a sequence of arcs of the trellis diagram).

Let us construct a trellis diagram by concatenating  $N$  state transition diagrams and appending dummy nodes  $s$  and  $t$  on its left and right sides, which are connected with zero-length arcs to the states  $x_0$  and  $x_N$ , respectively. By using Eq. (2.8), we see that, given the received sequence  $Z_N = \{z_1, z_2, \dots, z_N\}$ , the problem of maximizing  $p(Z_N | Y_N)$  is equivalent to the problem

$$\text{minimize} \sum_{k=1}^N -\ln(p(z_k | y_k))$$

over all binary sequences  $\{y_1, y_2, \dots, y_N\}$ .

This is equivalent to a problem of finding a shortest path in the trellis diagram from  $s$  to  $t$ , where the length of the arc associated with the codeword  $y_k$  is  $-\ln(p(z_k | y_k))$ , and the lengths of each arc incident to a dummy node is zero. From the shortest path and the trellis diagram, we can then obtain the corresponding data sequence  $\{\hat{w}_1, \dots, \hat{w}_N\}$ , which is accepted as the decoded data.

The maximum likelihood estimate  $\hat{Y}_N$  can be found by solving the corresponding shortest path problem using the Viterbi algorithm. In particular, the shortest distances  $D_{k+1}(x_{k+1})$  from  $s$  to any state  $x_{k+1}$  are computed by the DP recursion

$$D_{k+1}(x_{k+1}) = \min_{\substack{\text{all } x_k \text{ such that} \\ (x_k, x_{k+1}) \text{ is an arc}}} [D_k(x_k) - \ln(p(z_{k+1} | y_{k+1}))],$$

where  $y_{k+1}$  is the codeword corresponding to the arc  $(x_k, x_{k+1})$ . The final state  $\hat{x}_N$  on the shortest path is the one that minimizes  $D_N(x_N)$  over  $x_N$ .

### 2.3 SHORTEST PATH ALGORITHMS

We have seen that shortest path problems and deterministic finite-state optimal control problems are equivalent. The computational implications of this are twofold.

- (a) One can use DP to solve general shortest path problems. Note that there are several other shortest path methods, some of which have superior theoretical worst-case performance to DP. However, DP is often preferred in practice, particularly for problems with an acyclic graph structure and also when a parallel computer is available.
- (b) One can use general shortest path methods (other than DP) for deterministic finite-state optimal control problems. In most cases, DP is preferable to other shortest path methods, because it is tailored to the sequential nature of optimal control problems. However, there are important cases where other shortest path methods are preferable.

In this section we discuss several alternative shortest path methods. We motivate these methods by focusing on shortest path problems with a very large number of nodes. Suppose that there is only one origin and only one destination, as in shortest path problems arising from deterministic optimal control (cf. Fig. 2.1.1). Then it is often true that most of the nodes are not relevant to the shortest path problem in the sense that they are unlikely candidates for inclusion in a shortest path between the given origin and destination. Unfortunately, however, in the DP algorithm every node and arc will participate in the computation, so there arises the possibility of other more efficient methods.

A similar situation arises in some search problems that are common in artificial intelligence and combinatorial optimization. Generally, these problems involve decisions that can be broken down into stages. With proper reformulation, the decision stages can be made to correspond to arc selections in a shortest path problem, or to the stages of a DP algorithm. We provide some examples.

#### Example 2.3.1 (The Four Queens Problem)

Four queens must be placed on a  $4 \times 4$  portion of a chessboard so that no queen can attack another. In other words, the placement must be such that every row, column, or diagonal of the  $4 \times 4$  board contains at most one queen. Equivalently, we can view the problem as a sequence of problems; first, placing a queen in one of the first two squares in the top row, then placing another queen in the second row so that it is not attacked by the first, and similarly placing the third and fourth queens. (It is sufficient to consider only the first two squares of the top row, since the other two squares lead to symmetric positions.) We can associate positions with nodes of an acyclic graph where the root node  $s$  corresponds to the position with no queens and the terminal nodes correspond to the positions where no additional queens can be placed

without some queen attacking another. Let us connect each terminal position with an artificial node  $t$  by means of an arc. Let us also assign to all arcs length zero except for the artificial arcs connecting terminal positions with less than four queens with the artificial node  $t$ . These latter arcs are assigned the length  $\infty$  (see Fig. 2.3.1) to express the fact that they correspond to dead-end positions that cannot lead to a solution. Then, the four queens problem reduces to finding a shortest path from node  $s$  to node  $t$ .

Note that once the nodes of the graph are enumerated the problem is essentially solved. In this  $4 \times 4$  problem the number of nodes is small. However, we can think of similar problems with much larger memory requirements. For example, there is an eight queens problem where the board is  $8 \times 8$  instead of  $4 \times 4$ .

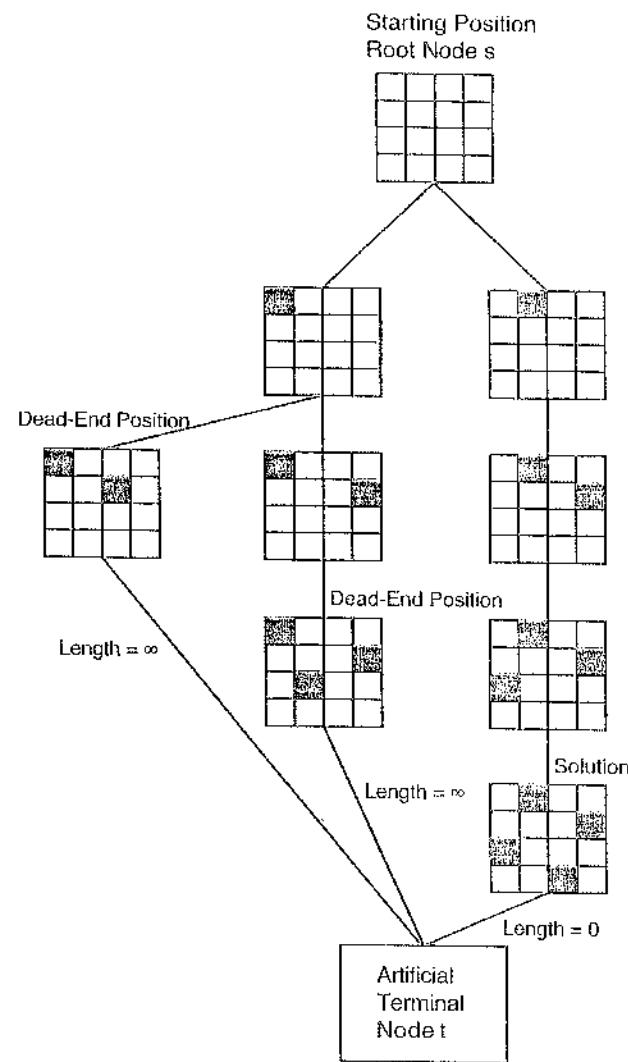
### Example 2.3.2 (The Traveling Salesman Problem)

An important model for scheduling a sequence of operations is the classical traveling salesman problem. Here we are given  $N$  cities and the mileage between each pair of cities. We wish to find a minimum-mileage trip that visits each of the cities exactly once and returns to the origin node. To convert this problem to a shortest path problem, we associate a node with every sequence of  $n$  distinct cities, where  $n \leq N$ . The construction and arc lengths of the corresponding graph are explained by means of an example in Fig. 2.3.2. The origin node  $s$  consists of city A, taken as the start. A sequence of  $n$  cities ( $n < N$ ) yields a sequence of  $(n+1)$  cities by adding a new city. Two such sequences are connected by an arc with length equal to the mileage between the last two of the  $n+1$  cities. Each sequence of  $N$  cities is connected to an artificial terminal node  $t$  with an arc having length equal to the distance from the last city of the sequence to the starting city A. Note that the number of nodes grows exponentially with the number of cities.

In the shortest path problem that we will consider in this section, there is a special node  $s$ , called the *origin*, and a special node  $t$ , called the *destination*. We will assume a single destination, but the methods to be discussed admit extensions to the case of multiple destinations (see Exercise 2.6). A node  $j$  is called a *child* of node  $i$  if there is an arc  $(i, j)$  connecting  $i$  with  $j$ . The length of arc  $(i, j)$  is denoted by  $a_{ij}$  and we assume that *all arcs have nonnegative length*. Exercise 2.7 deals with the case where all cycle lengths (rather than arc lengths) are assumed nonnegative. We wish to find a shortest path from origin to destination.

#### 2.3.1 Label Correcting Methods

We now discuss a general type of shortest path algorithm. The idea is to progressively discover shorter paths from the origin to every other node  $i$ , and to maintain the length of the shortest path found so far in a variable  $d_i$  called the *label* of  $i$ . Each time  $d_i$  is reduced following the discovery of a



**Figure 2.3.1** Shortest path formulation of the four queens problem. Symmetric positions resulting from placing a queen in one of the rightmost squares in the top row have been ignored. Squares containing a queen have been darkened. All arcs have length zero except for those connecting dead-end positions to the artificial terminal node.

shorter path to  $i$ , the algorithm checks to see if the labels  $d_j$  of the children  $j$  of  $i$  can be “corrected,” that is, they can be reduced by setting them to

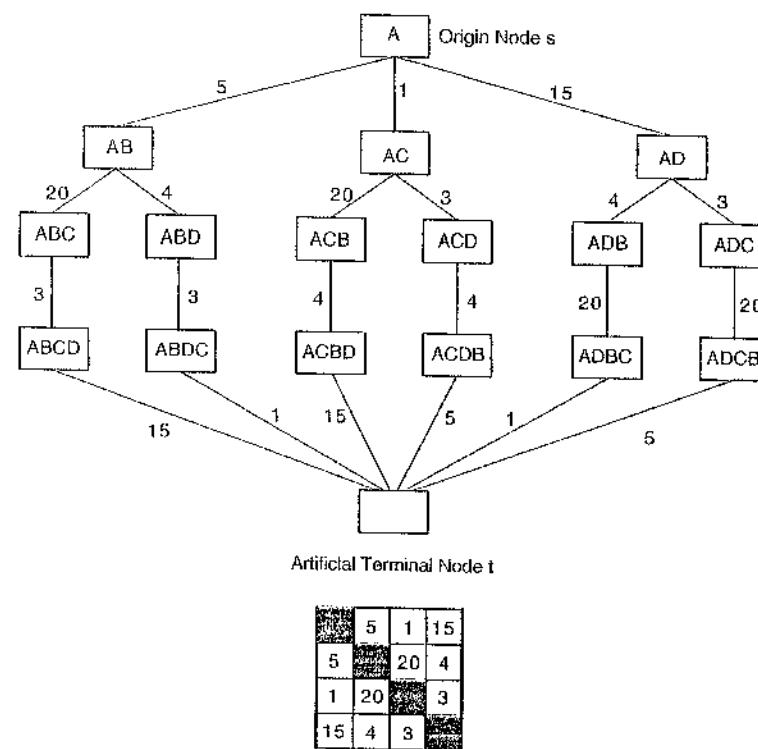


Figure 2.3.2 Example of a shortest path formulation of the traveling salesman problem. The distance between the four cities A, B, C, and D are shown in the table. The arc lengths are shown next to the arcs.

$d_i + a_{ij}$  [the length of the shortest path to  $i$  found thus far followed by arc  $(i, j)$ ]. The label  $d_t$  of the destination is maintained in a variable called UPPER, which plays a special role in the algorithm. The label  $d_s$  of the origin is initialized at 0 and remains at 0 throughout the algorithm. The labels of all other nodes are initialized at  $\infty$ , i.e.,  $d_i = \infty$  for all  $i \neq s$ .

The algorithm also makes use of a list of nodes called OPEN (another name frequently used is *candidate list*). The list OPEN contains nodes that are currently active in the sense that they are candidates for further examination by the algorithm and possible inclusion in the shortest path. Initially, OPEN contains just the origin node  $s$ . Each node that has entered OPEN at least once, except  $s$ , is assigned a “parent,” which is some other node. The parent nodes are not necessary for the computation of the shortest distance; they are needed for tracing the shortest path to the origin after the algorithm terminates. The steps of the algorithm are as follows (see also Fig. 2.3.3):

### Label Correcting Algorithm

**Step 1:** Remove a node  $i$  from OPEN and for each child  $j$  of  $i$ , execute step 2.

**Step 2:** If  $d_i + a_{ij} < \min\{d_j, \text{UPPER}\}$ , set  $d_j = d_i + a_{ij}$  and set  $i$  to be the parent of  $j$ . In addition, if  $j \neq t$ , place  $j$  in OPEN if it is not already in OPEN, while if  $j = t$ , set UPPER to the new value  $d_i + a_{it}$  of  $d_t$ .

**Step 3:** If OPEN is empty, terminate; else go to step 1.

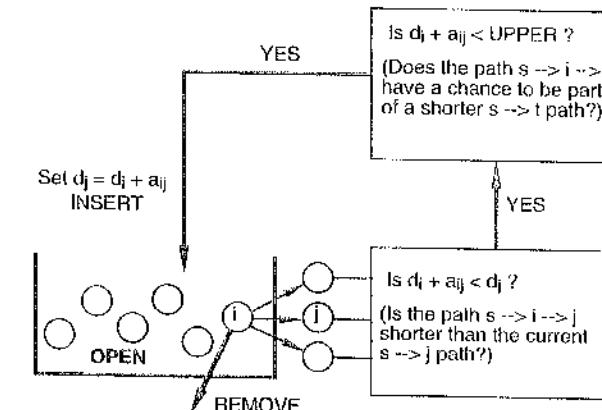


Figure 2.3.3 Diagrammatic illustration of the label correcting algorithm, with an interpretation of the tests for insertion of a node into the OPEN list.

It can be seen by induction that, throughout the algorithm,  $d_j$  is either  $\infty$  (if node  $j$  has not yet entered the OPEN list), or else it is the length of some path from  $s$  to  $j$  consisting of nodes that have entered the OPEN list at least once. In the latter case, the path can be constructed by tracing backward the parent nodes starting with the parent of node  $j$ . Furthermore, UPPER is either  $\infty$ , or else it is the length of some path from  $s$  to  $t$ , and consequently it is an upper bound of the shortest distance from  $s$  to  $t$ . The idea in the algorithm is that when a path from  $s$  to  $j$  is discovered, which is shorter than those considered earlier ( $d_i + a_{ij} < d_j$  in step 2), the value of  $d_j$  is accordingly reduced, and node  $j$  enters the OPEN list so that paths passing through  $j$  and reaching the children of  $j$  can be taken into account. It makes sense to do so, however, only when the path considered has a chance of leading to a path from  $s$  to  $t$  with length smaller than the upper bound UPPER of the shortest distance from  $s$  to  $t$ .

In view of the nonnegativity of the arc lengths, this is possible only if the path length  $d_i + a_{ij}$  is smaller than UPPER. This provides the rationale for entering  $j$  into OPEN in step 2 only if  $d_i + a_{ij} < \text{UPPER}$  (see Fig. 2.3.3).

Tracing the steps of the algorithm, we see that it will first remove node  $s$  from OPEN and sequentially examine its children. If  $t$  is not a child of  $s$ , the algorithm will place all children  $j$  of  $s$  in OPEN after setting  $d_j = a_{sj}$ . If  $t$  is a child of  $s$ , then the algorithm will place all children  $j$  of  $s$  examined before  $t$  in OPEN and will set their labels to  $a_{sj}$ ; then it will examine  $t$  and set UPPER to  $a_{st}$ ; finally, it will place each of the remaining children  $j$  of  $s$  in OPEN only if  $a_{sj}$  is less than the current value of UPPER, which is  $a_{st}$ . The algorithm will subsequently take a child  $i \neq t$  of  $s$  from OPEN, and sequentially place in OPEN those of its children  $j \neq i$  that satisfy the criterion of step 2, etc. Note that the origin  $s$  can never reenter OPEN because  $d_s$  cannot be reduced from its initial value of zero. Also, by the rules of the algorithm, the destination can never enter OPEN. When the algorithm terminates, we will show shortly that a shortest path can be obtained by tracing backwards the parent nodes starting from  $t$  and going towards  $s$ . Figure 2.3.4 illustrates the use of the algorithm to solve the traveling salesman problem of Fig. 2.3.2.

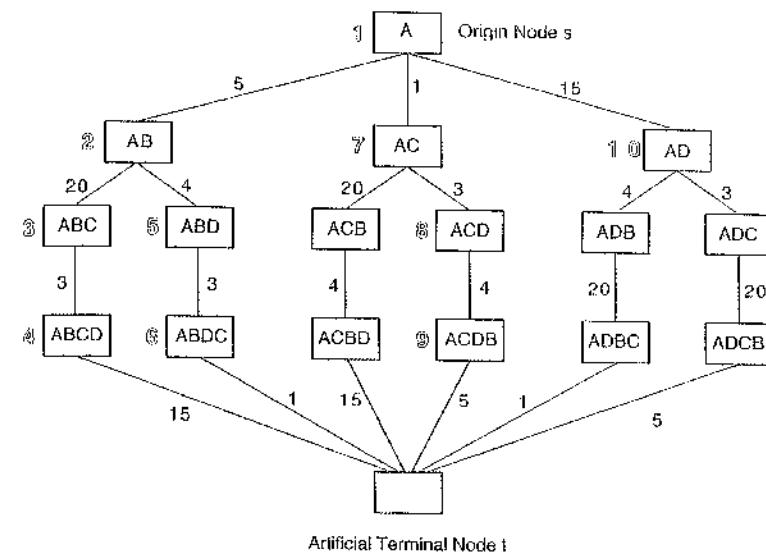
The following proposition establishes the validity of the algorithm.

**Proposition 2.3.1:** If there exists at least one path from the origin to the destination, the label correcting algorithm terminates with UPPER equal to the shortest distance from the origin to the destination. Otherwise the algorithm terminates with UPPER =  $\infty$ .

**Proof:** We first show that the algorithm will terminate. Indeed, each time a node  $j$  enters the OPEN list, its label is decreased and becomes equal to the length of some path from  $s$  to  $j$ . On the other hand, the number of distinct lengths of paths from  $s$  to  $j$  that are smaller than any given number is finite. The reason is that each path can be decomposed into a path with no repeated nodes (there is a finite number of distinct such paths), plus a (possibly empty) set of cycles, each having a nonnegative length. Therefore, there can be only a finite number of label reductions, implying that the algorithm will terminate.

Suppose that there is no path from  $s$  to  $t$ . Then a node  $i$  such that  $(i, t)$  is an arc cannot enter the OPEN list, because as argued earlier, this would establish that there is a path from  $s$  to  $i$ , and therefore also a path from  $s$  to  $t$ . Thus, based on the rules of the algorithm, UPPER can never be reduced from its initial value of  $\infty$ .

Suppose now that there is a path from  $s$  to  $t$ . Then, since there is a finite number of distinct lengths of paths from  $s$  to  $t$  that are smaller than any given number, there is also a shortest path. Let  $(s, j_1, j_2, \dots, j_k, t)$



Iter. No.	Node Exiting OPEN	OPEN at the End of Iteration	UPPER
0	-	1	$\infty$
1	1	2, 7, 10	$\infty$
2	2	3, 5, 7, 10	$\infty$
3	3	4, 5, 7, 10	$\infty$
4	4	5, 7, 10	43
5	5	6, 7, 10	43
6	6	7, 10	13
7	7	8, 10	13
8	8	9, 10	13
9	9	10	13
10	10	Empty	13

Figure 2.3.4 The algorithm applied to the traveling salesman problem of Fig. 2.3.2. The optimal solution ABDC is found after examining nodes 1 through 10 in the figure in that order. The table shows the successive contents of the OPEN list.

be a shortest path and let  $d^*$  be the corresponding shortest distance. We will show that the value of UPPER upon termination must be equal to  $d^*$ . Indeed, each subpath  $(s, j_1, j_2, \dots, j_m)$ ,  $m = 1, \dots, k$ , of the shortest path  $(s, j_1, j_2, \dots, j_k, t)$  must be a shortest path from  $s$  to  $j_m$ . If the value of

UPPER is larger than  $d^*$  at termination, the same must be true throughout the algorithm, and therefore UPPER will also be larger than the length of all the paths  $(s, j_1, \dots, j_m)$ ,  $m = 1, \dots, k$ , throughout the algorithm, in view of the nonnegative arc length assumption. It follows that node  $j_k$  will never enter the OPEN list with  $d_{j_k}$  equal to the shortest distance from  $s$  to  $j_k$ , since in this case UPPER would be set to  $d^*$  in step 2 immediately following the next time node  $j_k$  is examined by the algorithm in step 2. Similarly, and using also the nonnegative length assumption, this means that node  $j_{k-1}$  will never enter the OPEN list with  $d_{j_{k-1}}$  equal to the shortest distance from  $s$  to  $j_{k-1}$ . Proceeding backward, we conclude that  $j_1$  never enters the OPEN list with  $d_{j_1}$  equal to the shortest distance from  $s$  to  $j_1$  [which is equal to the length of the arc  $(s, j_1)$ ]. This happens, however, at the first iteration of the algorithm, obtaining a contradiction. It follows that at termination, UPPER will be equal to the shortest distance from  $s$  to  $t$ . Q.E.D.

From the preceding proof, it can also be seen that, upon termination of the algorithm, the path constructed by tracing the parent nodes backward from  $t$  to  $s$  has length equal to UPPER, so it is a shortest path from  $s$  to  $t$ . Thus the algorithm yields not just the shortest distance but also a shortest path, provided that we keep track of the parent of each node that enters OPEN.

An important property of the algorithm is that nodes  $j$  for which  $d_i + a_{ij} \geq$  UPPER in step 2 will not enter OPEN in the current iteration, and may possibly not enter in any subsequent iteration. As a result the number of nodes that enter OPEN may be much smaller than the total number of nodes. Furthermore, if a good lower bound to the shortest distance from  $s$  to  $t$  (or the shortest distance itself) is known, the computation can be terminated once UPPER reaches that bound within an acceptable tolerance. This is useful, for example, in the four queens problem, where the shortest distance is known to be zero or infinity. Then the algorithm will terminate once UPPER= 0, when a solution is found for the first time.

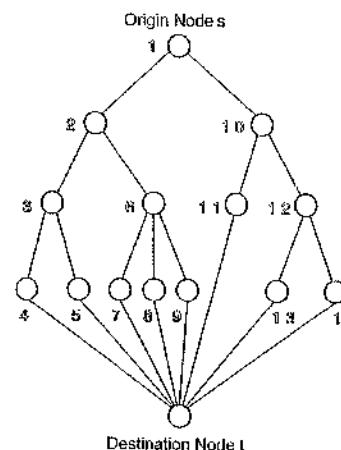
### Specific Label Correcting Methods

There is considerable freedom in selecting the node to be removed from OPEN at each iteration. This gives rise to several different methods. The following are some of the most important (the author's textbooks on network optimization [Ber91a], [Ber98a] contain a fuller account of label correcting methods and their analysis; [Ber91a] contains several computer code implementations).

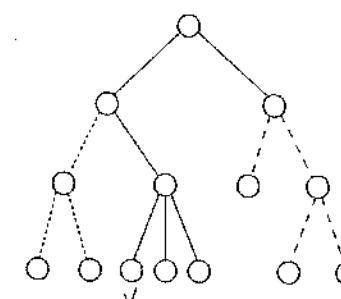
- (a) *Breadth-first search*, also known as the *Bellman-Ford* method, which adopts a first-in/first-out policy; that is, the node is always removed from the top of OPEN and each node entering OPEN is placed at the

bottom of OPEN. (Here and in the following methods except (c), we assume that OPEN is structured as a queue.)

- (b) *Depth-first search*, which adopts a last-in/first-out policy; that is, the node is always removed from the top of OPEN and each node entering OPEN is placed at the top of OPEN. One motivation for this method is that it often requires relatively little memory. For example, suppose that the graph has a tree-like structure whereby there is a unique path from the origin node to every node other than the destination as shown in Fig. 2.3.5. Then the nodes will enter OPEN only once and in the order shown in Fig. 2.3.5. At any one time, it is only necessary to store a small portion of the graph as shown in Fig. 2.3.6.



**Figure 2.3.5** Searching a tree in depth-first fashion. The numbers next to the nodes indicate the order in which nodes exit the OPEN list.



**Figure 2.3.6** Memory requirements of depth-first search for the graph of Fig. 2.3.5. At the time the node marked by the checkmark exits the OPEN list, only the solid-line portion of the tree is needed in memory. The dotted-line portion has been generated and purged from memory based on the rule that for a graph where there is only one path from the origin to every node other than the destination, it is unnecessary to store a node once all of its successors are out of the OPEN list. The broken-line portion of the tree has not yet been generated.

- (c) *Best-first search*, which at each iteration removes from OPEN a node with minimum value of label, i.e., a node  $i$  with

$$d_i = \min_{j \text{ in OPEN}} d_j.$$

This method, also known as *Dijkstra's method* or *label setting method*, has a particularly interesting property. It can be shown that in this method, a node will enter the OPEN list at most once (see Exercise 2.4). The drawback of this method is the overhead required to find at each iteration the node of OPEN that has minimum label. Several sophisticated methods have been developed to carry out this operation efficiently (see e.g., Bertsekas [Ber98a]).

- (d) *D'Esopo-Pape method*, which at each iteration removes the node that is at the top of OPEN, but inserts a node at the top of OPEN if it has already been in OPEN earlier, and inserts the node at the bottom of OPEN otherwise.
- (e) *Small-Label-First (SLF) method*, which at each iteration removes the node that is at the top of OPEN, but inserts a node  $i$  at the top of OPEN if its label  $d_i$  is less than or equal to the label  $d_j$  of the top node  $j$  of OPEN; otherwise it inserts  $i$  at the bottom of OPEN. This is a low-overhead approximation to the best-first search strategy. As a complement to the SLF strategy, one may also try to avoid removing nodes with relatively large labels from OPEN using the following device, known as the *Large-Label-Last (LLL) strategy*: at the start of an iteration, the top node of OPEN is compared with the average of the labels of the nodes in OPEN and if it is larger, the top node is placed at the bottom of OPEN and the new top node of OPEN is similarly tested against the average of the labels. In this way the removal of nodes with relatively large labels is postponed in favor of nodes with relatively small labels. The extra overhead required in this method is small: maintain the sum of the labels and the number of nodes in OPEN. When starting a new iteration, the ratio of these numbers gives the desired average. There are also several other methods, which are based on the idea of examining nodes with small labels first (see Bertsekas [Ber93], [Ber98a], and Bertsekas, Guerriero, and Musmanno [BGM96] for detailed descriptions and computational studies).

Generally, it appears that for nonnegative arc lengths, the number of iterations is reduced as the method is more successful in removing from OPEN nodes with a relatively small label. For a supporting heuristic argument, note that for a node  $j$  to reenter OPEN, some node  $i$  such that  $d_i + a_{ij} < d_j$  must first exit OPEN. Thus, the smaller  $d_j$  was at the previous exit of  $j$  from OPEN, the less likely it is that  $d_i + a_{ij}$  will subsequently become less than  $d_j$  for some node  $i$  in OPEN and arc  $(i, j)$ . In particular,

if  $d_j \leq \min_{i \in \text{OPEN}} d_i$ , it is impossible that subsequent to the exit of  $j$  from OPEN we will have  $d_i + a_{ij} < d_j$  for some  $i$  in OPEN, since the arc lengths  $a_{ij}$  are nonnegative. The SLF and other related but more sophisticated methods, often require a number of iterations, which is close to the minimum (the one required by the best-first method). However, they can be much faster than the best-first method, because they require much less overhead for determining the node to be removed from OPEN.

### 2.3.2 Label Correcting Variations - $A^*$ Algorithm

The generic label correcting algorithm need not be started with the initial conditions  $d_s = 0$  and  $d_i = \infty$  for  $i \neq s$  in order to work correctly. It can be shown, similar to Prop. 2.3.1, that one can use any set of initial labels  $d_i$  such that for each node  $i$ ,  $d_i$  is either  $\infty$  or else it is the length of some path from  $s$  to  $i$ . The scalar UPPER may be taken to be equal to  $d_t$  and the initial OPEN list may be taken to be the set  $\{i \mid d_i < \infty\}$ .

This kind of initialization is very useful if, by using heuristics or a known solution of a similar shortest path problem, we can construct a "good" path

$$P = (s, i_1, \dots, i_k, t)$$

from  $s$  to  $t$ . Then we can initialize the algorithm with

$$d_i = \begin{cases} \text{Length of portion of path } P \text{ from } s \text{ to } i & \text{if } i \in P, \\ \infty & \text{if } i \notin P, \end{cases}$$

with UPPER equal to  $d_t$ , and with the OPEN list equal to  $\{s, i_1, \dots, i_k\}$ . If  $P$  is a near-optimal path and consequently the initial value of UPPER is near its final value, the test for future admissibility into the candidate list will be relatively tight from the start of the algorithm and many unnecessary entrances of nodes into OPEN may be saved. In particular, it can be seen that all nodes whose shortest distances from the origin are greater or equal to the length of  $P$  will never enter the candidate list.

Another possibility, known as the  *$A^*$  algorithm*, is to strengthen the test  $d_i + a_{ij} < \text{UPPER}$  that node  $j$  must pass before it can be placed in the OPEN list in step 2. This can be done if a *positive underestimate*  $h_j$  of the shortest distance of node  $j$  to the destination is available. Such an estimate can be obtained from special knowledge about the problem at hand. We may then speed up the computation substantially by placing a node  $j$  in OPEN in step 2 only when

$$d_i + a_{ij} + h_j < \text{UPPER}$$

(instead of  $d_i + a_{ij} < \text{UPPER}$ ). In this way, fewer nodes will potentially be placed in OPEN before termination. Using the fact that  $h_j$  is an underestimate of the true shortest distance from  $j$  to the destination, it can be seen

that nodes  $j$  such that  $d_i + a_{ij} + h_j \geq \text{UPPER}$  need not enter OPEN, and the argument given in the proof of Prop. 2.3.1 shows that the algorithm using the preceding test will terminate with a shortest path.

The  $A^*$  algorithm is just one way to sharpen the test  $d_i + a_{ij} < \text{UPPER}$  for admission of node  $j$  into the OPEN list. An alternative idea is to try to reduce the value of UPPER by obtaining for the node  $j$  in step 2 an *upper bound*  $m_j$  of the shortest distance from  $j$  to the destination  $t$  (for example the length of some path from  $j$  to  $t$ ). Then if  $d_j + m_j < \text{UPPER}$  after step 2, we can reduce UPPER to  $d_j + m_j$ , thereby making the test for future admissibility into OPEN more stringent. This idea is used in some versions of the branch-and-bound algorithm, one of which we now describe.

### 2.3.3 Branch-and-Bound

Consider a problem of minimizing a cost function  $f(x)$  over a *finite* set of feasible solutions  $X$ . We have in mind problems where the number of feasible solutions is very large, so an enumeration and comparison of these solutions is impractical, e.g., the traveling salesman problem of Example 2.3.2. The idea of the branch-and-bound method is to avoid a complete enumeration by discarding solutions that, based on certain tests, have no chance of being optimal. This is similar to label correcting methods, where based on various tests that use the value of UPPER and other data, the insertion in the OPEN list of some nodes is avoided. In fact, we will see that the branch-and-bound method can be viewed as a form of label correcting method.

The key idea of the branch-and-bound method is to partition the feasible set into smaller subsets, and then use certain bounds on the attainable cost within some of the subsets to eliminate from further consideration other subsets. The rationale for this is captured in the following simple observation.

#### Bounding Principle

Given the problem of minimizing  $f(x)$  over  $x \in X$ , and two subsets  $Y_1 \subset X$  and  $Y_2 \subset X$ , suppose that we have bounds

$$\underline{f}_1 \leq \min_{x \in Y_1} f(x), \quad \bar{f}_2 \geq \min_{x \in Y_2} f(x).$$

Then, if  $\bar{f}_2 \leq \underline{f}_1$ , the solutions in  $Y_1$  may be disregarded since their cost cannot be smaller than the cost of the best solution in  $Y_2$ .

The branch-and-bound method calculates suitable upper and lower bounds, and uses the bounding principle to eliminate from consideration substantial portions of the feasible set. To describe the method, we use

an acyclic graph with nodes that correspond on a one-to-one basis with a collection  $\mathcal{X}$  of subsets of the feasible set  $X$ . We require the following:

1.  $X \in \mathcal{X}$  (i.e., the set of all solutions is a node).
2. For each solution  $x$ , we have  $\{x\} \in \mathcal{X}$  (i.e., each solution viewed as a singleton set is a node).
3. Each set  $Y \in \mathcal{X}$  that contains more than one solution  $x \in Y$  is partitioned into sets  $Y_1, \dots, Y_n \in \mathcal{X}$  such that  $Y_i \neq Y$  for all  $i$ :

$$\bigcup_{i=1}^n Y_i = Y.$$

The set  $Y$  is called the *parent* of  $Y_1, \dots, Y_n$ , and the sets  $Y_1, \dots, Y_n$  are called the *children* of  $Y$ .

4. Each set in  $\mathcal{X}$  other than  $X$  has at least one parent.

The collection of sets  $\mathcal{X}$  defines an acyclic graph with root node the set of all feasible solutions  $X$  and terminal nodes the singleton solutions  $\{x\}, x \in X$  (see Fig. 2.3.7). The arcs of the graph are those that connect parents  $Y$  and their children  $Y_i$ . Suppose that for every nonterminal node  $Y$  there is an algorithm that calculates upper and lower bounds  $\underline{f}_Y$  and  $\bar{f}_Y$  for the minimum cost over  $Y$ :

$$\underline{f}_Y \leq \min_{x \in Y} f(x) \leq \bar{f}_Y.$$

Assume further that the upper and lower bounds are exact for each singleton solution node  $\{x\}$ :

$$\underline{f}_{\{x\}} = f(x) = \bar{f}_{\{x\}}, \quad \text{for all } x \in X.$$

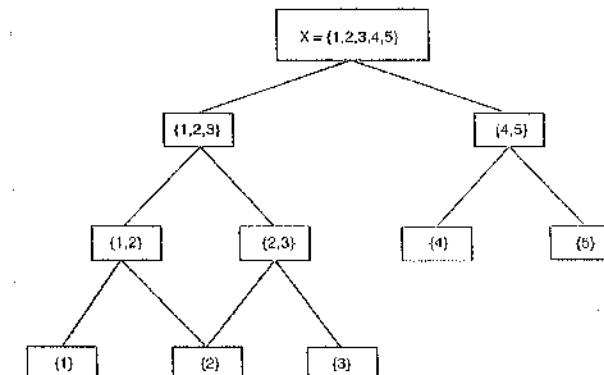


Figure 2.3.7 An acyclic graph corresponding to a branch-and-bound algorithm. Each node (subset) except those consisting of a single solution is partitioned into several other nodes (subsets).

Define now the length of an arc involving a parent  $Y$  and a child  $Y_i$ , to be the lower bound difference

$$\underline{f}_{Y_i} - \underline{f}_Y.$$

Then the length of any path from the origin node  $X$  to any node  $Y$  is  $\underline{f}_Y$ . Since  $\underline{f}_{\{x\}} = f(x)$  for all feasible solutions  $x \in X$ , it is clear that minimizing  $f(x)$  over  $x \in X$  is equivalent to finding a shortest path from the origin node to one of the singleton nodes  $\{x\}$ .

Consider now a variation of the label correcting method, where in addition we use our knowledge of the upper bounds  $\bar{f}_Y$  to reduce the value of UPPER. Initially, OPEN contains just  $X$ , and UPPER equals  $\bar{f}_X$ .

#### Branch-and-Bound Algorithm

**Step 1:** Remove a node  $Y$  from OPEN. For each child  $Y_j$  of  $Y$ , do the following: If  $\underline{f}_{Y_j} < \text{UPPER}$ , then place  $Y_j$  in OPEN. If in addition  $\bar{f}_{Y_j} < \text{UPPER}$ , then set  $\text{UPPER} = \bar{f}_{Y_j}$ , and if  $Y_j$  consists of a single solution, mark that solution as being the best solution found so far.

**Step 2: (Termination Test)** If OPEN is nonempty, go to step 1. Otherwise, terminate; the best solution found so far is optimal.

An alternative termination step 2 for the preceding algorithm is to set a tolerance  $\epsilon > 0$ , and check whether UPPER and the minimum lower bound  $\underline{f}_Y$  over all sets  $Y$  in the OPEN list differ by less than  $\epsilon$ . If so, the algorithm is terminated, and some set in OPEN must contain a solution that is within  $\epsilon$  of being optimal. There are a number of other variations of the algorithm. For example, if the upper bound  $\bar{f}_Y$  at a node is actually the cost  $f(x)$  of some element  $x \in Y$ , then this element can be taken as the best solution found so far whenever  $\bar{f}_Y < \text{UPPER}$  in step 2. Other variations relate to the method of selecting a node from OPEN in step 1. For example, two strategies of the best-first type are to select the node with minimal lower or upper bound. Note that it is neither practical nor necessary to generate *a priori* the acyclic graph of the branch-and-bound method. Instead, one may adaptively decide on the order and the manner in which the parent sets are partitioned into children sets based on the progress of the algorithm.

We finally note that to apply branch-and-bound effectively, it is important to have good algorithms for generating upper and lower bounds at each node. These bounds should be as sharp as practically possible. Then, fewer nodes will be admitted into OPEN, with attendant computational savings. Typically, continuous optimization problems (usually linear or network optimization problems) are used to obtain lower bounds to the optimal costs of the restricted problems  $\min_{x \in Y} f(x)$ , while various heuristics are used to construct corresponding feasible solutions whose costs can

be used as upper bounds. We refer to textbooks such as Nemhauser and Wolsey [NeW88], Bertsimas and Tsitsiklis [BeT97], Bertsekas [Ber98a], and Wolsey [Wol98] for fuller accounts.

#### 2.3.4 Constrained and Multiobjective Problems

In some shortest path contexts, there may be constraints on the resources required to traverse the optimal path, such as limits on time, fuel, etc. For example, there could be a restriction that the total time to travel through the optimal path  $P$  should not exceed a given threshold  $T$ , i.e.,

$$\sum_{(i,j) \in P} \tau_{ij} \leq T,$$

where  $\tau_{ij}$  is the time required to traverse arc  $(i,j)$ . Similarly, there could be a safety constraint, whereby the probability of being able to traverse the path  $P$  safely should be no less than a given threshold. Here, we assume that traversal of an arc  $(i,j)$  will be safe with a given probability  $p_{ij}$ . Assuming probabilistic independence of the safety of arc traversals, the probability that traversal of a path  $P$  will be safe is the product  $\prod_{(i,j) \in P} p_{ij}$ . The requirement that this probability is no less than a given threshold  $\beta$  translates to a path length constraint of the form

$$\sum_{(i,j) \in P} \ln(p_{ij}) \geq \ln(\beta).$$

We represent path constraints in the generic form

$$\sum_{(i,j) \in P} c_{ij}^m \leq b^m, \quad m = 1, \dots, M, \quad (2.9)$$

where  $c_{ij}^m$  is the amount of  $m$ th resource required to traverse arc  $(i,j)$ , and  $b^m$  is the total amount of  $m$ th resource available. Thus the problem is to find a path  $P$  that starts at the origin  $s$ , ends at the destination  $t$ , satisfies the constraints (2.9), and minimizes

$$\sum_{(i,j) \in P} a_{ij}.$$

We refer to this problem as the *constrained shortest path problem*, and we note that it is closely related to the *constraint feasibility problem*, where we simply want to find a path  $P$  that satisfies the constraints (2.9). In particular, the constraint feasibility problem is the special case of the constrained shortest path problem where  $a_{ij} = 0$  for all arcs  $(i,j)$ .

Conversely, the constrained shortest path problem is equivalent to the constraint feasibility problem involving the constraints (2.9) and the additional constraint

$$\sum_{(i,j) \in P} a_{ij} \leq L^*,$$

where  $L^*$  is the optimal path length (which, however, is generally unknown).

Another, closely related problem is the *multiobjective shortest path problem*, where we want to find a path  $P$  that simultaneously makes all the lengths

$$\sum_{(i,j) \in P} c_{ij}^m, \quad m = 1, \dots, M,$$

"small," in a sense that we will now make precise. In particular, for any set  $S \subset \mathbb{R}^M$ , let us call a vector  $x = (x_1, \dots, x_M) \in S$  *noninferior* if  $x$  is not dominated by any vector  $y = (y_1, \dots, y_M) \in S$ , in the sense that

$$y_m \leq x_m, \quad m = 1, \dots, M,$$

and with strict inequality for at least one  $m$  (see Fig. 2.3.8). More generally, given a problem with multiple cost functions  $f_1(x), \dots, f_M(x)$  and a constraint set  $X$ , we say that  $x$  is a *noninferior* solution if the vector of costs of  $x$ , i.e.,  $(f_1(x), \dots, f_M(x))$ , is a noninferior vector of the set of attainable costs

$$\{(f_1(y), \dots, f_M(y)) \mid y \in X\}.$$

Note that given a finite set of solutions, there is at least one noninferior solution. Furthermore, one can extract the set of noninferior solutions with a simple algorithm: sequentially test all solutions and discard those that are dominated by some not yet discarded solution, until no more solutions can be discarded.

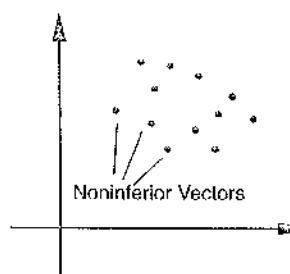


Figure 2.3.8 Illustration of the noninferior vectors of a finite set.

The multiobjective shortest path problem is to find a noninferior path  $P$ , i.e., one for which there is no other path  $P'$  satisfying

$$\sum_{(i,j) \in P'} c_{ij}^m \leq \sum_{(i,j) \in P} c_{ij}^m, \quad m = 1, \dots, M,$$

and with strict inequality for at least one  $m$ . Note that the constraint feasibility problem has a solution if and only if the subset of multiobjective/noninferior solutions that satisfy the constraints (2.9) is nonempty. It follows that the constraint feasibility problem can be easily solved once the set of all noninferior paths is computed. Similarly, the constrained shortest path problem can be solved by casting it as a multiobjective shortest path problem, where the multiple objectives correspond to the cost and the constraints. Given the set of all noninferior solutions, one obtains an optimal solution of the constrained shortest path problem (provided a feasible solution exists), by selecting a path from this set that satisfies the constraints and minimizes the cost. Because of the connections outlined above, the three problems, constrained shortest path, constraint feasibility, and multiobjective, fundamentally share the same mathematical structure, and can be addressed with similar methodology.

### Multiobjective DP Problems

We have seen that shortest path problems and deterministic finite-state DP problems are equivalent, so it is not surprising that the methodology for multiobjective and/or constrained shortest path problems is shared by multiobjective and/or constrained deterministic finite-state DP problems. A multiobjective version of such a problem involves a single controlled deterministic finite-state system

$$x_{k+1} = f_k(x_k, u_k),$$

and multiple cost functions of the form

$$g_N^m(x_N) + \sum_{k=0}^{N-1} g_k^m(x_k, u_k), \quad m = 1, \dots, M.$$

Let us provide an extension of the DP algorithm that finds the set of all noninferior solutions to the multiobjective deterministic DP problem involving the above system and cost functions. This algorithm proceeds backwards from the terminal time, and calculates for each stage  $k$  and state  $x_k$ , the set of noninferior control sequences for the tail (multiobjective) subproblem that starts at state  $x_k$ . The algorithm is based on a fairly evident extension of the principle of optimality:

If  $\{u_k, u_{k+1}, \dots, u_{N-1}\}$  is a noninferior control sequence for the tail subproblem that starts at  $x_k$ , then  $\{u_{k+1}, \dots, u_{N-1}\}$  is a noninferior control sequence for the tail subproblem that starts at  $f_k(x_k, u_k)$ .

This allows the convenient calculation of the set of noninferior solutions of tail subproblems using the sets of noninferior solutions of shorter subproblems.

More specifically, let  $\mathcal{F}_k(x_k)$  be the set of all  $M$ -tuples of costs-to-go

$$\left( g_N^1(x_N) + \sum_{i=k}^{N-1} g_i^1(x_i, u_i), \dots, g_N^M(x_N) + \sum_{i=k}^{N-1} g_i^M(x_i, u_i) \right), \quad (2.10)$$

which correspond to feasible control sequences  $\{u_k, \dots, u_{N-1}\}$  that start at  $x_k$  and are noninferior in the following sense: There is no other feasible control sequence  $\{u'_k, \dots, u'_{N-1}\}$  with corresponding state sequence  $\{x'_k, \dots, x'_{N-1}\}$  (where  $x'_k = x_k$ ) such that

$$g_N^m(x_N') + \sum_{i=k}^{N-1} g_i^m(x'_i, u'_i) \leq g_N^m(x_N) + \sum_{i=k}^{N-1} g_i^m(x_i, u_i), \quad m = 1, \dots, M,$$

and with strict inequality for at least one  $m$ . Note that  $\mathcal{F}_k(x_k)$  is a finite set (since the control constraint set is finite, which implies that the set of control sequences is finite). The sets  $\mathcal{F}_k(x_k)$  are generated by an algorithm that starts at the terminal time  $N$  with  $\mathcal{F}_N(x_N)$  consisting of just the vector of terminal costs,

$$\mathcal{F}_N(x_N) = \left\{ (g_N^1(x_N), \dots, g_N^M(x_N)) \right\},$$

and proceeds backwards according to the following process: Given the set  $\mathcal{F}_{k+1}(x_{k+1})$  for all states  $x_{k+1}$ , the algorithm generates for each state  $x_k$  the set of vectors

$$(g_k^1(x_k, u_k) + c^1, \dots, g_k^M(x_k, u_k) + c^M)$$

such that

$$(c^1, \dots, c^M) \in \mathcal{F}_{k+1}(f_k(x_k, u_k)), \quad u_k \in U_k(x_k);$$

then it obtains  $\mathcal{F}_k(x_k)$  by extracting the noninferior subset, i.e., by discarding from this set the vectors that are dominated by other vectors.

After  $N$  steps, this algorithm yields  $\mathcal{F}_0(x_0)$ , the set of all noninferior  $M$ -tuples of costs-to-go starting at initial state  $x_0$ . Note that the algorithm is similar to the ordinary DP algorithm: it just maintains a set of noninferior  $M$ -tuples of costs-to-go at each state  $x_k$ , rather than a single cost-to-go. Note also that by a similar argument to the one of Section 2.1, it is possible to construct a forward version of this algorithm.

### Constrained DP Problems

Let us now consider a related constrained DP problem with the same controlled system

$$x_{k+1} = f_k(x_k, u_k),$$

where we want to minimize the cost function

$$g_N^1(x_N) + \sum_{k=0}^{N-1} g_k^1(x_k, u_k) \quad (2.11)$$

subject to the constraints

$$g_N^m(x_N) + \sum_{k=0}^{N-1} g_k^m(x_k, u_k) \leq b^m, \quad m = 2, \dots, M. \quad (2.12)$$

We can solve this problem by finding the set of noninferior solutions/control sequences of the multiobjective DP problem involving the costs (2.10), by extracting from this set the subset of solutions/control sequences that satisfy the constraints (2.12), and by selecting from this subset a solution/control sequence that minimizes the cost (2.11).

However, we can enhance this algorithm by discarding at the earliest opportunity control sequences that cannot be part of a feasible control sequence. The rationale for this is related to the ideas underlying label correcting methods and the  $A^*$  algorithm in particular (cf. Section 2.3.2). For  $m = 2, \dots, M$ , let  $\tilde{J}_k^m(x_k)$  be the optimal cost to arrive to  $x_k$  from the given initial state  $x_0$  with cost per stage equal to  $g_i^m(x_i, u_i)$ . This is the minimal value of

$$\sum_{i=0}^{k-1} g_i^m(x_i, u_i), \quad m = 2, \dots, M,$$

subject to the constraint that the state at the  $k$ th stage is  $x_k$ , and can be calculated by the forward DP algorithm of Section 2.1. Consider now a DP-like algorithm that generates for each state and stage, a subset of  $M$ -tuples. It starts at the terminal time  $N$  with the set  $\mathcal{F}_N(x_N)$  that consists of just the vector of terminal costs,

$$\mathcal{F}_N(x_N) = \left\{ (g_N^1(x_N), \dots, g_N^M(x_N)) \right\}.$$

It proceeds backwards as follows: given the set  $\mathcal{F}_{k+1}(x_{k+1})$  for each state  $x_{k+1}$ , it generates for each state  $x_k$  the set of  $M$ -tuples

$$(g_k^1(x_k, u_k) + c^1, \dots, g_k^M(x_k, u_k) + c^M) \quad (2.13)$$

such that

$$(c^1, \dots, c^M) \in \mathcal{F}_{k+1}(f_k(x_k, u_k)), \quad u_k \in U_k(x_k),$$

and

$$\tilde{J}_k^m(x_k) + g_k^m(x_k, u_k) + c^m \leq b^m, \quad m = 2, \dots, M; \quad (2.14)$$

then it obtains  $\mathcal{F}_k(x_k)$  by extracting the noninferior subset, i.e., by discarding from this set the elements that are dominated by other elements.

Note that  $M$ -tuples of the form (2.13) that violate the condition (2.14) correspond to paths that cannot be feasible, so they can be safely excluded from further consideration [in fact  $\tilde{J}_k^m(x_k)$  may be replaced by any conveniently available underestimate in the condition (2.14); using  $\tilde{J}_k^m(x_k)$  makes this condition as sharp as possible]. The set  $\mathcal{F}_0(x_0)$  obtained by the algorithm after  $N$  steps consists of  $M$ -tuples of the cost and the constraint function values

$$g_N^m(x_N) + \sum_{k=0}^{N-1} g_k^m(x_k, u_k), \quad m = 1, \dots, M,$$

which correspond to all the feasible solutions that are noninferior. The first component of an  $M$ -tuple in  $\mathcal{F}_0(x_0)$  corresponds to the cost (2.11). An element of  $\mathcal{F}_0(x_0)$  whose first component is minimal is an optimal solution of the constrained shortest path problem. The advantage of using the criterion (2.14) is that it allows us to discard as early as possible infeasible solutions, and accordingly reduce the size of the sets  $\mathcal{F}_k(x_k)$  and the attendant computation.

Note also that if an upper bound, call it **UPPER**, is known for the optimal path length, it can be used to introduce the additional constraint

$$g_N^1(x_N) + \sum_{k=0}^{N-1} g_k^1(x_k, u_k) \leq \text{UPPER},$$

and to make the test (2.14) more effective by augmenting it with the additional inequality

$$\tilde{J}_k^1(x_k) + g_k^1(x_k, u_k) + c^1 \leq \text{UPPER}.$$

Any  $M$ -tuple  $(c^1, \dots, c^M)$  that violates this condition corresponds to paths that cannot be optimal, so it can be safely excluded from further consideration. This idea may be further enhanced by introducing schemes to reduce **UPPER** as the algorithm progresses, similar to label correcting methods.

Clearly multiobjective and constrained DP algorithms require quite a bit more computation and storage than ordinary DP for the same system.

For this reason, there have been many efforts to develop approximate solution methods. These methods are outside our scope and we refer to the literature on the subject.

The preceding DP algorithms for multiobjective and constrained problems are easily adapted to shortest path problems. One possibility is to use the transformation described in Section 2.1 to reformulate the shortest path problem as a (multiobjective or constrained) deterministic finite-state problem. The latter problem can in turn be solved using the DP-like algorithms just described. It is also possible to use versions of label correcting algorithms, including the  $A^*$  variant, for multiobjective and constrained shortest path problems. A label of a node now is not just a single number, but rather it is an  $M$ -dimensional vector with components that correspond to the  $M$  cost functions; see the end-of-chapter references.

## 2.4 NOTES, SOURCES, AND EXERCISES

Work on the shortest path problem is very extensive. Literature surveys are given by Dreyfus [Dre69], Deo and Pang [DeP84], and Gallo and Pallottino [GaP88]. For a detailed textbook treatment of shortest paths, see Bertsekas [Ber98a] (the chapter on shortest paths of this book is www-accessible), and also [Ber91a], which contains a variety of associated computer codes.

For a treatment of critical path analysis, see Elmaghraby [Elm78]. A tutorial survey of Hidden Markov Models is given by Rabiner [Rab89]. The Viterbi algorithm, first proposed in [Vit67], is also discussed by Forney [For73]. For applications in communication systems, see Proakis and Salehi [PrS94], and Sklar [Skl88]. For applications in speech recognition, see Rabiner [Rab89] and Picone [Pic90]. For applications in data network routing, see Bertsekas and Gallager [BeG92].

Label correcting methods draw their origin from the works of Bellman [Bel57] and Ford [For56]. The D'Esopo-Pape algorithm appeared in [Pap74] and is based on an earlier suggestion of D'Esopo. For a discussion of various implementations of Dijkstra's algorithm, see Bertsekas [Ber98a]. The SLF method and some variations were proposed by the author in [Ber93]; see also Bertsekas, Guerricco, and Musmanno [BCM96], where the LLL strategy as well as implementations on a parallel computer of various label correcting methods are discussed. The  $A^*$  method was proposed by Hart, Nilsson, and Raphael [HNR68] (with corrections in [HNR72]). See also the texts by Nilsson [Nil71], [Nil80], and Pearl [Pea84], which provide a broader discussion of the application of shortest path methods in artificial intelligence.

The Dijkstra algorithm has been extended to continuous space shortest path problems by Tsitsiklis [Tsi75]. The SLF/LLL methods have also

been similarly extended by Bertsekas, Guerriero, and Musmanno [BGM95], and by Polymenakos, Bertsekas, and Tsitsiklis [PBT98].

There is extensive literature on exact and approximate solution methods for constrained and multiobjective shortest path and DP problems. Analogs of label correcting and Dijkstra-like methods were proposed by Vincke [Vin74] and Hansen [Han80], respectively; see also Jaffe [Jaf84] and Martins [Mar84]. Recent work includes Guerriero and Musmanno [GuM01], who investigate analogs of the SLF/LLL methods, and give many references and computational results. For a multiobjective version of the  $A^*$  method, see Stewart and White [StW91], who also survey earlier work.

## EXERCISES

### 2.1

Find a shortest path from each node to node 6 for the graph of Fig. 2.4.1 by using the DP algorithm.

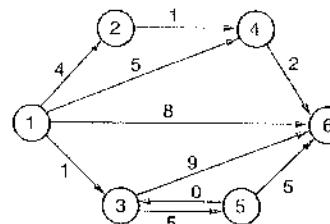


Figure 2.4.1 Graph for Exercise 2.1. The arc lengths are shown next to the arcs.

### 2.2

Find a shortest path from node 1 to node 5 for the graph of Fig. 2.4.2 by using the label correcting method of Section 2.3.1.

### 2.3

Air transportation is available between  $n$  cities, in some cases directly and in others through intermediate stops and change of carrier. The airfare between cities

### Sec. 2.4 Notes, Sources, and Exercises

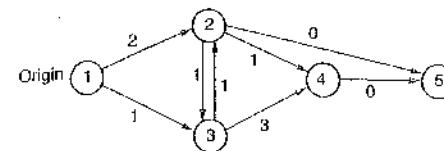


Figure 2.4.1 Graph for Exercise 2.2. The arc lengths are shown next to the arcs.

$i$  and  $j$  is denoted by  $a_{ij}$ . We assume that  $a_{ij} = a_{ji}$ , and for notational convenience, we write  $a_{ij} = \infty$  if there is no direct flight between  $i$  and  $j$ . The problem is to find the cheapest airfare for going between two cities perhaps through intermediate stops. Let  $n = 6$  and  $a_{12} = 30$ ,  $a_{13} = 60$ ,  $a_{14} = 25$ ,  $a_{15} = a_{16} = \infty$ ,  $a_{23} = a_{24} = a_{25} = \infty$ ,  $a_{26} = 50$ ,  $a_{34} = 35$ ,  $a_{35} = a_{36} = \infty$ ,  $a_{45} = 15$ ,  $a_{46} = \infty$ ,  $a_{56} = 15$ . Find the cheapest airfare from every city to every other city by using the DP algorithm.

### 2.4 (Dijkstra's Algorithm for Shortest Paths) [www](#)

Consider the best-first version of the label correcting algorithm of Section 2.3.1. Here at each iteration we remove from OPEN a node that has minimum label over all nodes in OPEN.

- Show that each node  $j$  will enter OPEN at most once, and show that at the time it exits OPEN, its label  $d_j$  is equal to the shortest distance from  $s$  to  $j$ . Hint: Use the nonnegative arc length assumption to argue that in the label correcting algorithm, in order for the node  $i$  that exits OPEN to reenter, there must exist another node  $k$  in OPEN with  $d_k + a_{ki} < d_i$ .
- Show that the number of arithmetic operations required for termination is bounded by  $cN^2$  where  $N$  is the number of nodes and  $c$  is some constant.

### 2.5 (Label Correcting for Acyclic Graphs)

Consider a shortest path problem involving an acyclic graph. Let  $S_k$  be the set of nodes  $i$  such that all paths from the origin to  $i$  have  $k$  arcs or less and at least one such path has  $k$  arcs. Consider a label correcting algorithm that removes from OPEN a node of  $S_k$  only if there are no nodes of  $S_1, \dots, S_{k-1}$  in OPEN. Show that each node will enter OPEN at most once. How does this result relate to the type of shortest path problem arising from deterministic DP (cf. Fig. 2.1.1)?

### 2.6 (Label Correcting with Multiple Destinations) [www](#)

Consider the problem of finding a shortest path from node  $s$  to each node in a subset  $T$ , assuming that all arc lengths are nonnegative. Show that the following modified version of the label correcting algorithm of Section 2.3.1 solves the problem. Initially,  $\text{UPPER} = \infty$ ,  $d_s = 0$ , and  $d_i = \infty$  for all  $i \neq s$ .

**Step 1:** Remove a node  $i$  from OPEN and for each child  $j$  of  $i$ , execute step 2.  
**Step 2:** If  $d_i + a_{ij} < \min\{d_j, \text{UPPER}\}$ , set  $d_j = d_i + a_{ij}$ , set  $i$  to be the parent of  $j$ , and place  $j$  in OPEN if it is not already in OPEN. In addition, if  $j \in T$ , set  $\text{UPPER} = \max_{t \in T} d_t$ .

**Step 3:** If OPEN is empty, terminate; else go to step 1.

Prove a termination property such as the one of Prop. 2.3.1 for this algorithm.

## 2.7 (Label Correcting with Negative Arc Lengths)

Consider the problem of finding a shortest path from node  $s$  to node  $t$ , and assume that all cycle lengths are nonnegative (instead of all arc lengths being nonnegative). Suppose that a scalar  $u_j$  is known for each node  $j$ , which is an underestimate of the shortest distance from  $j$  to  $t$  ( $u_j$  can be taken  $-\infty$  if no underestimate is known). Consider a modified version of the typical iteration of the label correcting algorithm of Section 2.3.1, where step 2 is replaced by the following:

**Modified Step 2:** If  $d_i + a_{ij} < \min\{d_j, \text{UPPER} - u_j\}$ , set  $d_j = d_i + a_{ij}$  and set  $i$  to be the parent of  $j$ . In addition, if  $j \neq t$ , place  $j$  in OPEN if it is not already in OPEN, while if  $j = t$ , set  $\text{UPPER}$  to the new value  $d_i + a_{it}$  of  $d_t$ .

- Show that the algorithm terminates with a shortest path, assuming there is at least one path from  $s$  to  $t$  (cf. Prop. 2.3.1).
- Why is the algorithm of Section 2.3.1 a special case of the one of this exercise?

## 2.8

We have a set of  $N$  objects, denoted  $1, 2, \dots, N$ , which we want to group in clusters that consist of consecutive objects. For each cluster  $i, i+1, \dots, j$ , there is an associated cost  $a_{ij}$ . We want to find a grouping of the objects in clusters such that the total cost is minimum. Formulate the problem as a shortest path problem, and write a DP algorithm for its solution. (Note: An example of this problem arises in typesetting programs, such as TEX/LATEX, that break down a paragraph into lines in a way that optimizes the paragraph's appearance.)

## 2.9 (Shortest Path Tour Problem [BeC04])

Consider a problem of finding a shortest path from a given origin node  $s$  to a given destination node  $t$  in a graph with nonnegative arc lengths. However, there is the constraint that the path should successively pass through at least one node from given node subsets  $T_1, T_2, \dots, T_N$  (i.e., for all  $k$ , pass through some node from the subset  $T_k$  after passing through at least one node of the subsets  $T_1, \dots, T_{k-1}$ ).

- Formulate this as a dynamic programming problem.

- Show that a solution can be obtained by solving a sequence of ordinary shortest path problems, each involving a single origin and multiple destinations.

## 2.10 (Two-Sided Dijkstra Algorithm [Nic66])

Consider a problem of finding a shortest path from a given origin node  $s$  to a given destination node  $t$  in a graph with nonnegative arc lengths. Consider an algorithm that maintains two subsets of nodes,  $W$  and  $V$ , with the following properties:

- $s \in W$  and  $t \in V$ .
- If  $i \in W$  and  $j \notin W$ , then the shortest distance from  $s$  to  $i$  is less than or equal to the shortest distance from  $s$  to  $j$ .
- If  $i \in V$  and  $j \notin V$ , then the shortest distance from  $i$  to  $t$  is less than or equal to the shortest distance from  $j$  to  $t$ .

At each iteration the algorithm adds a new node to  $W$  and a new node to  $V$  (the Dijkstra algorithm can be used for this purpose), and terminates when  $W$  and  $V$  have a node in common. Let  $d_i^s$  be the shortest distance from  $s$  to  $i$  using paths all the nodes of which, with the possible exception of  $i$ , lie in  $W$  ( $d_i^s = \infty$  if no such path exists), and let  $d_i^t$  be the shortest distance from  $i$  to  $t$  using paths all the nodes of which, with the possible exception of  $i$ , lie in  $V$  ( $d_i^t = \infty$  if no such path exists).

- Show that upon termination, the shortest distance  $D_{st}$  from  $s$  to  $t$  is given by

$$D_{st} = \min_{i \in W} \{d_i^s + d_i^t\} = \min_{i \in W \cup V} \{d_i^s + d_i^t\} = \min_{i \in V} \{d_i^s + d_i^t\}.$$

- Show that the conclusion of part (a) holds if the algorithm is terminated once the condition

$$\min_{i \in W} \{d_i^s + d_i^t\} \leq \max_{i \in W} d_i^s + \max_{i \in V} d_i^t$$

holds, even if the sets  $W$  and  $V$  have no node in common.

## 2.11 (DP on Two Parallel Processors [Las85])

Formulate a DP algorithm to solve the deterministic problem of Section 2.1 on a parallel computer with two processors. One processor should execute a forward algorithm and the other a backward algorithm.

## 2.12 (Doubling Algorithms)

Consider a deterministic finite-state problem that is time-invariant in the sense that the state and control spaces, the cost per stage, and the system equation

are the same for each stage. Let  $J_k(x, y)$  be the optimal cost to reach state  $y$  at time  $k$  starting from state  $x$  at time 0. Show that for all  $k$

$$J_{2k}(x, y) = \min_z \{ J_k(x, z) + J_k(z, y) \}.$$

Discuss how this equation may be used with advantage to solve problems with a large number of stages.

### 2.13 (Distributed Shortest Path Computation [Ber82a])

Consider the problem of finding a shortest path from nodes  $1, 2, \dots, N$  to node  $t$ , and assume that all arc lengths are nonnegative and all cycle lengths are positive. Consider the iteration

$$d_i^{k+1} = \min_j [a_{ij} + d_j^k], \quad i = 1, 2, \dots, N, \quad (2.15)$$

$$d_t^{k+1} = 0.$$

- (a) It was shown in Section 2.1 that if the initial condition is  $d_i^0 = \infty$  for  $i = 1, \dots, N$  and  $d_t^0 = 0$ , then the iteration (2.15) yields the shortest distances in  $N$  steps. Show that if the initial condition is  $d_i^0 = 0$ , for all  $i = 1, \dots, N, t$ , then the iteration (2.15) yields the shortest distances in a finite number of steps.
- (b) Assume that the iteration

$$d_i := \min_j [a_{ij} + d_j] \quad (2.16)$$

is executed at node  $i$  in parallel with the corresponding iteration for  $d_j$  at every other node  $j$ . However, the times of execution of this iteration at the various nodes are not synchronized. Furthermore, each node  $i$  communicates the results of its latest computation of  $d_i$  at arbitrary times with potentially large communication delays. Therefore, there is the possibility of a node executing iteration (2.16) several times before receiving a communication from every other neighboring node. Assume that each node never stops executing iteration (2.16) and transmitting the result to the other nodes. Show that the values  $\bar{d}_i^T$  available at time  $T$  at the corresponding nodes  $i$  are equal to the shortest distances for all  $T$  greater than a finite time  $\bar{T}$ . *Hint:* Let  $\bar{d}_i^k$  and  $\underline{d}_i^k$  be generated by iteration (2.16) when starting from the first and the second initial conditions in part (a), respectively. Show that for every  $k$  there exists a time  $T_k$  such that for all  $T \geq T_k$  and  $k$ , we have  $\underline{d}_i^k \leq \bar{d}_i^T \leq \bar{d}_i^k$ . *Note:* For a detailed analysis of asynchronous iterations, including algorithms for shortest paths and DP, see Bertsekas and Tsitsiklis [BeT89], Ch. 6. Distributed asynchronous shortest path algorithms find extensive application in the problem of packet routing in data communication networks. For a related discussion and analysis, see Bertsekas and Gallager [BeG92], Ch. 5.

### 2.14 (Shortest Paths for an Infinite Number of Nodes)

Consider the shortest path problem of Section 2.3, except that the number of nodes in the graph may be countably infinite (although the number of outgoing arcs from each node is finite). We assume that the length of each arc is a positive integer. Furthermore, there is at least one path from the origin node  $s$  to the destination node  $t$ . Consider the label correcting algorithm as stated and initialized in Section 2.3.1, except that  $\text{UPPER}$  is initially set to some integer that is an upper bound to the shortest distance from  $s$  to  $t$ . Show that the algorithm will terminate in a finite number of steps with  $\text{UPPER}$  equal to the shortest distance from  $s$  to  $t$ . *Hint:* Show that there is a finite number of nodes whose shortest distance from  $s$  does not exceed the initial value of  $\text{UPPER}$ .

### 2.15 (Path Bottleneck Problem)

Consider the framework of the shortest path problem. For any path  $P$ , define the *bottleneck arc* of  $P$  as an arc that has maximum length over all arcs of  $P$ . Consider the problem of finding a path whose length of bottleneck arc is minimum, among the paths connecting an origin node and a destination node. Develop and justify an analog of the label correcting method of Section 2.3.1. *Hint:* Replace  $d_i + a_{ij}$  with  $\max\{d_i, a_{ij}\}$ .

### 2.16

Air transportation is available between all pairs of  $n$  cities, but because of a perverse fare structure, it may be more economical to go from one city to another through intermediate stops. A cost-minded traveler wants to find the minimum cost fare to go from an origin city  $s$  to a destination city  $t$ . The airfare between cities  $i$  and  $j$  is denoted by  $a_{ij}$ , and for the  $m$ th intermediate stop, there is a stopover cost  $c_m$  ( $a_{ij}$  and  $c_m$  are assumed positive). Thus, for example, to go from  $s$  to  $t$  directly it costs  $a_{st}$ , while to go from  $s$  to  $t$  with intermediate stops at cities  $i_1$  and  $i_2$ , it costs  $a_{si_1} + c_1 + a_{i_1 i_2} + c_2 + a_{i_2 t}$ .

- (a) Formulate the problem as a shortest path problem, and identify the nodes, arcs, and arc costs.
- (b) Write a corresponding DP algorithm that finds an optimal solution in  $n - 2$  stages.
- (c) Assume that  $c_m$  is the same for all  $m$ . Devise a rule for detecting that an optimal solution has been found before iteration  $n - 2$  of the DP algorithm.

### 2.17

A businessman operates out of a van that he sets up in one of two locations on each day. If he operates in location  $i$  (where  $i = 1, 2$ ) on day  $k$ , he makes a known and predictable profit, denoted  $r_k^i$ . However, each time he moves from one location to the other, he pays a setup cost  $c$ . The businessman wants to maximize his total profit over  $N$  days.

- (a) Show that the problem can be formulated as a shortest path problem, and write the corresponding DP algorithm.
- (b) Suppose he is at location  $i$  on day  $k$ . Let

$$R_k^i = r_k^{\bar{i}} - r_k^i,$$

where  $\bar{i}$  denotes the location that is not equal to  $i$ . Show that if  $R_k^i \leq 0$  it is optimal to stay at location  $i$ , while if  $R_k^i \geq 2c$ , it is optimal to switch.

- (c) Suppose that on each day there is a probability of rain  $p_i$  at location  $i$  independently of rain in the other location, and independently of whether it rained on other days. If he is at location  $i$  and it rains, his profit for the day is reduced by a factor  $\beta_i$ . Can the problem still be formulated as a shortest path problem? Write a DP algorithm.
- (d) Suppose there is a possibility of rain as in part (c), but the businessman receives an accurate rain forecast just before making the decision to switch or not switch locations. Can the problem still be formulated as a shortest path problem? Write a DP algorithm.

## 3

# Deterministic Continuous-Time Optimal Control

## Contents

3.1. Continuous-Time Optimal Control . . . . .	p. 106
3.2. The Hamilton-Jacobi-Bellman Equation . . . . .	p. 109
3.3. The Pontryagin Minimum Principle . . . . .	p. 115
3.3.1. An Informal Derivation Using the HJB Equation	p. 115
3.3.2. A Derivation Based on Variational Ideas	p. 125
3.3.3. Minimum Principle for Discrete-Time Problems	p. 129
3.4. Extensions of the Minimum Principle . . . . .	p. 131
3.4.1. Fixed Terminal State . . . . .	p. 131
3.4.2. Free Initial State . . . . .	p. 135
3.4.3. Free Terminal Time . . . . .	p. 135
3.4.4. Time-Varying System and Cost . . . . .	p. 138
3.4.5. Singular Problems . . . . .	p. 139
3.5. Notes, Sources, and Exercises . . . . .	p. 142

In this chapter, we provide an introduction to continuous-time deterministic optimal control. We derive the analog of the DP algorithm, which is the Hamilton-Jacobi-Bellman equation. Furthermore, we develop a celebrated theorem of optimal control, the Pontryagin Minimum Principle and its variations. We discuss two different derivations of this theorem, one of which is based on DP. We also illustrate the theorem by means of examples.

### 3.1 CONTINUOUS-TIME OPTIMAL CONTROL

We consider a continuous-time dynamic system

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)), \quad 0 \leq t \leq T, \\ x(0) &: \text{given},\end{aligned}\tag{3.1}$$

where  $x(t) \in \mathbb{R}^n$  is the state vector at time  $t$ ,  $\dot{x}(t) \in \mathbb{R}^n$  is the vector of first order time derivatives of the states at time  $t$ ,  $u(t) \in U \subset \mathbb{R}^m$  is the control vector at time  $t$ ,  $U$  is the control constraint set, and  $T$  is the terminal time. The components of  $f$ ,  $x$ ,  $\dot{x}$ , and  $u$  will be denoted by  $f_i$ ,  $x_i$ ,  $\dot{x}_i$ , and  $u_i$ , respectively. Thus, the system (3.1) represents the  $n$  first order differential equations

$$\frac{dx_i(t)}{dt} = f_i(x(t), u(t)), \quad i = 1, \dots, n.$$

We view  $\dot{x}(t)$ ,  $x(t)$ , and  $u(t)$  as column vectors. We assume that the system function  $f_i$  is continuously differentiable with respect to  $x$  and is continuous with respect to  $u$ . The admissible control functions, also called *control trajectories*, are the piecewise continuous functions  $\{u(t) \mid t \in [0, T]\}$  with  $u(t) \in U$  for all  $t \in [0, T]$ .

We should stress at the outset that the subject of this chapter is highly sophisticated, and it is beyond our scope to develop it according to high standards of mathematical rigor. In particular, we assume that, for any admissible control trajectory  $\{u(t) \mid t \in [0, T]\}$ , the system of differential equations (3.1) has a unique solution, which is denoted  $\{x^u(t) \mid t \in [0, T]\}$  and is called the corresponding *state trajectory*. In a more rigorous treatment, the issue of existence and uniqueness of this solution would have to be addressed more carefully.

We want to find an admissible control trajectory  $\{u(t) \mid t \in [0, T]\}$ , which, together with its corresponding state trajectory  $\{x(t) \mid t \in [0, T]\}$ , minimizes a cost function of the form

$$h(x(T)) + \int_0^T g(x(t), u(t)) dt,$$

where the functions  $g$  and  $h$  are continuously differentiable with respect to  $x$ , and  $g$  is continuous with respect to  $u$ .

#### Example 3.1.1 (Motion Control)

A unit mass moves on a line under the influence of a force  $u$ . Let  $x_1(t)$  and  $x_2(t)$  be the position and velocity of the mass at time  $t$ , respectively. From a given  $(x_1(0), x_2(0))$  we want to bring the mass “near” a given final position-velocity pair  $(\bar{x}_1, \bar{x}_2)$  at time  $T$ . In particular, we want to

$$\text{minimize } |x_1(T) - \bar{x}_1|^2 + |x_2(T) - \bar{x}_2|^2$$

subject to the control constraint

$$|u(t)| \leq 1, \quad \text{for all } t \in [0, T].$$

The corresponding continuous-time system is

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = u(t),$$

and the problem fits the general framework given earlier with cost functions given by

$$\begin{aligned}h(x(T)) &= |x_1(T) - \bar{x}_1|^2 + |x_2(T) - \bar{x}_2|^2, \\ g(x(t), u(t)) &= 0, \quad \text{for all } t \in [0, T].\end{aligned}$$

There are many variations of the problem; for example, the final position and/or velocity may be fixed. These variations can be handled by various reformulations of the general continuous-time optimal control problem, which will be given later.

#### Example 3.1.2 (Resource Allocation)

A producer with production rate  $x(t)$  at time  $t$  may allocate a portion  $u(t)$  of his/her production rate to reinvestment and  $1 - u(t)$  to production of a storable good. Thus  $x(t)$  evolves according to

$$\dot{x}(t) = \gamma u(t)x(t),$$

where  $\gamma > 0$  is a given constant. The producer wants to maximize the total amount of product stored

$$\int_0^T (1 - u(t))x(t) dt$$

subject to

$$0 \leq u(t) \leq 1, \quad \text{for all } t \in [0, T].$$

The initial production rate  $x(0)$  is a given positive number.

**Example 3.1.3 (Calculus of Variations Problems)**

Calculus of variations problems involve finding (possibly multidimensional) curves  $x(t)$  with certain optimality properties. They are among the most celebrated problems of applied mathematics and have been worked on by many of the illustrious mathematicians of the past 300 years (Euler, Lagrange, Bernoulli, Gauss, etc.). We will see that calculus of variations problems can be reformulated as optimal control problems. We illustrate this reformulation by a simple example.

Suppose that we want to find a minimum length curve that starts at a given point and ends at a given line. The answer is of course evident, but we want to derive it by using a continuous-time optimal control formulation. Without loss of generality, we let  $(0, \alpha)$  be the given point, and we let the given line be the vertical line that passes through  $(T, 0)$ , as shown in Fig. 3.1.1. Let also  $(t, x(t))$  be the points of the curve  $(0 \leq t \leq T)$ . The portion of the curve joining the points  $(t, x(t))$  and  $(t+dt, x(t+dt))$  can be approximated, for small  $dt$ , by the hypotenuse of a right triangle with sides  $dt$  and  $\dot{x}(t)dt$ . Thus the length of this portion is

$$\sqrt{(dt)^2 + (\dot{x}(t))^2 dt},$$

which is equal to

$$\sqrt{1 + (\dot{x}(t))^2} dt.$$

The length of the entire curve is the integral over  $[0, T]$  of this expression, so the problem is to

$$\begin{aligned} & \text{minimize } \int_0^T \sqrt{1 + (\dot{x}(t))^2} dt \\ & \text{subject to } x(0) = \alpha. \end{aligned}$$

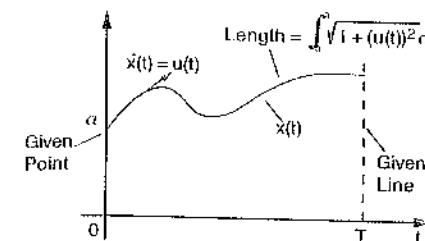
To reformulate the problem as a continuous-time optimal control problem, we introduce a control  $u$  and the system equation

$$\dot{x}(t) = u(t), \quad x(0) = \alpha.$$

Our problem then becomes

$$\text{minimize } \int_0^T \sqrt{1 + (u(t))^2} dt.$$

This is a problem that fits our continuous-time optimal control framework.



**Figure 3.1.1** Problem of finding a curve of minimum length from a given point to a given line, and its formulation as a calculus of variations problem.

### 3.2 THE HAMILTON-JACOBI-BELLMAN EQUATION

We will now derive informally a partial differential equation, which is satisfied by the optimal cost-to-go function, under certain assumptions. This equation is the continuous-time analog of the DP algorithm, and will be motivated by applying DP to a discrete-time approximation of the continuous-time optimal control problem.

Let us divide the time horizon  $[0, T]$  into  $N$  pieces using the discretization interval

$$\delta = \frac{T}{N}.$$

We denote

$$x_k = x(k\delta), \quad k = 0, 1, \dots, N,$$

$$u_k = u(k\delta), \quad k = 0, 1, \dots, N,$$

and we approximate the continuous-time system by

$$x_{k+1} = x_k + f(x_k, u_k) \cdot \delta$$

and the cost function by

$$h(x_N) + \sum_{k=0}^{N-1} g(x_k, u_k) \cdot \delta.$$

We now apply DP to the discrete-time approximation. Let

$J^*(t, x)$  : Optimal cost-to-go at time  $t$  and state  $x$   
for the continuous-time problem,

$\tilde{J}^*(t, x)$  : Optimal cost-to-go at time  $t$  and state  $x$   
for the discrete-time approximation.

The DP equations are

$$\tilde{J}^*(N\delta, x) = h(x),$$

$$\tilde{J}^*(k\delta, x) = \min_{u \in U} [g(x, u) \cdot \delta + \tilde{J}^*((k+1)\delta, x + f(x, u) \cdot \delta)], \quad k = 0, \dots, N-1.$$

Assuming that  $\tilde{J}^*$  has the required differentiability properties, we expand it into a first order Taylor series around  $(k\delta, x)$ , obtaining

$$\begin{aligned} \tilde{J}^*((k+1)\delta, x + f(x, u) \cdot \delta) &= \tilde{J}^*(k\delta, x) + \nabla_t \tilde{J}^*(k\delta, x) \cdot \delta \\ &\quad + \nabla_x \tilde{J}^*(k\delta, x)' f(x, u) \cdot \delta + o(\delta), \end{aligned}$$

where  $o(\delta)$  represents second order terms satisfying  $\lim_{\delta \rightarrow 0} o(\delta)/\delta = 0$ ,  $\nabla_t$  denotes partial derivative with respect to  $t$ , and  $\nabla_x$  denotes the  $n$ -dimensional (column) vector of partial derivatives with respect to  $x$ . Substituting in the DP equation, we obtain

$$\begin{aligned} \tilde{J}^*(k\delta, x) &= \min_{u \in U} [g(x, u) \cdot \delta + \tilde{J}^*(k\delta, x) + \nabla_t \tilde{J}^*(k\delta, x) \cdot \delta \\ &\quad + \nabla_x \tilde{J}^*(k\delta, x)' f(x, u) \cdot \delta + o(\delta)]. \end{aligned}$$

Cancelling  $\tilde{J}^*(k\delta, x)$  from both sides, dividing by  $\delta$ , and taking the limit as  $\delta \rightarrow 0$ , while assuming that the discrete-time cost-to-go function yields in the limit its continuous-time counterpart,

$$\lim_{k \rightarrow \infty, \delta \rightarrow 0, k\delta=t} \tilde{J}^*(k\delta, x) = J^*(t, x), \quad \text{for all } t, x,$$

we obtain the following equation for the cost-to-go function  $J^*(t, x)$ :

$$0 = \min_{u \in U} [g(x, u) + \nabla_t J^*(t, x) + \nabla_x J^*(t, x)' f(x, u)], \quad \text{for all } t, x,$$

with the boundary condition  $J^*(T, x) = h(x)$ .

This is the *Hamilton-Jacobi-Bellman (HJB) equation*. It is a *partial differential equation*, which should be satisfied for all time-state pairs  $(t, x)$  by the cost-to-go function  $J^*(t, x)$ , based on the preceding informal derivation, which assumed among other things, differentiability of  $J^*(t, x)$ . In fact we do not know a priori that  $J^*(t, x)$  is differentiable, so we do not know if  $J^*(t, x)$  solves this equation. However, it turns out that if we can solve the HJB equation analytically or computationally, then we can obtain an optimal control policy by minimizing its right-hand-side. This is shown in the following proposition, whose statement is reminiscent of a corresponding statement for discrete-time DP: if we can execute the DP algorithm, which may not be possible due to excessive computational requirements, we can find an optimal policy by minimization of the right-hand side.

**Proposition 3.2.1: (Sufficiency Theorem)** Suppose  $V(t, x)$  is a solution to the HJB equation; that is,  $V$  is continuously differentiable in  $t$  and  $x$ , and is such that

$$0 = \min_{u \in U} [g(x, u) + \nabla_t V(t, x) + \nabla_x V(t, x)' f(x, u)], \quad \text{for all } t, x, \quad (3.2)$$

$$V(T, x) = h(x), \quad \text{for all } x. \quad (3.3)$$

Suppose also that  $\mu^*(t, x)$  attains the minimum in Eq. (3.2) for all  $t$  and  $x$ . Let  $\{x^*(t) \mid t \in [0, T]\}$  be the state trajectory obtained from the given initial condition  $x(0)$  when the control trajectory  $w^*(t) := \mu^*(t, x^*(t))$ ,  $t \in [0, T]$  is used [that is,  $x^*(0) = x(0)$  and for all  $t \in [0, T]$ ,  $\dot{x}^*(t) = f(x^*(t), \mu^*(t, x^*(t)))$ ; we assume that this differential equation has a unique solution starting at any pair  $(t, x)$  and that the control trajectory  $\{\mu^*(t, x^*(t)) \mid t \in [0, T]\}$  is piecewise continuous as a function of  $t$ ]. Then  $V$  is equal to the optimal cost-to-go function, i.e.,

$$V(t, x) = J^*(t, x), \quad \text{for all } t, x.$$

Furthermore, the control trajectory  $\{u^*(t) \mid t \in [0, T]\}$  is optimal.

**Proof:** Let  $\{\hat{u}(t) \mid t \in [0, T]\}$  be any admissible control trajectory and let  $\{\hat{x}(t) \mid t \in [0, T]\}$  be the corresponding state trajectory. From Eq. (3.2) we have for all  $t \in [0, T]$

$$0 \leq g(\hat{x}(t), \hat{u}(t)) + \nabla_t V(t, \hat{x}(t)) + \nabla_x V(t, \hat{x}(t))' f(\hat{x}(t), \hat{u}(t)).$$

Using the system equation  $\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{u}(t))$ , the right-hand side of the above inequality is equal to the expression

$$g(\hat{x}(t), \hat{u}(t)) + \frac{d}{dt}(V(t, \hat{x}(t))),$$

where  $d/dt(\cdot)$  denotes total derivative with respect to  $t$ . Integrating this expression over  $t \in [0, T]$ , and using the preceding inequality, we obtain

$$0 \leq \int_0^T g(\hat{x}(t), \hat{u}(t)) dt + V(T, \hat{x}(T)) - V(0, \hat{x}(0)).$$

Thus by using the terminal condition  $V(T, x) = h(x)$  of Eq. (3.3) and the initial condition  $\hat{x}(0) = x(0)$ , we have

$$V(0, x(0)) \leq h(\hat{x}(T)) + \int_0^T g(\hat{x}(t), \hat{u}(t)) dt.$$

If we use  $u^*(t)$  and  $x^*(t)$  in place of  $\hat{u}(t)$  and  $\hat{x}(t)$ , respectively, the preceding inequalities becomes equalities, and we obtain

$$V(0, x(0)) = h(x^*(T)) + \int_0^T g(x^*(t), u^*(t)) dt.$$

Therefore the cost corresponding to  $\{u^*(t) \mid t \in [0, T]\}$  is  $V(0, x(0))$  and is no larger than the cost corresponding to any other admissible control trajectory  $\{\hat{u}(t) \mid t \in [0, T]\}$ . It follows that  $\{u^*(t) \mid t \in [0, T]\}$  is optimal and that

$$V(0, x(0)) = J^*(0, x(0)).$$

We now note that the preceding argument can be repeated with any initial time  $t \in [0, T]$  and any initial state  $x$ . We thus obtain

$$V(t, x) = J^*(t, x), \quad \text{for all } t, x.$$

Q.E.D.

### Example 3.2.1

To illustrate the HJB equation, let us consider a simple example involving the scalar system

$$\dot{x}(t) = u(t),$$

with the constraint  $|u(t)| \leq 1$  for all  $t \in [0, T]$ . The cost is

$$\frac{1}{2}(x(T))^2.$$

The HJB equation here is

$$0 = \min_{|u| \leq 1} [\nabla_t V(t, x) + \nabla_x V(t, x)u], \quad \text{for all } t, x, \quad (3.4)$$

with the terminal condition

$$V(T, x) = \frac{1}{2}x^2. \quad (3.5)$$

There is an evident candidate for optimality, namely moving the state towards 0 as quickly as possible, and keeping it at 0 once it is at 0. The corresponding control policy is

$$\mu^*(t, x) = -\text{sgn}(x) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x > 0. \end{cases} \quad (3.6)$$

For a given initial time  $t$  and initial state  $x$ , the cost associated with this policy can be calculated to be

$$J^*(t, x) = \frac{1}{2} \left( \max\{0, |x| - (T-t)\} \right)^2. \quad (3.7)$$

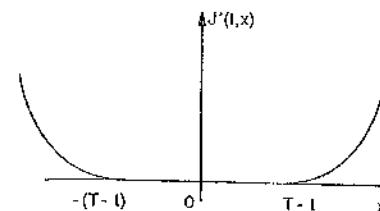


Figure 3.2.1 Optimal cost-to-go function  $J^*(t, x)$  for Example 3.2.1.

This function, which is illustrated in Fig. 3.2.1, satisfies the terminal condition (3.5), since  $J^*(T, x) = (1/2)x^2$ . Let us verify that this function also satisfies the HJB Eq. (3.4), and that  $u = -\text{sgn}(x)$  attains the minimum in the right-hand side of the equation for all  $t$  and  $x$ . Proposition 3.2.1 will then guarantee that the state and control trajectories corresponding to the policy  $\mu^*(t, x)$  are optimal.

Indeed, we have

$$\nabla_t J^*(t, x) = \max\{0, |x| - (T-t)\},$$

$$\nabla_x J^*(t, x) = \text{sgn}(x) \cdot \max\{0, |x| - (T-t)\}.$$

Substituting these expressions, the HJB Eq. (3.4) becomes

$$0 = \min_{|u| \leq 1} [1 + \text{sgn}(x) \cdot u] \max\{0, |x| - (T-t)\}, \quad (3.8)$$

which can be seen to hold as an identity for all  $(t, x)$ . Furthermore, the minimum is attained for  $u = -\text{sgn}(x)$ . We therefore conclude based on Prop. 3.2.1 that  $J^*(t, x)$  as given by Eq. (3.7) is indeed the optimal cost-to-go function, and that the policy defined by Eq. (3.6) is optimal. Note, however, that the optimal policy is not unique. Based on Prop. 3.2.1, any policy for which the minimum is attained in Eq. (3.8) is optimal. In particular, when  $|x(t)| \leq T-t$ , applying any control from the range  $[-1, 1]$  is optimal.

The preceding derivation generalizes to the case of the cost

$$h(x(T)),$$

where  $h$  is a nonnegative differentiable convex function with  $h(0) = 0$ . The corresponding optimal cost-to-go function is

$$J^*(t, x) = \begin{cases} h(x - (T-t)) & \text{if } x > T-t, \\ h(x + (T-t)) & \text{if } x < -(T-t), \\ 0 & \text{if } |x| \leq T-t, \end{cases}$$

and can be similarly verified to be a solution of the HJB equation.

### Example 3.2.2 (Linear-Quadratic Problems)

Consider the  $n$ -dimensional linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$

where  $A$  and  $B$  are given matrices, and the quadratic cost

$$x(T)'Q_T x(T) + \int_0^T (x(t)'Qx(t) + u(t)'Ru(t)) dt,$$

where the matrices  $Q_T$  and  $Q$  are symmetric positive semidefinite, and the matrix  $R$  is symmetric positive definite (Appendix A defines positive definite and semidefinite matrices). The HJB equation is

$$\begin{aligned} 0 &= \min_{u \in \mathbb{R}^m} [x'Qx + u'Ru + \nabla_t V(t, x) + \nabla_x V(t, x)'(Ax + Bu)], \\ V(T, x) &= x'Q_T x. \end{aligned} \quad (3.9)$$

Let us try a solution of the form

$$V(t, x) = x'K(t)x, \quad K(t) : n \times n \text{ symmetric},$$

and see if we can solve the HJB equation. We have  $\nabla_x V(t, x) = 2K(t)x$  and  $\nabla_t V(t, x) = x'\dot{K}(t)x$ , where  $\dot{K}(t)$  is the matrix with elements the first order derivatives of the elements of  $K(t)$  with respect to time. By substituting these expressions in Eq. (3.9), we obtain

$$0 = \min_u [x'Qx + u'Ru + x'\dot{K}(t)x + 2x'K(t)Ax + 2x'K(t)Bu]. \quad (3.10)$$

The minimum is attained at a  $u$  for which the gradient with respect to  $u$  is zero, that is,

$$2B'K(t)x + 2Ru = 0$$

or

$$u = -R^{-1}B'K(t)x. \quad (3.11)$$

Substituting the minimizing value of  $u$  in Eq. (3.10), we obtain

$$0 = x'(\dot{K}(t) + K(t)A + A'K(t) - K(t)BR^{-1}B'K(t) + Q)x, \quad \text{for all } (t, x).$$

Therefore, in order for  $V(t, x) = x'K(t)x$  to solve the HJB equation,  $K(t)$  must satisfy the following matrix differential equation (known as the *continuous-time Riccati equation*)

$$\dot{K}(t) = -K(t)A - A'K(t) + K(t)BR^{-1}B'K(t) - Q \quad (3.12)$$

with the terminal condition

$$K(T) = Q_T. \quad (3.13)$$

Reversing the argument, we see that if  $K(t)$  is a solution of the Riccati equation (3.12) with the boundary condition (3.13), then  $V(t, x) = x'K(t)x$  is a solution of the HJB equation. Thus, by using Prop. 3.2.1, we conclude that the optimal cost-to-go function is

$$J^*(t, x) = x'K(t)x.$$

Furthermore, in view of the expression derived for the control that minimizes in the right-hand side of the HJB equation [cf. Eq. (3.11)], an optimal policy is

$$\mu^*(t, x) = -R^{-1}B'K(t)x.$$

## 3.3 THE PONTRYAGIN MINIMUM PRINCIPLE

In this section we discuss the continuous-time and the discrete-time versions of the Minimum Principle, starting with a DP-based informal argument.

### 3.3.1 An Informal Derivation Using the HJB Equation

Recall the HJB equation

$$0 = \min_{u \in U} [g(x, u) + \nabla_t J^*(t, x) + \nabla_x J^*(t, x)'f(x, u)], \quad \text{for all } t, x, \quad (3.14)$$

$$J^*(T, x) = h(x), \quad \text{for all } x. \quad (3.15)$$

We argued that the optimal cost-to-go function  $J^*(t, x)$  satisfies this equation under some conditions. Furthermore, the sufficiency theorem of the preceding section suggests that if for a given initial state  $x(0)$ , the control trajectory  $\{u^*(t) \mid t \in [0, T]\}$  is optimal with corresponding state trajectory  $\{x^*(t) \mid t \in [0, T]\}$ , then for all  $t \in [0, T]$ ,

$$u^*(t) = \arg \min_{u \in U} [g(x^*(t), u) + \nabla_x J^*(t, x^*(t))'f(x^*(t), u)]. \quad (3.16)$$

Note that to obtain the optimal control trajectory via this equation, we do not need to know  $\nabla_x J^*$  at *all* values of  $x$  and  $t$ ; it is sufficient to know  $\nabla_x J^*$  at only *one* value of  $x$  for each  $t$ , that is, to know only  $\nabla_x J^*(t, x^*(t))$ .

The Minimum Principle is basically the preceding Eq. (3.16). Its application is facilitated by streamlining the computation of  $\nabla_x J^*(t, x^*(t))$ . It turns out that we can often calculate  $\nabla_x J^*(t, x^*(t))$  along the optimal state trajectory far more easily than we can solve the HJB equation. In particular,  $\nabla_x J^*(t, x^*(t))$  satisfies a certain differential equation, called the *adjoint equation*. We will derive this equation informally by differentiating the HJB equation (3.14). We first need the following lemma, which indicates how to differentiate functions involving minima.

**Lemma 3.3.1:** Let  $F(t, x, u)$  be a continuously differentiable function of  $t \in \mathbb{R}$ ,  $x \in \mathbb{R}^n$ , and  $u \in \mathbb{R}^m$ , and let  $U$  be a convex subset of  $\mathbb{R}^m$ . Assume that  $\mu^*(t, x)$  is a continuously differentiable function such that

$$\mu^*(t, x) = \arg \min_{u \in U} F(t, x, u), \quad \text{for all } t, x.$$

Then

$$\nabla_t \left\{ \min_{u \in U} F(t, x, u) \right\} = \nabla_t F(t, x, \mu^*(t, x)), \quad \text{for all } t, x,$$

$$\nabla_x \left\{ \min_{u \in U} F(t, x, u) \right\} = \nabla_x F(t, x, \mu^*(t, x)), \quad \text{for all } t, x.$$

[Note: On the left-hand side,  $\nabla_t \{\cdot\}$  and  $\nabla_x \{\cdot\}$  denote the gradients of the function  $G(t, x) = \min_{u \in U} F(t, x, u)$  with respect to  $t$  and  $x$ , respectively. On the right-hand side,  $\nabla_t$  and  $\nabla_x$  denote the vectors of partial derivatives of  $F$  with respect to  $t$  and  $x$ , respectively, evaluated at  $(t, x, \mu^*(t, x))$ .]

**Proof:** For notational simplicity, denote  $y = (t, x)$ ,  $F(y, u) = F(t, x, u)$ , and  $\mu^*(y) = \mu^*(t, x)$ . Since  $\min_{u \in U} F(y, u) = F(y, \mu^*(y))$ ,

$$\nabla \left\{ \min_{u \in U} F(y, u) \right\} = \nabla_u F(y, \mu^*(y)) + \nabla \mu^*(y) \nabla_u F(y, \mu^*(y)).$$

We will prove the result by showing that the second term in the right-hand side above is zero. This is true when  $U = \mathbb{R}^m$ , because then  $\mu^*(y)$  is an unconstrained minimum of  $F(y, u)$  and  $\nabla_u F(y, \mu^*(y)) = 0$ . More generally, for every fixed  $y$ , we have

$$(u - \mu^*(y))' \nabla_u F(y, \mu^*(y)) \geq 0, \quad \text{for all } u \in U,$$

[see Eq. (B.2) in Appendix B]. Now by Taylor's Theorem, we have that when  $y$  changes to  $y + \Delta y$ , the minimizing  $\mu^*(y)$  changes from  $\mu^*(y)$  to some vector  $\mu^*(y + \Delta y) = \mu^*(y) + \nabla \mu^*(y)' \Delta y + o(\|\Delta y\|)$  of  $U$ , so

$$(\nabla \mu^*(y)' \Delta y + o(\|\Delta y\|))' \nabla_u F(y, \mu^*(y)) \geq 0, \quad \text{for all } \Delta y,$$

implying that

$$\nabla \mu^*(y) \nabla_u F(y, \mu^*(y)) = 0.$$

Q.E.D.

Consider the HJB equation (3.14), and for any  $(t, x)$ , suppose that  $\mu^*(t, x)$  is a control attaining the minimum in the right-hand side. We make the restrictive assumptions that  $U$  is a convex set, and that  $\mu^*(t, x)$  is continuously differentiable in  $(t, x)$ , so that we can use Lemma 3.3.1. (We note, however, that alternative derivations of the Minimum Principle do not require these assumptions; see Section 3.3.2.)

We differentiate both sides of the HJB equation with respect to  $x$  and with respect to  $t$ . In particular, we set to zero the gradient with respect to  $x$  and  $t$  of the function

$$g(x, \mu^*(t, x)) + \nabla_t J^*(t, x) + \nabla_x J^*(t, x)' f(x, \mu^*(t, x)),$$

and we rely on Lemma 3.3.1 to disregard the terms involving the derivatives of  $\mu^*(t, x)$  with respect to  $t$  and  $x$ . We obtain for all  $(t, x)$ ,

$$\begin{aligned} 0 &= \nabla_x g(x, \mu^*(t, x)) + \nabla_{xt}^2 J^*(t, x) + \nabla_{xx}^2 J^*(t, x) f(x, \mu^*(t, x)) \\ &\quad + \nabla_x f(x, \mu^*(t, x)) \nabla_x J^*(t, x), \end{aligned} \quad (3.17)$$

$$0 = \nabla_{tt}^2 J^*(t, x) + \nabla_{xt}^2 J^*(t, x)' f(x, \mu^*(t, x)), \quad (3.18)$$

where  $\nabla_x f(x, \mu^*(t, x))$  is the matrix

$$\nabla_x f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

with the partial derivatives evaluated at the argument  $(x, \mu^*(t, x))$ .

The above equations hold for all  $(t, x)$ . Let us specialize them along an optimal state and control trajectory  $\{(x^*(t), u^*(t)) \mid t \in [0, T]\}$ , where  $u^*(t) = \mu^*(t, x^*(t))$  for all  $t \in [0, T]$ . We have for all  $t$ ,

$$\dot{x}^*(t) = f(x^*(t), u^*(t)),$$

so that the term

$$\nabla_{xt}^2 J^*(t, x^*(t)) + \nabla_{xx}^2 J^*(t, x^*(t)) f(x^*(t), u^*(t))$$

in Eq. (3.17) is equal to the following total derivative with respect to  $t$

$$\frac{d}{dt} (\nabla_x J^*(t, x^*(t))).$$

Similarly, the term

$$\nabla_{tt}^2 J^*(t, x^*(t)) + \nabla_{xt}^2 J^*(t, x^*(t))' f(x^*(t), u^*(t))$$

in Eq. (3.18) is equal to the total derivative

$$\frac{d}{dt} (\nabla_t J^*(t, x^*(t))).$$

Thus, by denoting

$$p(t) = \nabla_x J^*(t, x^*(t)), \quad (3.19)$$

$$p_0(t) = \nabla_x J^*(t, x^*(t)), \quad (3.20)$$

Eq. (3.17) becomes

$$\dot{p}(t) = -\nabla_x f(x^*(t), u^*(t))p(t) - \nabla_x g(x^*(t), u^*(t)) \quad (3.21)$$

and Eq. (3.18) becomes

$$\dot{p}_0(t) = 0$$

or equivalently,

$$p_0(t) = \text{constant}, \quad \text{for all } t \in [0, T]. \quad (3.22)$$

Equation (3.21) is a system of  $n$  first order differential equations known as the *adjoint equation*. From the boundary condition

$$J^*(T, x) = h(x), \quad \text{for all } x,$$

we have, by differentiation with respect to  $x$ , the relation  $\nabla_x J^*(T, x) = \nabla h(x)$ , and by using the definition  $\nabla_x J^*(t, x^*(t)) = p(t)$ , we obtain

$$p(T) = \nabla h(x^*(T)). \quad (3.23)$$

Thus, we have a terminal boundary condition for the adjoint equation (3.21).

To summarize, along optimal state and control trajectories  $x^*(t)$ ,  $u^*(t)$ ,  $t \in [0, T]$ , the adjoint equation (3.21) holds together with the boundary condition (3.23), while Eq. (3.16) and the definition of  $p(t)$  imply that  $u^*(t)$  satisfies

$$u^*(t) = \arg \min_{u \in U} [g(x^*(t), u) + p(t)'f(x^*(t), u)], \quad \text{for all } t \in [0, T]. \quad (3.24)$$

### Hamiltonian Formulation

Motivated by the condition (3.24), we introduce the Hamiltonian function mapping triplets  $(x, u, p) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$  to real numbers and given by

$$H(x, u, p) = g(x, u) + p'f(x, u).$$

Note that both the system and the adjoint equations can be compactly written in terms of the Hamiltonian as

$$\dot{x}^*(t) = \nabla_p H(x^*(t), u^*(t), p(t)), \quad \dot{p}(t) = -\nabla_x H(x^*(t), u^*(t), p(t)).$$

We state the Minimum Principle in terms of the Hamiltonian function.

**Proposition 3.3.1: (Minimum Principle)** Let  $\{u^*(t) \mid t \in [0, T]\}$  be an optimal control trajectory and let  $\{x^*(t) \mid t \in [0, T]\}$  be the corresponding state trajectory, i.e.,

$$\dot{x}^*(t) = f(x^*(t), u^*(t)), \quad x^*(0) = x(0) : \text{given.}$$

Let also  $p(t)$  be the solution of the adjoint equation

$$\dot{p}(t) = -\nabla_x H(x^*(t), u^*(t), p(t)),$$

with the boundary condition

$$p(T) = \nabla h(x^*(T)),$$

where  $h(\cdot)$  is the terminal cost function. Then, for all  $t \in [0, T]$ ,

$$u^*(t) = \arg \min_{u \in U} H(x^*(t), u, p(t)).$$

Furthermore, there is a constant  $C$  such that

$$H(x^*(t), u^*(t), p(t)) = C, \quad \text{for all } t \in [0, T].$$

All the assertions of the Minimum Principle have been (informally) derived earlier except for the last assertion. To see why the Hamiltonian function is constant for  $t \in [0, T]$  along the optimal state and control trajectories, note that by Eqs. (3.14), (3.19), and (3.20), we have for all  $t \in [0, T]$

$$H(x^*(t), u^*(t), p(t)) = -\nabla_t J^*(t, x^*(t)) = -p_0(t),$$

and  $p_0(t)$  is constant by Eq. (3.22). We should note here that the Hamiltonian function need not be constant along the optimal trajectory if the system and cost are not time-independent, contrary to our assumption thus far (see Section 3.4.4).

It is important to note that the Minimum Principle provides *necessary* optimality conditions, so all optimal control trajectories satisfy these conditions, but if a control trajectory satisfies these conditions, it is not necessarily optimal. Further analysis is needed to guarantee optimality. One method that often works is to prove that an optimal control trajectory exists, and to verify that there is only one control trajectory satisfying the conditions of the Minimum Principle (or that all control trajectories satisfying these conditions have equal cost). Another possibility to conclude optimality arises when the system function  $f$  is linear in  $(x, u)$ , the

constraint set  $U$  is convex, and the cost functions  $h$  and  $g$  are convex. Then it can be shown that the conditions of the Minimum Principle are both necessary and sufficient for optimality.

The Minimum Principle can often be used as the basis of a numerical solution. One possibility is the *two-point boundary problem method*. In this method, we use the minimum condition

$$u^*(t) = \arg \min_{u \in U} H(x^*(t), u, p(t)),$$

to express  $u^*(t)$  in terms of  $x^*(t)$  and  $p(t)$ . We then substitute the result into the system and the adjoint equations, to obtain a set of  $2n$  first order differential equations in the components of  $x^*(t)$  and  $p(t)$ . These equations can be solved using the split boundary conditions

$$x^*(0) = x(0), \quad p(T) = \nabla h(x^*(T)).$$

The number of boundary conditions (which is  $2n$ ) is equal to the number of differential equations, so that we generally expect to be able to solve these differential equations numerically (although in practice this may not be simple).

Using the Minimum Principle to obtain an analytical solution is possible in many interesting problems, but typically requires considerable creativity. We give some simple examples.

### Example 3.3.1 (Calculus of Variations Continued)

Consider the problem of finding the curve of minimum length from a point  $(0, \alpha)$  to the line  $\{(T, y) \mid y \in \mathfrak{M}\}$ . In Section 3.1, we formulated this problem as the problem of finding an optimal control trajectory  $\{u(t) \mid t \in [0, T]\}$  that minimizes

$$\int_0^T \sqrt{1 + (u(t))^2} dt$$

subject to

$$\dot{x}(t) = u(t), \quad x(0) = \alpha.$$

Let us apply the preceding necessary conditions. The Hamiltonian is

$$H(x, u, p) = \sqrt{1 + u^2} + pu,$$

and the adjoint equation is

$$\dot{p}(t) = 0, \quad p(T) = 0.$$

It follows that

$$p(t) = 0, \quad \text{for all } t \in [0, T],$$

### Sec. 3.3 The Pontryagin Minimum Principle

so minimization of the Hamiltonian gives

$$u^*(t) = \arg \min_{u \in \mathfrak{U}} \sqrt{1 + u^2} = 0, \quad \text{for all } t \in [0, T].$$

Therefore we have  $\dot{x}^*(t) = 0$  for all  $t$ , which implies that  $x^*(t)$  is constant. Using the initial condition  $x^*(0) = \alpha$ , it follows that

$$x^*(t) = \alpha, \quad \text{for all } t \in [0, T].$$

We thus obtain the (a priori obvious) optimal solution, which is the horizontal line passing through  $(0, \alpha)$ . Note that since the Minimum Principle is only a necessary condition for optimality, it does not guarantee that the horizontal line solution is optimal. For such a guarantee, we should invoke the linearity of the system function, and the convexity of the cost function. As mentioned (but not proved) earlier, under these conditions, the Minimum Principle is both necessary and sufficient for optimality.

### Example 3.3.2 (Resource Allocation Continued)

Consider the optimal production problem (Example 3.1.2). We want to maximize

$$\int_0^T (1 - u(t)) x(t) dt$$

subject to

$$0 \leq u(t) \leq 1, \quad \text{for all } t \in [0, T],$$

$$\dot{x}(t) = \gamma u(t)x(t), \quad x(0) > 0 : \text{given.}$$

The Hamiltonian is

$$H(x, u, p) = (1 - u)x + p\gamma ux.$$

The adjoint equation is

$$\begin{aligned} \dot{p}(t) &= -\gamma u^*(t)p(t) - 1 + u^*(t), \\ p(T) &= 0. \end{aligned}$$

Maximization of the Hamiltonian over  $u \in [0, 1]$  yields

$$u^*(t) = \begin{cases} 0 & \text{if } p(t) < \frac{1}{\gamma}, \\ 1 & \text{if } p(t) \geq \frac{1}{\gamma}. \end{cases}$$

Since  $p(T) = 0$ , for  $t$  close to  $T$  we will have  $p(t) < 1/\gamma$  and  $u^*(t) = 0$ . Therefore, for  $t$  near  $T$  the adjoint equation has the form  $\dot{p}(t) = -1$  and  $p(t)$  has the form shown in Fig. 3.3.1.

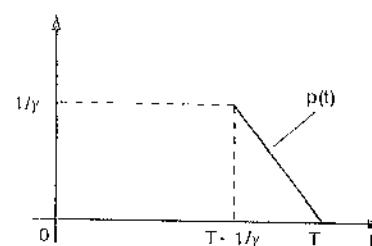


Figure 3.3.1 Form of the adjoint variable  $p(t)$  for  $t$  near  $T$  in the resource allocation example.

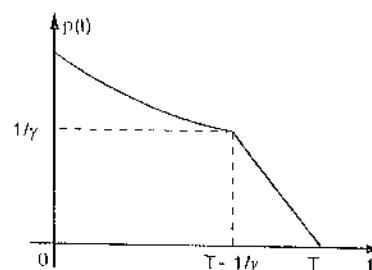
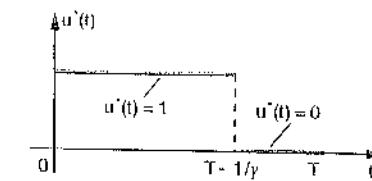


Figure 3.3.2 Form of the adjoint variable  $p(t)$  and the optimal control in the resource allocation example.



Thus, near  $t = T$ ,  $p(t)$  decreases with slope  $-1$ . For  $t = T - 1/\gamma$ ,  $p(t)$  is equal to  $1/\gamma$ , so  $u^*(t)$  changes to  $u^*(t) = 1$ . It follows that for  $t < T - 1/\gamma$ , the adjoint equation is

$$\dot{p}(t) = -\gamma p(t)$$

or

$$p(t) = e^{-\gamma t} \cdot \text{constant}.$$

Piecing together  $p(t)$  for  $t$  greater and less than  $T - 1/\gamma$ , we obtain the form shown in Fig. 3.3.2 for  $p(t)$  and  $u^*(t)$ . Note that if  $T < 1/\gamma$ , the optimal control is  $u^*(t) = 0$  for all  $t \in [0, T]$ ; that is, for a short enough horizon, it does not pay to reinvest at any time.

### Example 3.3.3 (A Linear-Quadratic Problem)

Consider the one-dimensional linear system

$$\dot{x}(t) = ax(t) + bu(t),$$

where  $a$  and  $b$  are given scalars. We want to find an optimal control over a given interval  $[0, T]$  that minimizes the quadratic cost

$$\frac{1}{2}q \cdot (x(T))^2 + \frac{1}{2} \int_0^T (u(t))^2 dt,$$

where  $q$  is a given positive scalar. There are no constraints on the control, so we have a special case of the linear-quadratic problem of Example 3.2.2. We will solve this problem via the Minimum Principle.

The Hamiltonian here is

$$H(x, u, p) = \frac{1}{2}u^2 + p(ax + bu),$$

and the adjoint equation is

$$\dot{p}(t) = -ap(t),$$

with the terminal condition

$$p(T) = qx^*(T).$$

The optimal control is obtained by minimizing the Hamiltonian with respect to  $u$ , yielding

$$u^*(t) = \arg \min_u \left[ \frac{1}{2}u^2 + p(t)(ax^*(t) + bu) \right] \dots \rightarrow b_p(t). \quad (3.25)$$

We will extract the optimal solution from these conditions using two different approaches.

In the first approach, we solve the two-point boundary value problem discussed following Prop. 3.3.1. In particular, by eliminating the control from the system equation using Eq. (3.25), we obtain

$$\dot{x}^*(t) = ax^*(t) - b^2 p(t).$$

Also, from the adjoint equation, we see that

$$p(t) = e^{-at}\xi, \quad \text{for all } t \in [0, T],$$

where  $\xi = p(0)$  is an unknown parameter. The last two equations yield

$$x^*(t) = ax^*(t) - b^2 e^{-at}\xi. \quad (3.26)$$

This differential equation, together with the given initial condition  $x^*(0) = x(0)$  and the terminal condition

$$x^*(T) = \frac{e^{-aT}\xi}{q},$$

(which is the terminal condition for the adjoint equation) can be solved for the unknown variable  $\xi$ . In particular, it can be verified that the solution of the differential equation (3.26) is given by

$$x^*(t) = x(0)e^{at} + \frac{b^2\xi}{2a}(e^{-at} - e^{at}),$$

and  $\xi$  can be obtained from the last two relations. Given  $\xi$ , we obtain  $p(t) = e^{-at}\xi$ , and from  $p(t)$ , we can then determine the optimal control trajectory as  $u^*(t) = -bp(t)$ ,  $t \in [0, T]$  [cf. Eq. (3.25)].

In the second approach, we basically derive the Riccati equation encountered in Example 3.2.2. In particular, we hypothesize a linear relation between  $x^*(t)$  and  $p(t)$ , that is,

$$K(t)x^*(t) = p(t), \quad \text{for all } t \in [0, T],$$

and we show that  $K(t)$  can be obtained by solving the Riccati equation. Indeed, from Eq. (3.25) we have

$$u^*(t) = -bK(t)x^*(t),$$

which by substitution in the system equation, yields

$$\dot{x}^*(t) = (a - b^2 K(t))x^*(t).$$

By differentiating the equation  $K(t)x^*(t) = p(t)$  and by also using the adjoint equation, we obtain

$$\dot{K}(t)x^*(t) + K(t)\dot{x}^*(t) = \dot{p}(t) = -ap(t) = -aK(t)x^*(t).$$

By combining the last two relations, we have

$$\dot{K}(t)x^*(t) + K(t)(a - b^2 K(t))x^*(t) = -aK(t)x^*(t),$$

from which we see that  $K(t)$  should satisfy

$$\dot{K}(t) = -2aK(t) + b^2(K(t))^2.$$

This is the Riccati equation of Example 3.2.2, specialized to the problem of the present example. This equation can be solved using the terminal condition

$$K(T) = q,$$

which is implied by the terminal condition  $p(T) = qx^*(T)$  for the adjoint equation. Once  $K(t)$  is known, the optimal control is obtained in the closed-loop form  $u^*(t) = -bK(t)x^*(t)$ . By reversing the preceding arguments, this control can then be shown to satisfy all the conditions of the Minimum Principle.

### 3.3.2 A Derivation Based on Variational Ideas

In this subsection we outline an alternative and more rigorous proof of the Minimum Principle. This proof is primarily directed towards the advanced reader, and is based on making small variations in the optimal trajectory and comparing it with neighboring trajectories.

For convenience, we restrict attention to the case where the cost is

$$h(x(T)).$$

The more general cost

$$h(x(T)) + \int_0^T g(x(t), u(t)) dt \quad (3.27)$$

can be reformulated as a terminal cost by introducing a new state variable  $y$  and the additional differential equation

$$\dot{y}(t) = g(x(t), u(t)). \quad (3.28)$$

The cost then becomes

$$h(x(T)) + y(T), \quad (3.29)$$

and the Minimum Principle corresponding to this terminal cost yields the Minimum Principle for the general cost (3.27).

We introduce some assumptions:

**Convexity Assumption:** For every state  $x$  the set

$$D = \{f(x, u) \mid u \in U\}$$

is convex.

The convexity assumption is satisfied if  $U$  is a convex set and  $f$  is linear in  $u$  [and  $g$  is linear in  $u$  in the case where there is an integral cost of the form (3.27), which is reformulated as a terminal cost by using the additional state variable  $y$  of Eq. (3.28)]. Thus the convexity assumption is quite restrictive. However, the Minimum Principle typically holds without the convexity assumption, because even when the set  $D = \{f(x, u) \mid u \in U\}$  is nonconvex, any vector in the convex hull of  $D$  can be generated by quick alternation between vectors from  $D$  (for an example, see Exercise 3.10). This involves the complicated mathematical concept of *randomized* or *relaxed* controls and will not be discussed further.

**Regularity Assumption:** Let  $u(t)$  and  $u^*(t)$ ,  $t \in [0, T]$ , be any two admissible control trajectories and let  $\{x^*(t) \mid t \in [0, T]\}$  be the state trajectory corresponding to  $u^*(t)$ . For any  $\epsilon \in [0, 1]$ , the solution  $\{x_\epsilon(t) \mid t \in [0, T]\}$  of the system

$$\dot{x}_\epsilon(t) = (1 - \epsilon)f(x_\epsilon(t), u^*(t)) + \epsilon f(x_\epsilon(t), u(t)), \quad (3.30)$$

with  $x_\epsilon(0) = x^*(0)$ , satisfies

$$x_\epsilon(t) = x^*(t) + \epsilon \xi(t) + o(\epsilon), \quad (3.31)$$

where  $\{\xi(t) \mid t \in [0, T]\}$  is the solution of the linear differential system

$$\dot{\xi}(t) = \nabla_x f(x^*(t), u^*(t)) \xi(t) + f(x^*(t), u(t)) - f(x^*(t), u^*(t)), \quad (3.32)$$

with initial condition  $\xi(0) = 0$ .

The regularity assumption “typically” holds because from Eq. (3.30) we have

$$\begin{aligned} \dot{x}_\epsilon(t) - \dot{x}^*(t) &= f(x_\epsilon(t), u^*(t)) - f(x^*(t), u^*(t)) \\ &\quad + \epsilon \left( f(x_\epsilon(t), u(t)) - f(x_\epsilon(t), u^*(t)) \right), \end{aligned}$$

so from a first order Taylor series expansion we obtain

$$\begin{aligned} \delta \dot{x}(t) &= \nabla f(x^*(t), u^*(t))' \delta x(t) + o(\|\delta x(t)\|) \\ &\quad + \epsilon \left( f(x_\epsilon(t), u(t)) - f(x_\epsilon(t), u^*(t)) \right), \end{aligned}$$

where

$$\delta x(t) = x_\epsilon(t) - x^*(t).$$

Dividing by  $\epsilon$  and taking the limit as  $\epsilon \rightarrow 0$ , we see that the function

$$\xi(t) = \lim_{\epsilon \rightarrow 0} \delta x(t)/\epsilon, \quad t \in [0, T],$$

should “typically” solve the linear system of differential equations (3.32), while satisfying Eq. (3.31).

In fact, if the system is linear of the form

$$\dot{x}(t) = Ax(t) + Bu(t),$$

where  $A$  and  $B$  are given matrices, it can be shown that the regularity assumption is satisfied. To see this, note that Eqs. (3.30) and (3.32) take the forms

$$\dot{x}_\epsilon(t) = Ax_\epsilon(t) + Bu^*(t) + \epsilon B(u(t) - u^*(t)),$$

and

$$\dot{\xi}(t) = A\xi(t) + B(u(t) - u^*(t)),$$

respectively. Thus, taking into account the initial conditions  $x_\epsilon(0) = x^*(0)$  and  $\xi(0) = 0$ , we see that

$$x_\epsilon(t) = x^*(t) + \epsilon \xi(t), \quad t \in [0, T],$$

so the regularity condition (3.31) is satisfied.

We now prove the Minimum Principle assuming the convexity and regularity assumptions above. Suppose that  $\{u^*(t) \mid t \in [0, T]\}$  is an optimal control trajectory, and let  $\{x^*(t) \mid t \in [0, T]\}$  be the corresponding state trajectory. Then for any other admissible control trajectory  $\{u(t) \mid t \in [0, T]\}$  and any  $\epsilon \in [0, 1]$ , the convexity assumption guarantees that for each  $t$ , there exists a control  $\bar{u}(t) \in U$  such that

$$f(x_\epsilon(t), \bar{u}(t)) = (1 - \epsilon)f(x_\epsilon(t), u^*(t)) + \epsilon f(x_\epsilon(t), u(t)).$$

Thus, the state trajectory  $\{x_\epsilon(t) \mid t \in [0, T]\}$  of Eq. (3.30) corresponds to the admissible control trajectory  $\{\bar{u}(t) \mid t \in [0, T]\}$ . Hence, using the optimality of  $\{x^*(t) \mid t \in [0, T]\}$  and the regularity assumption, we have

$$\begin{aligned} h(x^*(T)) &< h(x_\epsilon^*(T)) \\ &= h(x^*(T) + \epsilon \xi(T) + o(\epsilon)) \\ &= h(x^*(T)) + \epsilon \nabla h(x^*(T))' \xi(T) + o(\epsilon), \end{aligned}$$

which implies that

$$\nabla h(x^*(T))' \xi(T) \geq 0. \quad (3.33)$$

Using a standard result in the theory of linear differential equations (see e.g. [CoL65]), the solution of the linear differential system (3.32) can be written in closed form as

$$\xi(t) = \Phi(t, \tau) \xi(\tau) + \int_\tau^t \Phi(t, \tau) \left( f(x^*(\tau), u(\tau)) - f(x^*(\tau), u^*(\tau)) \right) d\tau, \quad (3.34)$$

where the square matrix  $\Phi$  satisfies for all  $t$  and  $\tau$ ,

$$\frac{\partial \Phi(t, \tau)}{\partial \tau} = -\Phi(t, \tau) \nabla_x f(x^*(\tau), u^*(\tau))', \quad (3.35)$$

$$\Phi(t, t) = I.$$

Since  $\xi(0) = 0$ , we have from Eq. (3.34),

$$\xi(T) = \int_0^T \Phi(T, t) \left( f(x^*(t), u(t)) - f(x^*(t), u^*(t)) \right) dt. \quad (3.36)$$

Define

$$p(T) = \nabla h(x^*(T)), \quad p(t) = \Phi(T, t)' p(T), \quad t \in [0, T]. \quad (3.37)$$

By differentiating with respect to  $t$ , we obtain

$$\dot{p}(t) = \frac{\partial \Phi(T, t)'}{\partial t} p(T).$$

Combining this equation with Eqs. (3.35) and (3.37), we see that  $p(t)$  is generated by the differential equation

$$\dot{p}(t) = -\nabla_x f(x^*(t), u^*(t)) p(t),$$

with the terminal condition

$$p(T) = \nabla h(x^*(T)).$$

This is the adjoint equation corresponding to  $\{(x^*(t), u^*(t)) \mid t \in [0, T]\}$ .

Now, to obtain the Minimum Principle, we note that from Eqs. (3.33), (3.36), and (3.37) we have

$$\begin{aligned} 0 &\leq p(T)' \xi(T) \\ &= p(T)' \int_0^T \Phi(T, t) (f(x^*(t), u(t)) - f(x^*(t), u^*(t))) dt \\ &= \int_0^T p(t)' (f(x^*(t), u(t)) - f(x^*(t), u^*(t))) dt, \end{aligned} \quad (3.38)$$

from which it can be shown that for all  $t$  at which  $u^*(\cdot)$  is continuous, we have

$$p(t)' f(x^*(t), u^*(t)) \leq p(t)' f(x^*(t), u), \quad \text{for all } u \in U. \quad (3.39)$$

Indeed, if for some  $\hat{u} \in U$  and  $t_0 \in [0, T]$ , we have

$$p(t_0)' f(x^*(t_0), u^*(t_0)) > p(t_0)' f(x^*(t_0), \hat{u}),$$

while  $\{u^*(t) \mid t \in [0, T]\}$  is continuous at  $t_0$ , we would also have

$$p(t)' f(x^*(t), u^*(t)) > p(t)' f(x^*(t), \hat{u}),$$

for all  $t$  in some nontrivial interval  $I$  containing  $t_0$ . By taking

$$u(t) = \begin{cases} \hat{u} & \text{for } t \in I, \\ u^*(t) & \text{for } t \notin I, \end{cases}$$

we would then obtain a contradiction of Eq. (3.38).

We have thus proved the Minimum Principle (3.39) under the convexity and regularity assumptions, and the assumption that there is only a terminal cost  $h(x(T))$ . We have also seen that in the case where the constraint set  $U$  is convex and the system is linear, the convexity and regularity assumptions are satisfied. To prove the Minimum Principle for the more general integral cost function (3.27), we can apply the preceding development to the system of differential equations  $\dot{x} = f(x, u)$  augmented by the additional Eq. (3.28) and the equivalent terminal cost (3.29). The corresponding convexity and regularity assumptions are automatically satisfied if the constraint set  $U$  is convex and the system function  $f(x, u)$  as well as the cost function  $g(x, u)$  are linear. This is necessary in order to maintain the linearity of the augmented system, thereby maintaining the validity of the regularity assumption.

### 3.3.3 Minimum Principle for Discrete-Time Problems

In this subsection we briefly derive a version of the Minimum Principle for discrete-time deterministic optimal control problems. Interestingly, it is essential to make some convexity assumptions in order for the Minimum Principle to hold. For continuous-time problems these convexity assumptions are typically not needed, because, as mentioned earlier, the differential system can generate any  $\dot{x}(t)$  in the convex hull of the set of possible vectors  $f(x(t), u(t))$  by quick alternation between different controls (see for example Exercise 3.10).

Suppose that we want to find a control sequence  $(u_0, u_1, \dots, u_{N-1})$  and a corresponding state sequence  $(x_0, x_1, \dots, x_N)$ , which minimize

$$J(u) = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k),$$

subject to the discrete-time system constraints

$$x_{k+1} = f_k(x_k, u_k), \quad k = 0, \dots, N-1, \quad x_0 : \text{given},$$

and the control constraints

$$u_k \in U_k \subset \mathbb{R}^m, \quad k = 0, \dots, N-1.$$

We first develop an expression for the gradient  $\nabla J(u_0, \dots, u_{N-1})$ . We have, using the chain rule,

$$\begin{aligned} \nabla_{u_{N-1}} J(u_0, \dots, u_{N-1}) &= \nabla_{u_{N-1}} \left( g_N(f_{N-1}(x_{N-1}, u_{N-1})) \right. \\ &\quad \left. + g_{N-1}(x_{N-1}, u_{N-1}) \right) \\ &= \nabla_{u_{N-1}} f_{N-1} \cdot \nabla g_N + \nabla_{u_{N-1}} g_{N-1}, \end{aligned}$$

where all gradients are evaluated along the control trajectory  $(u_0, \dots, u_{N-1})$  and the corresponding state trajectory. Similarly, for all  $k$ ,

$$\begin{aligned}\nabla_{u_k} J(u_0, \dots, u_{N-1}) &= \nabla_{u_k} f_k \cdot \nabla_{x_{k+1}} f_{k+1} \cdots \nabla_{x_{N-1}} f_{N-1} \cdot \nabla g_N \\ &\quad + \nabla_{u_k} f_k \cdot \nabla_{x_{k+1}} f_{k+1} \cdots \nabla_{x_{N-2}} f_{N-2} \cdot \nabla_{x_{N-1}} g_{N-1} \\ &\quad \cdots \\ &\quad + \nabla_{u_k} f_k \cdot \nabla_{x_{k+1}} g_{k+1} \\ &\quad + \nabla_{u_k} g_k,\end{aligned}\tag{3.40}$$

which can be written in the form

$$\nabla_{u_k} J(u_0, \dots, u_{N-1}) = \nabla_{u_k} f_k \cdot p_{k+1} + \nabla_{u_k} g_k,$$

for an appropriate vector  $p_{k+1}$ , or

$$\nabla_{u_k} J(u_0, \dots, u_{N-1}) = \nabla_{u_k} H_k(x_k, u_k, p_{k+1}),\tag{3.41}$$

where  $H_k$  is the Hamiltonian function defined by

$$H_k(x_k, u_k, p_{k+1}) = g_k(x_k, u_k) + p'_{k+1} f_k(x_k, u_k).$$

It can be seen from Eq. (3.40) that the vectors  $p_{k+1}$  are generated backwards by the *discrete-time adjoint equation*

$$p_k = \nabla_{x_k} f_k \cdot p_{k+1} + \nabla_{x_k} g_k, \quad k = 1, \dots, N-1,$$

with terminal condition

$$p_N = \nabla g_N.$$

We will assume that the constraint sets  $U_k$  are convex, so that we can apply the optimality condition

$$\sum_{k=0}^{N-1} \nabla_{u_k} J(u_0^*, \dots, u_{N-1}^*)'(u_k - u_k^*) \geq 0,$$

for all feasible  $(u_0, \dots, u_{N-1})$  (see Appendix B). This condition can be decomposed into the  $N$  conditions

$$\nabla_{u_k} J(u_0^*, \dots, u_{N-1}^*)'(u_k - u_k^*) \geq 0, \quad \text{for all } u_k \in U_k, \quad k = 0, \dots, N-1.\tag{3.42}$$

We thus obtain:

**Proposition 3.3.2: (Discrete-Time Minimum Principle)** Suppose that  $(u_0^*, u_1^*, \dots, u_{N-1}^*)$  is an optimal control trajectory and that  $(x_0^*, x_1^*, \dots, x_N^*)$  is the corresponding state trajectory. Assume also that the constraint sets  $U_k$  are convex. Then for all  $k = 0, \dots, N-1$ , we have

$$\nabla_{u_k} H_k(x_k^*, u_k^*, p_{k+1})'(u_k - u_k^*) \geq 0, \quad \text{for all } u_k \in U_k,\tag{3.43}$$

where the vectors  $p_1, \dots, p_N$  are obtained from the adjoint equation

$$p_k = \nabla_{x_k} f_k \cdot p_{k+1} + \nabla_{x_k} g_k, \quad k = 1, \dots, N-1,$$

with the terminal condition

$$p_N = \nabla g_N(x_N^*).$$

The partial derivatives above are evaluated along the optimal state and control trajectories. If, in addition, the Hamiltonian  $H_k$  is a convex function of  $u_k$  for any fixed  $x_k$  and  $p_{k+1}$ , we have

$$u_k^* = \arg \min_{u_k \in U_k} H_k(x_k^*, u_k, p_{k+1}), \quad \text{for all } k = 0, \dots, N-1.\tag{3.44}$$

**Proof:** Equation (3.43) is a restatement of the necessary condition (3.42) using the expression (3.41) for the gradient of  $J$ . If  $H_k$  is convex with respect to  $u_k$ , Eq. (3.42) is a sufficient condition for the minimum condition (3.44) to hold (see Appendix B). **Q.E.D.**

## 3.4 EXTENSIONS OF THE MINIMUM PRINCIPLE

We now consider some variations of the continuous-time optimal control problem and derive corresponding variations of the Minimum Principle.

### 3.4.1 Fixed Terminal State

Suppose that in addition to the initial state  $x(0)$ , the final state  $x(T)$  is given. Then the preceding informal derivations still hold except that the terminal condition  $J^*(T, x) = h(x)$  is not true anymore. In effect, here we have

$$J^*(T, x) = \begin{cases} 0 & \text{if } x = x(T), \\ \infty & \text{otherwise.} \end{cases}$$

Thus  $J^*(T, x)$  cannot be differentiated with respect to  $x$ , and the terminal boundary condition  $p(T) = \nabla h(x^*(T))$  for the adjoint equation does not hold. However, as compensation, we have the extra condition

$$x(T) : \text{given},$$

thus maintaining the balance between boundary conditions and unknowns.

If only *some* of the terminal states are fixed, that is,

$$x_i(T) : \text{given}, \quad \text{for all } i \in I,$$

where  $I$  is some index set, we have the partial boundary condition

$$p_j(T) = \frac{\partial h(x^*(T))}{\partial x_j}, \quad \text{for all } j \notin I,$$

for the adjoint equation.

### Example 3.4.1

Consider the problem of finding the curve of minimum length connecting two points  $(0, \alpha)$  and  $(T, \beta)$ . This is a fixed endpoint variation of Example 3.3.1 in the preceding section. We have

$$\dot{x}(t) = u(t),$$

$$x(0) = \alpha, \quad x(T) = \beta,$$

and the cost is

$$\int_0^T \sqrt{1 + (u(t))^2} dt.$$

The adjoint equation is

$$\dot{p}(t) = 0,$$

implying that

$$p(t) = \text{constant}, \quad \text{for all } t \in [0, T].$$

Minimization of the Hamiltonian,

$$\min_{u \in \mathbb{R}} \left[ \sqrt{1 + u^2} + p(t)u \right],$$

yields

$$u^*(t) = \text{constant}, \quad \text{for all } t \in [0, T].$$

Thus the optimal trajectory  $\{x^*(t) \mid t \in [0, T]\}$  is a straight line. Since this trajectory must pass through  $(0, \alpha)$  and  $(T, \beta)$ , we obtain the (a priori obvious) optimal solution shown in Fig. 3.4.1.

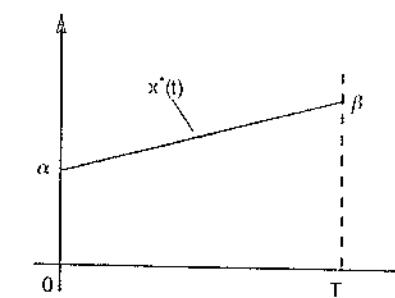


Figure 3.4.1 Optimal solution of the problem of connecting the two points  $(0, \alpha)$  and  $(T, \beta)$  with a minimum length curve (cf. Example 3.4.1).

### Example 3.4.2 (The Brachistochrone Problem)

In 1696 Johann Bernoulli challenged the mathematical world of his time with a problem that played an instrumental role in the development of the calculus of variations: Given two points A and B, find a curve connecting A and B such that a body moving along the curve under the force of gravity reaches B in minimum time (see Fig. 3.4.2). Let A be  $(0, 0)$  and B be  $(T, -b)$  with  $b > 0$ . Then it can be seen that the problem is to find  $\{x(t) \mid t \in [0, T]\}$  with  $x(0) = 0$  and  $x(T) = b$ , which minimizes

$$\int_0^T \frac{\sqrt{1 + (\dot{x}(t))^2}}{\sqrt{2\gamma x(t)}} dt,$$

where  $\gamma$  is the acceleration due to gravity. Here  $\{(t, -x(t)) \mid t \in [0, T]\}$ , is the desired curve, the term  $\sqrt{1 + (\dot{x}(t))^2} dt$  is the length of the curve from  $x(t)$  to  $x(t + dt)$ , and the term  $\sqrt{2\gamma x(t)}$  is the velocity of the body upon reaching the level  $x(t)$  [if  $m$  and  $v$  denote the mass and the velocity of the body, the kinetic energy is  $mv^2/2$ , which at level  $x(t)$  must be equal to the change in potential energy, which is  $mgy(t)$ ; this yields  $v = \sqrt{2\gamma x(t)}$ ].

We introduce the system  $\dot{x} = u$ , and we obtain a fixed terminal state problem [ $x(0) = 0$  and  $x(T) = b$ ]. Letting

$$g(x, u) = \frac{\sqrt{1 + u^2}}{\sqrt{2\gamma x}},$$

the Hamiltonian is

$$H(x, u, p) = g(x, u) + pu.$$

We minimize the Hamiltonian by setting to zero its derivative with respect to  $u$ :

$$p(t) = -\nabla_u g(x^*(t), u^*(t)).$$

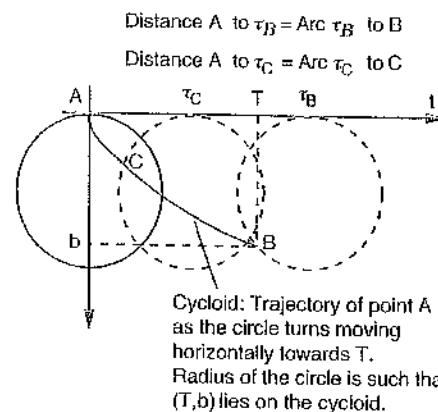


Figure 3.4.2 Formulation and optimal solution of the brachistochrone problem.

We know from the Minimum Principle that the Hamiltonian is constant along an optimal trajectory, i.e.,

$$g(x^*(t), u^*(t)) - \nabla_u g(x^*(t), u^*(t)) u^*(t) = \text{constant}, \quad \text{for all } t \in [0, T].$$

Using the expression for  $g$ , this can be written as

$$\sqrt{1 + (u^*(t))^2} - \frac{(u^*(t))^2}{\sqrt{2\gamma x^*(t)}} = \text{constant}, \quad \text{for all } t \in [0, T],$$

or equivalently

$$\frac{1}{\sqrt{1 + (u^*(t))^2} \sqrt{2\gamma x^*(t)}} = \text{constant}, \quad \text{for all } t \in [0, T].$$

Using the relation  $\dot{x}^*(t) = u^*(t)$ , this yields

$$\dot{x}^*(t)(1 + \dot{x}^*(t)^2) = C, \quad \text{for all } t \in [0, T],$$

for some constant  $C$ . Thus an optimal trajectory satisfies the differential equation

$$\dot{x}^*(t) = \sqrt{\frac{C - x^*(t)}{x^*(t)}}, \quad \text{for all } t \in [0, T].$$

The solution of this differential equation was known at Bernoulli's time to be a *cycloid*; see Fig. 3.4.2. The unknown parameters of the cycloid are determined by the boundary conditions  $x^*(0) = 0$  and  $x^*(T) = b$ .

### 3.4.2 Free Initial State

If the initial state  $x(0)$  is not fixed but is subject to optimization, we have

$$J^*(0, x^*(0)) \leq J^*(0, x), \quad \text{for all } x \in \mathbb{R}^n,$$

yielding

$$\nabla_x J^*(0, x^*(0)) = 0$$

and the extra boundary condition for the adjoint equation

$$p(0) = 0.$$

Also if there is a cost  $\ell(x(0))$  on the initial state, i.e., the cost is

$$\ell(x(0)) + \int_0^T g(x(t), u(t)) dt + h(x(T)),$$

the boundary condition becomes

$$p(0) = -\nabla \ell(x^*(0)).$$

This follows by setting to zero the gradient with respect to  $x$  of  $\ell(x) + J(0, x)$ , i.e.,

$$\nabla_x \{\ell(x) + J(0, x)\}|_{x=x^*(0)} = 0.$$

### 3.4.3 Free Terminal Time

Suppose the initial state and/or the terminal state are given, but the terminal time  $T$  is subject to optimization.

Let  $\{(x^*(t), u^*(t)) \mid t \in [0, T]\}$  be an optimal state-control trajectory pair and let  $T^*$  be the optimal terminal time. Then if the terminal time were fixed at  $T^*$ , the pair  $\{(u^*(t), x^*(t)) \mid t \in [0, T^*]\}$  would satisfy the conditions of the Minimum Principle. In particular,

$$u^*(t) = \arg \min_{u \in U} H(x^*(t), u, p(t)), \quad \text{for all } t \in [0, T^*],$$

where  $p(t)$  is the solution of the adjoint equation. What we lose with the terminal time being free, we gain with an extra condition derived as follows.

We argue that if the terminal time were fixed at  $T^*$  and the initial state were fixed at the given  $x(0)$ , but instead the initial time were subject to optimization, it would be optimal to start at  $t = 0$ . This means that the first order variation of the optimal cost with respect to the initial time must be zero; i.e.,

$$\nabla_t J^*(t, x^*(t))|_{t=0} = 0.$$

The HJB equation can be written along the optimal trajectory as

$$\nabla_t J^*(t, x^*(t)) = -H(x^*(t), u^*(t), p(t)), \quad \text{for all } t \in [0, T^*]$$

[cf. Eqs. (3.14) and (3.19)], so the preceding two equations yield

$$H(x^*(0), u^*(0), p(0)) = 0.$$

Since the Hamiltonian was shown earlier to be constant along the optimal trajectory, we obtain for the case of a free terminal time

$$H(x^*(t), u^*(t), p(t)) = 0, \quad \text{for all } t \in [0, T^*].$$

### Example 3.4.3 (Minimum-Time Problem)

A unit mass object moves horizontally under the influence of a force  $u(t)$ , so that

$$\ddot{y}(t) = u(t),$$

where  $y(t)$  is the position of the object at time  $t$ . Given the object's initial position  $y(0)$  and initial velocity  $\dot{y}(0)$ , it is required to bring the object to rest (zero velocity) at a given position, say zero, while using at most unit magnitude force,

$$-1 \leq u(t) \leq 1, \quad \text{for all } t.$$

We want to accomplish this transfer in minimum time. Thus, we want to

$$\text{minimize } T = \int_0^T 1 dt.$$

Note that the integral cost,  $g(x(t), u(t)) = 1$ , is unusual here; it does not depend on the state or the control. However, the theory does not preclude this possibility, and the problem is still meaningful because the terminal time  $T$  is free and subject to optimization.

Let the state variables be

$$x_1(t) = y(t), \quad x_2(t) = \dot{y}(t),$$

so the system equation is

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = u(t).$$

The initial state  $(x_1(0), x_2(0))$  is given and the terminal state is also given

$$x_1(T) = 0, \quad x_2(T) = 0.$$

If  $\{u^*(t) \mid t \in [0, T]\}$  is an optimal control trajectory,  $u^*(t)$  must minimize the Hamiltonian for each  $t$ , i.e.,

$$u^*(t) := \arg \min_{-1 \leq u \leq 1} [1 + p_1(t)x_2^*(t) + p_2(t)u].$$

Therefore

$$u^*(t) = \begin{cases} 1 & \text{if } p_2(t) < 0, \\ -1 & \text{if } p_2(t) \geq 0. \end{cases}$$

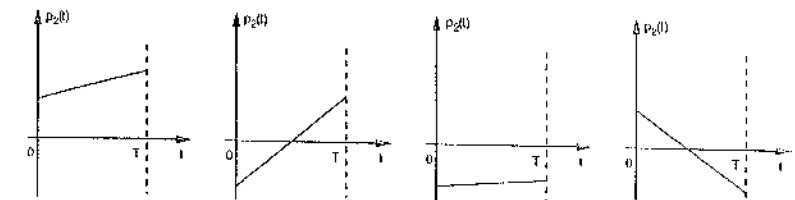
The adjoint equation is

$$\dot{p}_1(t) = 0, \quad \dot{p}_2(t) = -p_1(t),$$

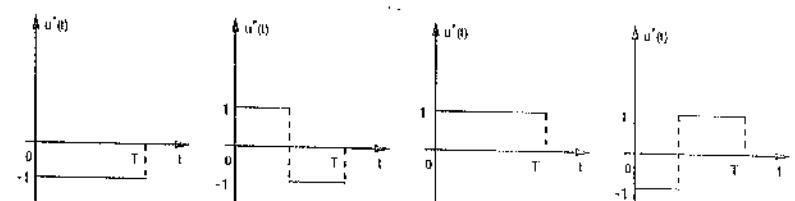
so

$$p_1(t) = c_1, \quad p_2(t) = c_2 - c_1 t,$$

where  $c_1$  and  $c_2$  are constants. It follows that  $\{p_2(t) \mid t \in [0, T]\}$  has one of the four forms shown in Fig. 3.4.3(a); that is,  $\{p_2(t) \mid t \in [0, T]\}$  switches at most once in going from negative to positive or vice versa. [Note that it is not possible for  $p_2(t)$  to be equal to 0 for all  $t$  because this implies that  $p_1(t)$  is also equal to 0 for all  $t$ , so that the Hamiltonian is equal to 1 for all  $t$ ; the necessary conditions require that the Hamiltonian be 0 along the optimal trajectory.] The corresponding control trajectories are shown in Fig. 3.4.3(b). The conclusion is that, for each  $t$ ,  $u^*(t)$  is either +1 or -1, and  $\{u^*(t) \mid t \in [0, T]\}$  has at most one switching point in the interval  $[0, T]$ .



(a)



(b)

Figure 3.4.3 (a) Possible forms of the adjoint variable  $p_2(t)$ . (b) Corresponding forms of the optimal control trajectory.

To determine the precise form of the optimal control trajectory, we use the given initial and final states. For  $u(t) \equiv \zeta$ , where  $\zeta = \pm 1$ , the system evolves according to

$$x_1(t) = x_1(0) + x_2(0)t + \frac{\zeta}{2}t^2, \quad x_2(t) = x_2(0) + \zeta t.$$

By eliminating the time  $t$  in these two equations, we see that for all  $t$

$$x_1(t) - \frac{1}{2\zeta} (x_2(t))^2 = x_1(0) - \frac{1}{2\zeta} (x_2(0))^2.$$

Thus for intervals where  $u(t) \equiv 1$ , the system moves along the curves where  $x_1(t) - \frac{1}{2}(x_2(t))^2$  is constant, shown in Fig. 3.4.4(a). For intervals where  $u(t) \equiv -1$ , the system moves along the curves where  $x_1(t) + \frac{1}{2}(x_2(t))^2$  is constant, shown in Fig. 3.4.4(b).

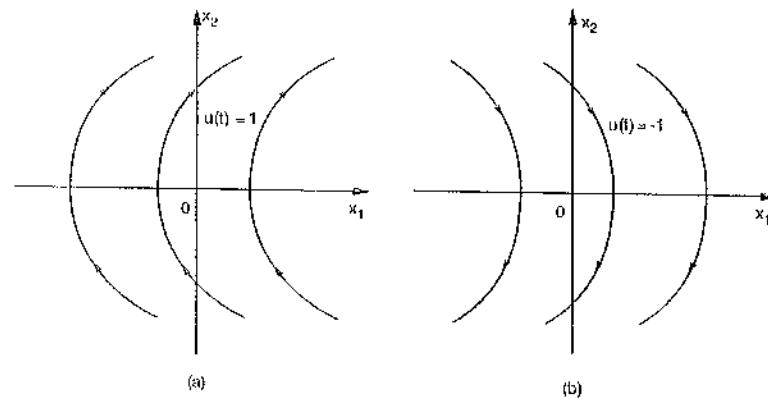


Figure 3.4.4 State trajectories when the control is  $u(t) \equiv 1$  [Fig. (a)] and when the control is  $u(t) \equiv -1$  [Fig. (b)].

To bring the system from the initial state  $(x_1(0), x_2(0))$  to the origin with at most one switch in the value of control, we must apply control according to the following rules involving the *switching curve* shown in Fig. 3.4.5.

- If the initial state lies *above* the switching curve, use  $u^*(t) \equiv -1$  until the state hits the switching curve; then use  $u^*(t) \equiv 1$  until reaching the origin.
- If the initial state lies *below* the switching curve, use  $u^*(t) \equiv 1$  until the state hits the switching curve; then use  $u^*(t) \equiv -1$  until reaching the origin.
- If the initial state lies on the top (bottom) part of the switching curve, use  $u^*(t) \equiv -1$  [ $u^*(t) \equiv 1$ , respectively] until reaching the origin.

#### 3.4.4 Time-Varying System and Cost

If the system equation and the integral cost depend on the time  $t$ , i.e.,

$$\dot{x}(t) = f(x(t), u(t), t),$$

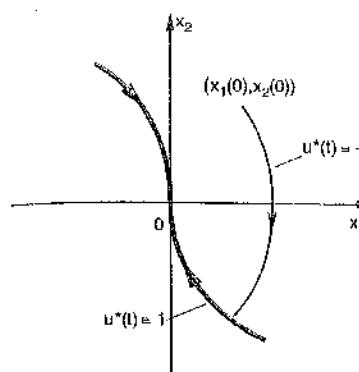


Figure 3.4.5 Switching curve (shown with a thick line) and closed-loop optimal control for the minimum time example.

$$\text{cost.} = h(x(T)) + \int_0^T g(x(t), u(t), t) dt,$$

we can convert the problem to one involving a time-independent system and cost by introducing an extra state variable  $y(t)$  representing time:

$$\dot{y}(t) = 1, \quad y(0) = 0,$$

$$\dot{x}(t) = f(x(t), u(t), y(t)), \quad x(0) : \text{given},$$

$$\text{cost.} = h(x(T)) + \int_0^T g(x(t), u(t), y(t)) dt.$$

After working out the corresponding optimality conditions, we see that they are the same as when the system and cost are time-independent. The only difference is that the Hamiltonian need not be constant along the optimal trajectory.

#### 3.4.5 Singular Problems

In some cases, the minimum condition

$$u^*(t) = \arg \min_{u \in U} H(x^*(t), u, p(t), t) \quad (3.45)$$

is insufficient to determine  $u^*(t)$  for all  $t$ , because the values of  $x^*(t)$  and  $p(t)$  are such that  $H(x^*(t), u, p(t), t)$  is independent of  $u$  over a nontrivial interval of time. Such problems are called *singular*. Their optimal trajectories consist of portions, called *regular arcs*, where  $u^*(t)$  can be determined from the minimum condition (3.45), and other portions, called *singular arcs*, which can be determined from the condition that the Hamiltonian is independent of  $u$ .

### Example 3.4.4 (Road Construction)

Suppose that we want to construct a road over a one-dimensional terrain whose ground elevation (altitude measured from some reference point) is known and is given by  $z(t)$ ,  $t \in [0, T]$ . The elevation of the road is denoted by  $x(t)$ ,  $t \in [0, T]$ , and the difference  $x(t) - z(t)$  must be made up by fill-in or excavation. It is desired to minimize

$$\frac{1}{2} \int_0^T (x(t) - z(t))^2 dt,$$

subject to the constraint that the gradient of the road  $\dot{x}(t)$  lies between  $-a$  and  $a$ , where  $a$  is a specified maximum allowed slope. Thus we have the constraint

$$|u(t)| \leq a, \quad t \in [0, T],$$

where

$$\dot{x}(t) = u(t), \quad t \in [0, T].$$

The adjoint equation here is

$$\dot{p}(t) = -x^*(t) + z(t),$$

with the terminal condition

$$p(T) = 0.$$

Minimization of the Hamiltonian

$$H(x^*(t), u, p(t), t) = \frac{1}{2}(x^*(t) - z(t))^2 + p(t)u$$

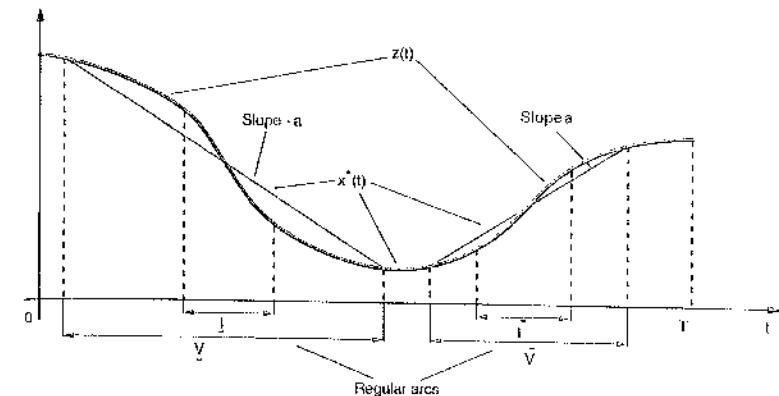
with respect to  $u$  yields

$$u^*(t) = \arg \min_{|u| \leq a} p(t)u,$$

for all  $t$ , and shows that optimal trajectories are obtained by concatenation of three types of arcs:

- (a) Regular arcs where  $p(t) > 0$  and  $u^*(t) = -a$  (maximum downhill slope arcs).
- (b) Regular arcs where  $p(t) < 0$  and  $u^*(t) = a$  (maximum uphill slope arcs).
- (c) Singular arcs where  $p(t) = 0$  and  $u^*(t)$  can take any value in  $[-a, a]$  that maintains the condition  $p(t) = 0$ . From the adjoint equation we see that singular arcs are those along which  $p(t) = 0$  and  $x^*(t) = z(t)$ , i.e., the road follows the ground elevation (no fill-in or excavation). Along such arcs we must have

$$\dot{z}(t) = u^*(t) \in [-a, a].$$



**Figure 3.4.6** Graphical method for solving the road construction example. The sharply uphill (downhill) intervals  $\bar{I}$  (respectively,  $I$ ) are first identified, and are then embedded within maximum uphill (respectively, downhill) slope regular arcs  $\bar{V}$  (respectively,  $V$ ) within which the total fill-in is equal to the total excavation. The regular arcs are joined by singular arcs where there is no fill-in or excavation. The graphical process is started at the endpoint  $t = T$ .

Optimal solutions can be obtained by a graphical method using the above observations. Consider the *sharply uphill intervals*  $\bar{I}$  such that  $\dot{z}(t) \geq a$  for all  $t \in \bar{I}$ , and the *sharply downhill intervals*  $I$  such that  $\dot{z}(t) \leq -a$  for all  $t \in I$ . Clearly, within each sharply uphill interval  $\bar{I}$  the optimal slope is  $u^*(t) = a$ , but the optimal slope is also equal to  $a$  within a larger maximum uphill slope interval  $\bar{V} \supset \bar{I}$ , which is such that  $p(t) < 0$  within  $\bar{V}$  and

$$p(t_1) = p(t_2) = 0$$

at the endpoints  $t_1$  and  $t_2$  of  $\bar{V}$ . In view of the form of the adjoint equation, we see that the endpoints  $t_1$  and  $t_2$  of  $\bar{V}$  should be such that

$$\int_{t_1}^{t_2} (z(t) - x^*(t)) dt = 0;$$

that is, the *total fill-in should be equal to the total excavation within  $\bar{V}$* , (see Fig. 3.4.6). Similarly, each sharply downhill interval  $I$  should be contained within a larger maximum downhill slope interval  $\bar{V} \supset I$ , which is such that  $p(t) > 0$  within  $\bar{V}$ , while the *total fill-in should be equal to the total excavation within  $\bar{V}$* , (see Fig. 3.4.6). Thus the regular arcs consist of the intervals  $\bar{V}$  and  $V$  described above. Between the regular arcs there can be one or more singular arcs where  $x^*(t) = z(t)$ . The optimal solution can be pieced together starting at the endpoint  $t = T$  [where we know that  $p(T) = 0$ ], and proceeding backwards.

### 3.5 NOTES, SOURCES, AND EXERCISES

The calculus of variations is a classical subject that originated with the works of the great mathematicians of the 17th and 18th centuries. Its rigorous development (by modern mathematical standards) took place in the 1930s and 1940s, with the work of a group of mathematicians that originated mostly from the University of Chicago; Bliss, McShane, and Hestenes are some of the most prominent members of this group. Curiously, this development preceded the development of nonlinear programming by many years.<sup>†</sup> The modern theory of deterministic optimal control has its roots primarily in the work of Pontryagin, Boltyanski, Gamkrelidze, and Mishchenko in the 1950s [PBG65]. A highly personal but controversial historical account of this work is given by Boltyanski in [BMS96]. The theoretical and applications literature on the subject is very extensive. We give three representative references: the book by Athans and Falb [AtF66] (a classical extensive text that includes engineering applications), the book by Hestenes [Hes66] (a rigorous mathematical treatment, containing important work that predates the work of Pontryagin et al.), and the book by Luenberger [Lue69] (which deals with optimal control within a broader infinite dimensional context). The author's nonlinear programming book [Ber99] gives a detailed treatment of optimality conditions and computational methods for discrete-time optimal control.

### EXERCISES

#### 3.1

Solve the problem of Example 3.2.1 for the case where the cost function is

$$(x(T))^2 + \int_0^T (u(t))^2 dt.$$

Also, calculate the cost-to-go function  $J^*(t, x)$  and verify that it satisfies the HJB equation.

<sup>†</sup> In the 30s and 40s journal space was at a premium, and finite-dimensional optimization research was thought to be a simple special case of the calculus of variations, thus insufficiently challenging or novel for publication. Indeed the modern optimality conditions of finite-dimensional optimization subject to equality and inequality constraints were first developed in the 1939 Master's thesis by Karush, but first appeared in a journal quite a few years later under the names of other researchers.

#### 3.2

A young investor has earned in the stock market a large amount of money  $S$  and plans to spend it so as to maximize his enjoyment through the rest of his life without working. He estimates that he will live exactly  $T$  more years and that his capital  $x(t)$  should be reduced to zero at time  $T$ , i.e.,  $x(T) = 0$ . Also he models the evolution of his capital by the differential equation

$$\frac{dx(t)}{dt} = \alpha x(t) - u(t),$$

where  $x(0) = S$  is his initial capital,  $\alpha > 0$  is a given interest rate, and  $u(t) \geq 0$  is his rate of expenditure. The total enjoyment he will obtain is given by

$$\int_0^T e^{-\beta t} \sqrt{u(t)} dt.$$

Here  $\beta$  is some positive scalar, which serves to discount future enjoyment. Find the optimal  $\{u(t) | t \in [0, T]\}$ .

#### 3.3

Consider the system of reservoirs shown in Fig. 3.5.1. The system equations are

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) + u(t), \\ \dot{x}_2(t) &= x_1(t),\end{aligned}$$

and the control constraint is  $0 \leq u(t) \leq 1$  for all  $t$ . Initially

$$x_1(0) = x_2(0) = 0.$$

We want to maximize  $x_2(1)$  subject to the constraint  $x_1(1) = 0.5$ . Solve the problem.

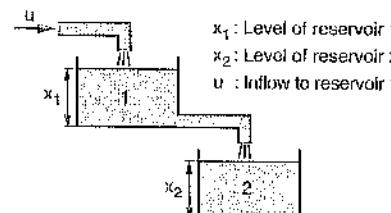


Figure 3.5.1 Reservoir system for Exercise 3.3.

## 3.4

Work out the minimum-time problem (Example 3.4.3) for the case where there is friction and the object's position moves according to

$$\ddot{y}(t) = -a\dot{y}(t) + u(t),$$

where  $a > 0$  is given. Hint: The solution of the system

$$\dot{p}_1(t) = 0,$$

$$\dot{p}_2(t) = -p_1(t) + ap_2(t),$$

is

$$p_1(t) = p_1(0),$$

$$p_2(t) = \frac{1}{a}(1 - e^{at})p_1(0) + e^{at}p_2(0).$$

The trajectories of the system for  $u(t) \equiv -1$  and  $u(t) \equiv 1$  are sketched in Fig. 3.5.2.

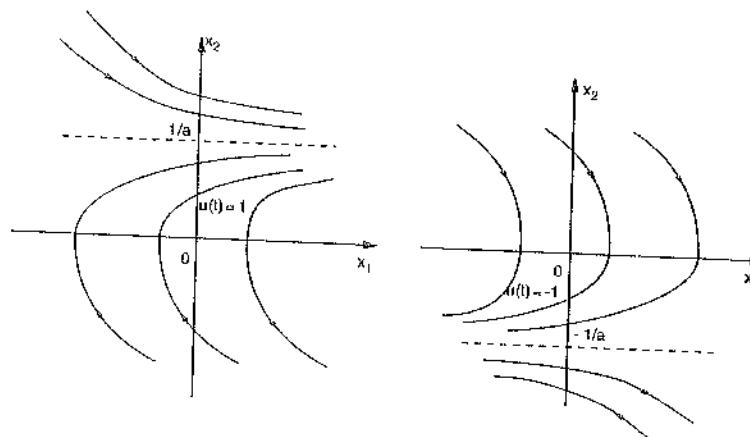


Figure 3.5.2 State trajectories of the system of Exercise 3.4 for  $u(t) \equiv -1$  and  $u(t) \equiv 1$ .

## 3.5 (Isoperimetric Problem)

Analyze the problem of finding a curve  $\{x(t) \mid t \in [0, T]\}$  that maximizes the area under  $x$ ,

$$\int_0^T x(t) dt,$$

## Sec. 3.5 Notes, Sources, and Exercises

subject to the constraints

$$x(0) = a, \quad x(T) = b, \quad \int_0^T \sqrt{1 + (\dot{x}(t))^2} dt = L,$$

where  $a$ ,  $b$ , and  $L$  are given positive scalars. The last constraint is known as an isoperimetric constraint; it requires that the length of the curve be  $L$ . Hint: Introduce the system  $\dot{x}_1 = u$ ,  $\dot{x}_2 = \sqrt{1 + u^2}$ , and view the problem as a fixed terminal state problem. Show that the sine of the optimal  $u^*(t)$  depends linearly on  $t$ . Under some assumptions on  $a$ ,  $b$ , and  $L$ , the optimal curve is a circular arc.

3.6 (L'Hôpital's Problem) [www](#)

Let  $a$ ,  $b$ , and  $T$  be positive scalars, and let  $A = (0, a)$  and  $B = (T, b)$  be two points in a medium within which the velocity of propagation of light is proportional to the vertical coordinate. Thus the time it takes for light to propagate from  $A$  to  $B$  along a curve  $\{x(t) \mid t \in [0, T]\}$  is

$$\int_0^T \frac{\sqrt{1 + (\dot{x}(t))^2}}{Cx(t)} dt,$$

where  $C$  is a given positive constant. Find the curve of minimum travel time of light from  $A$  to  $B$ , and show that it is an arc of a circle of the form

$$x(t)^2 + (t - d)^2 = D,$$

where  $d$  and  $D$  are some constants.

## 3.7

A boat moves with constant unit velocity in a stream moving at constant velocity  $s$ . The problem is to find the steering angle  $u(t)$ ,  $0 \leq t \leq T$ , which minimizes the time  $T$  required for the boat to move between the point  $(a, 0)$  to a given point  $(a, b)$ . The equations of motion are

$$\dot{x}_1(t) = s + \cos u(t), \quad \dot{x}_2(t) = \sin u(t),$$

where  $x_1(t)$  and  $x_2(t)$  are the positions of the boat parallel and perpendicular to the stream velocity, respectively. Show that the optimal solution is to steer at a constant angle.

## 3.8

A unit mass object moves on a straight line from a given initial position  $x_1(0)$  and velocity  $x_2(0)$ . Find the force  $\{u(t) \mid t \in [0, 1]\}$  that brings the object at time 1 to rest [ $x_2(1) = 0$ ] at position  $x_1(1) = 0$ , and minimizes

$$\int_0^1 (u(t))^2 dt.$$

## 3.9 (www)

Use the Minimum Principle to solve the linear-quadratic problem of Example 3.2.2. *Hint:* Follow the lines of Example 3.3.3.

## 3.10 (On the Need for Convexity Assumptions)

Solve the continuous-time problem involving the system  $\dot{x}(t) = u(t)$ , the terminal cost  $(x(T))^2$ , and the control constraint  $u(t) = -1$  or  $1$  for all  $t$ , and show that the solution satisfies the Minimum Principle. Show that, depending on the initial state  $x_0$ , this may not be true for the discrete-time version involving the system  $x_{k+1} = x_k + u_k$ , the terminal cost  $x_N^2$ , and the control constraint  $u_k = -1$  or  $1$  for all  $k$ .

## 3.11

Use the discrete-time Minimum Principle to solve Exercise 1.14 of Chapter 1, assuming that each  $w_k$  is fixed at a known deterministic value.

## 3.12

Use the discrete-time Minimum Principle to solve Exercise 1.15 of Chapter 1, assuming that  $\gamma_k$  and  $\delta_k$  are fixed at known deterministic values.

## 3.13 (Lagrange Multipliers and the Minimum Principle)

Consider the discrete-time optimal control problem of Section 3.3.3, where there are no control constraints ( $U = \mathbb{R}^m$ ). Introduce a Lagrange multiplier vector  $p_{k+1}$  for each of the constraints

$$f_k(x_k, u_k) - x_{k+1} = 0, \quad k = 0, \dots, N-1,$$

and form the Lagrangian function

$$g_N(x_N) + \sum_{k=0}^{N-1} \left( g_k(x_k, u_k) + p'_{k+1} (f_k(x_k, u_k) - x_{k+1}) \right)$$

(cf. Appendix B). View both the state and the control vectors as the optimization variables of the problem, and show that by differentiation of the Lagrangian function with respect to  $x_k$  and  $u_k$ , we obtain the discrete-time Minimum Principle.

# Problems with Perfect State Information

---

**Contents**


---

4.1. Linear Systems and Quadratic Cost . . . . .	p. 148
4.2. Inventory Control . . . . .	p. 162
4.3. Dynamic Portfolio Analysis . . . . .	p. 170
4.4. Optimal Stopping Problems . . . . .	p. 176
4.5. Scheduling and the Interchange Argument . . . . .	p. 186
4.6. Set-Membership Description of Uncertainty . . . . .	p. 190
4.6.1. Set-Membership Estimation . . . . .	p. 191
4.6.2. Control with Unknown-but-Bounded Disturbances . . . . .	p. 197
4.7. Notes, Sources, and Exercises . . . . .	p. 201

---

In this chapter we consider a number of applications of discrete-time stochastic optimal control with perfect state information. These applications are special cases of the basic problem of Section 1.2 and can be addressed via the DP algorithm. In all these applications the stochastic nature of the disturbances is significant. For this reason, in contrast with the deterministic problems of the preceding two chapters, the use of closed-loop control is essential to achieve optimal performance.

#### 4.1 LINEAR SYSTEMS AND QUADRATIC COST

In this section we consider the special case of a linear system

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

and the quadratic cost

$$\mathbb{E}_{w_k} \left\{ x_N' Q_N x_N + \sum_{k=0}^{N-1} (x_k' Q_k x_k + u_k' R_k u_k) \right\}.$$

In these expressions,  $x_k$  and  $u_k$  are vectors of dimension  $n$  and  $m$ , respectively, and the matrices  $A_k$ ,  $B_k$ ,  $Q_k$ ,  $R_k$  are given and have appropriate dimension. We assume that the matrices  $Q_k$  are positive semidefinite symmetric, and the matrices  $R_k$  are positive definite symmetric. The controls  $u_k$  are unconstrained. The disturbances  $w_k$  are independent random vectors with given probability distributions that do not depend on  $x_k$  and  $u_k$ . Furthermore, each  $w_k$  has zero mean and finite second moment.

The problem described above is a popular formulation of a regulation problem whereby we want to keep the state of the system close to the origin. Such problems are common in the theory of automatic control of a motion or a process. The quadratic cost function is often reasonable because it induces a high penalty for large deviations of the state from the origin but a relatively small penalty for small deviations. Also, the quadratic cost is frequently used, even when it is not entirely justified, because it leads to a nice analytical solution. A number of variations and generalizations have similar solutions. For example, the disturbances  $w_k$  could have nonzero means and the quadratic cost could have the form

$$\mathbb{E} \left\{ (x_N - \bar{x}_N)' Q_N (x_N - \bar{x}_N) + \sum_{k=0}^{N-1} ((x_k - \bar{x}_k)' Q_k (x_k - \bar{x}_k) + u_k' R_k u_k) \right\},$$

which expresses a desire to keep the state of the system close to a given trajectory  $(\bar{x}_0, \bar{x}_1, \dots, \bar{x}_N)$  rather than close to the origin. Another generalized version of the problem arises when  $A_k$ ,  $B_k$  are independent random

matrices, rather than being known. This case is considered at the end of this section.

Applying now the DP algorithm, we have

$$J_N(x_N) = x_N' Q_N x_N,$$

$$J_k(x_k) = \min_{u_k} E \{ x_k' Q_k x_k + u_k' R_k u_k + J_{k+1}(A_k x_k + B_k u_k + w_k) \}, \quad (4.1)$$

It turns out that the cost-to-go functions  $J_k$  are quadratic and as a result the optimal control law is a linear function of the state. These facts can be verified by straightforward induction. We write Eq. (4.1) for  $k = N-1$ ,

$$\begin{aligned} J_{N-1}(x_{N-1}) &= \min_{u_{N-1}} E \{ x_{N-1}' Q_{N-1} x_{N-1} + u_{N-1}' R_{N-1} u_{N-1} \\ &\quad + (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1})' Q_N \\ &\quad \cdot (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1}) \}, \end{aligned}$$

and we expand the last quadratic form in the right-hand side. We then use the fact  $E\{w_{N-1}\} = 0$  to eliminate the term  $E\{w_{N-1}' Q_N (A_{N-1} x_{N-1} + B_{N-1} u_{N-1})\}$ , and we obtain

$$\begin{aligned} J_{N-1}(x_{N-1}) &= x_{N-1}' Q_{N-1} x_{N-1} + \min_{u_{N-1}} [u_{N-1}' R_{N-1} u_{N-1} \\ &\quad + u_{N-1}' B_{N-1}' Q_N B_{N-1} u_{N-1} + 2x_{N-1}' A_{N-1}' Q_N B_{N-1} u_{N-1}] \\ &\quad + x_{N-1}' A_{N-1}' Q_N A_{N-1} x_{N-1} + E\{w_{N-1}' Q_N w_{N-1}\}. \end{aligned}$$

By differentiating with respect to  $u_{N-1}$  and by setting the derivative equal to zero, we obtain

$$(R_{N-1} + B_{N-1}' Q_N B_{N-1}) u_{N-1} = -B_{N-1}' Q_N A_{N-1} x_{N-1}.$$

The matrix multiplying  $u_{N-1}$  on the left is positive definite (and hence invertible), since  $R_{N-1}$  is positive definite and  $B_{N-1}' Q_N B_{N-1}$  is positive semidefinite. As a result, the minimizing control vector is given by

$$u_{N-1}^* = -(R_{N-1} + B_{N-1}' Q_N B_{N-1})^{-1} B_{N-1}' Q_N A_{N-1} x_{N-1}.$$

By substitution into the expression for  $J_{N-1}$ , we have

$$J_{N-1}(x_{N-1}) = x_{N-1}' K_{N-1} x_{N-1} + E\{w_{N-1}' Q_N w_{N-1}\},$$

where by straightforward calculation, the matrix  $K_{N-1}$  is verified to be

$$\begin{aligned} K_{N-1} &= A_{N-1}' (Q_N - Q_N B_{N-1} (B_{N-1}' Q_N B_{N-1} + R_{N-1})^{-1} B_{N-1}' Q_N) A_{N-1} \\ &\quad + Q_{N-1}. \end{aligned}$$

The matrix  $K_{N-1}$  is clearly symmetric. It is also positive semidefinite. To see this, note that from the preceding calculation we have for  $x \in \mathbb{R}^n$

$$\begin{aligned} x' K_{N-1} x &= \min_u [x' Q_{N-1} x + u' R_{N-1} u \\ &\quad + (A_{N-1} x + B_{N-1} u)' Q_N (A_{N-1} x + B_{N-1} u)]. \end{aligned}$$

Since  $Q_{N-1}$ ,  $R_{N-1}$ , and  $Q_N$  are positive semidefinite, the expression within brackets is nonnegative. Minimization over  $u$  preserves nonnegativity, so it follows that  $x' K_{N-1} x \geq 0$  for all  $x \in \mathbb{R}^n$ . Hence  $K_{N-1}$  is positive semidefinite.

Since  $J_{N-1}$  is a positive semidefinite quadratic function (plus an inconsequential constant term), we may proceed similarly and obtain from the DP equation (4.1) the optimal control law for stage  $N - 2$ . As earlier, we show that  $J_{N-2}$  is a positive semidefinite quadratic function, and by proceeding sequentially, we obtain the optimal control law for every  $k$ . It has the form

$$\mu_k^*(x_k) = L_k x_k, \quad (4.2)$$

where the gain matrices  $L_k$  are given by the equation

$$L_k = -(B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1} A_k,$$

and where the symmetric positive semidefinite matrices  $K_k$  are given recursively by the algorithm

$$K_N = Q_N, \quad (4.3)$$

$$K_k = A_k' (K_{k+1} - K_{k+1} B_k (B_k' K_{k+1} B_k + R_k)^{-1} B_k' K_{k+1}) A_k + Q_k. \quad (4.4)$$

Just like DP, this algorithm starts at the terminal time  $N$  and proceeds backwards. The optimal cost is given by

$$J_0(x_0) = x_0' K_0 x_0 + \sum_{k=0}^{N-1} E\{w_k' K_{k+1} w_k\}.$$

The control law (4.2) is simple and attractive for implementation in engineering applications: the current state  $x_k$  is being fed back as input through the linear feedback gain matrix  $L_k$  as shown in Fig. 4.1.1. This accounts in part for the popularity of the linear-quadratic formulation. As we will see in Chapter 5, the linearity of the control law is still maintained even for problems where the state  $x_k$  is not completely observable (imperfect state information).

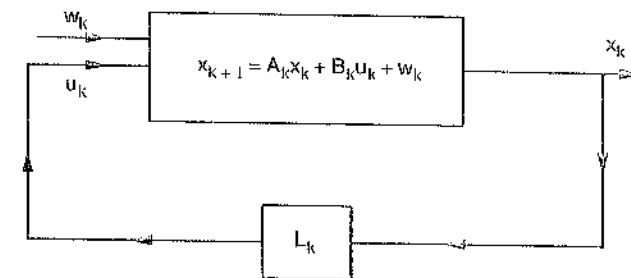


Figure 4.1.1 Linear feedback structure of the optimal controller for the linear-quadratic problem.

### The Riccati Equation and Its Asymptotic Behavior

Equation (4.4) is called the *discrete-time Riccati equation*. It plays an important role in control theory. Its properties have been studied extensively and exhaustively. One interesting property of the Riccati equation is that if the matrices  $A_k$ ,  $B_k$ ,  $Q_k$ ,  $R_k$  are constant and equal to  $A$ ,  $B$ ,  $Q$ ,  $R$ , respectively, then the solution  $K_k$  converges as  $k \rightarrow -\infty$  (under mild assumptions) to a steady-state solution  $K$  satisfying the *algebraic Riccati equation*

$$K = A' (K - KB(B'KB + R)^{-1}B'K)A + Q. \quad (4.5)$$

This property, to be proved shortly, indicates that for the system

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, 1, \dots, N-1,$$

and a large number of stages  $N$ , one can reasonably approximate the control law (4.2) by the control law  $\{\mu^*, \mu^*, \dots, \mu^*\}$ , where

$$\mu^*(x) = Lx, \quad (4.6)$$

$$L = -(B'KB + R)^{-1}B'KA,$$

and  $K$  solves the algebraic Riccati equation (4.5). This control law is *stationary*; that is, it does not change over time.

We now turn to proving convergence of the sequence of matrices  $\{K_k\}$  generated by the Riccati equation (4.4). We first introduce the notions of controllability and observability, which are very important in control theory.

**Definition 4.1.1:** A pair  $(A, B)$ , where  $A$  is an  $n \times n$  matrix and  $B$  is an  $n \times m$  matrix, is said to be *controllable* if the  $n \times nm$  matrix

$$[B, AB, A^2B, \dots, A^{n-1}B]$$

has full rank (i.e., has linearly independent rows). A pair  $(A, C)$ , where  $A$  is an  $n \times n$  matrix and  $C$  an  $m \times n$  matrix, is said to be *observable* if the pair  $(A', C')$  is controllable, where  $A'$  and  $C'$  denote the transposes of  $A$  and  $C$ , respectively.

One may show that if the pair  $(A, B)$  is controllable, then for any initial state  $x_0$ , there exists a sequence of control vectors  $u_0, u_1, \dots, u_{n-1}$  that force the state  $x_n$  of the system

$$x_{k+1} = Ax_k + Bu_k$$

to be equal to zero at time  $n$ . Indeed, by successively applying the above equation for  $k = n-1, n-2, \dots, 0$ , we obtain

$$x_n = A^n x_0 + Bu_{n-1} + ABu_{n-2} + \dots + A^{n-1}Bu_0$$

or equivalently

$$x_n - A^n x_0 = (B, AB, \dots, A^{n-1}B) \begin{pmatrix} u_{n-1} \\ u_{n-2} \\ \vdots \\ u_0 \end{pmatrix}. \quad (4.7)$$

If  $(A, B)$  is controllable, the matrix  $(B, AB, \dots, A^{n-1}B)$  has full rank and as a result the right-hand side of Eq. (4.7) can be made equal to any vector in  $\mathbb{R}^n$  by appropriate selection of  $(u_0, u_1, \dots, u_{n-1})$ . In particular, one can choose  $(u_0, u_1, \dots, u_{n-1})$  so that the right-hand side of Eq. (4.7) is equal to  $-A^n x_0$ , which implies  $x_n = 0$ . This property explains the name "controllable pair" and in fact is often used to define controllability.

The notion of observability has an analogous interpretation in the context of estimation problems; that is, given measurements  $z_0, z_1, \dots, z_{n-1}$  of the form  $z_k = Cx_k$ , it is possible to infer the initial state  $x_0$  of the system  $x_{k+1} = Ax_k$ , in view of the relation

$$\begin{pmatrix} z_{n-1} \\ \vdots \\ z_1 \\ z_0 \end{pmatrix} = \begin{pmatrix} CA^{n-1} \\ \vdots \\ CA \\ C \end{pmatrix} x_0.$$

Alternatively, it can be seen that observability is equivalent to the property that, in the absence of control, if  $Cx_k \rightarrow 0$  then  $x_k \rightarrow 0$ .

The notion of stability is of paramount importance in control theory. In the context of our problem it is important that the stationary control law (4.6) results in a stable closed-loop system; that is, in the absence of input disturbance, the state of the system

$$x_{k+1} = (A + BL)x_k, \quad k = 0, 1, \dots,$$

tends to zero as  $k \rightarrow \infty$ . Since  $x_k = (A + BL)^k x_0$ , it follows that the closed-loop system is stable if and only if  $(A + BL)^k \rightarrow 0$ , or equivalently (see Appendix A), if and only if the eigenvalues of the matrix  $(A + BL)$  are strictly within the unit circle.

The following proposition shows that for a stationary controllable system and constant matrices  $Q$  and  $R$ , the solution of the Riccati equation (4.4) converges to a positive definite symmetric matrix  $K$  for an arbitrary positive semidefinite symmetric initial matrix. In addition, the proposition shows that the corresponding closed-loop system is stable. The proposition also requires an observability assumption, namely, that  $Q$  can be written as  $C'C$ , where the pair  $(A, C)$  is observable. Note that if  $r$  is the rank of  $Q$ , there exists an  $r \times n$  matrix  $C$  of rank  $r$  such that  $Q = C'C$  (see Appendix A). The implication of the observability assumption is that in the absence of control, if the state cost per stage  $x_k' Q x_k$  tends to zero or equivalently  $Cx_k \rightarrow 0$ , then also  $x_k \rightarrow 0$ .

To simplify notation, we reverse the time indexing of the Riccati equation. Thus,  $P_k$  in the following proposition corresponds to  $K_{N-k}$  in Eq. (4.4). A graphical proof of the proposition for the case of a scalar system is given in Fig. 4.1.2.

**Proposition 4.4.1:** Let  $A$  be an  $n \times n$  matrix,  $B$  be an  $n \times m$  matrix,  $Q$  be an  $n \times n$  positive semidefinite symmetric matrix, and  $R$  be an  $m \times m$  positive definite symmetric matrix. Consider the discrete-time Riccati equation

$$P_{k+1} = A'(P_k - P_k B(B'P_k B + R)^{-1}B'P_k)A + Q, \quad k = 0, 1, \dots, \quad (4.8)$$

where the initial matrix  $P_0$  is an arbitrary positive semidefinite symmetric matrix. Assume that the pair  $(A, B)$  is controllable. Assume also that  $Q$  may be written as  $C'C$ , where the pair  $(A, C)$  is observable. Then:

- (a) There exists a positive definite symmetric matrix  $P$  such that for every positive semidefinite symmetric initial matrix  $P_0$  we have

$$\lim_{k \rightarrow \infty} P_k = P.$$

Furthermore,  $P$  is the unique solution of the algebraic matrix equation

$$P = A' \left( P - PB(B'PB + R)^{-1}B'P \right) A + Q \quad (4.9)$$

within the class of positive semidefinite symmetric matrices.

- (b) The corresponding closed-loop system is stable; that is, the eigenvalues of the matrix

$$D = A + BL, \quad (4.10)$$

where

$$L = -(B'PB + R)^{-1}B'PA, \quad (4.11)$$

are strictly within the unit circle.

**Proof:** The proof proceeds in several steps. First we show convergence of the sequence generated by Eq. (4.8) when the initial matrix  $P_0$  is equal to zero. Next we show that the corresponding matrix  $D$  of Eq. (4.10) satisfies  $D^k \rightarrow 0$ . Then we show the convergence of the sequence generated by Eq. (4.8) when  $P_0$  is any positive semidefinite symmetric matrix, and finally we show uniqueness of the solution of Eq. (4.9).

*Initial Matrix  $P_0 = 0$ .* Consider the optimal control problem of finding  $u_0, u_1, \dots, u_{k-1}$  that minimize

$$\sum_{i=0}^{k-1} (x_i' Q x_i + u_i' R u_i)$$

subject to

$$x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, \dots, k-1,$$

where  $x_0$  is given. The optimal value of this problem, according to the theory of this section, is  $x_0' P_k(0) x_0$ ,

where  $P_k(0)$  is given by the Riccati equation (4.8) with  $P_0 = 0$ . For any control sequence  $(u_0, u_1, \dots, u_k)$  we have

$$\sum_{i=0}^{k-1} (x_i' Q x_i + u_i' R u_i) \leq \sum_{i=0}^k (x_i' Q x_i + u_i' R u_i)$$

and hence

$$\begin{aligned} x_0' P_k(0) x_0 &= \min_{u_i} \sum_{i=0, \dots, k-1}^{k-1} (x_i' Q x_i + u_i' R u_i) \\ &\leq \min_{u_i} \sum_{i=0, \dots, k}^{k} (x_i' Q x_i + u_i' R u_i) \\ &= x_0' P_{k+1}(0) x_0, \end{aligned}$$

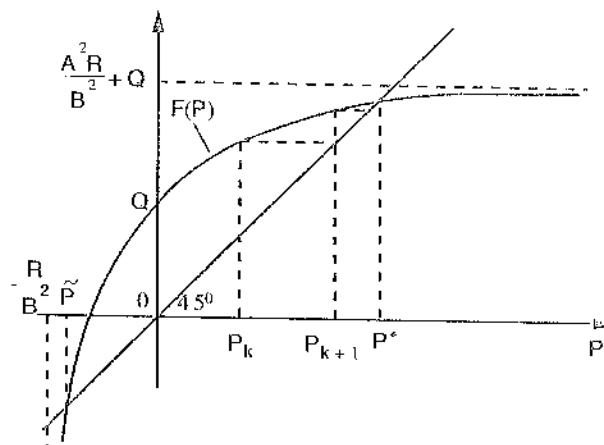


Figure 4.1.2 Graphical proof of Prop. 4.4.1 for the case of a scalar stationary system (one-dimensional state and control), assuming that  $A \neq 0$ ,  $B \neq 0$ ,  $Q > 0$ , and  $R > 0$ . The Riccati equation (4.8) is given by

$$P_{k+1} = A^2 \left( P_k - \frac{B^2 P_k^2}{B^2 P_k + R} \right) + Q,$$

which can be equivalently written as

$$P_{k+1} = F(P_k),$$

where the function  $F$  is given by

$$F(P) = \frac{A^2 R P}{B^2 P + R} + Q.$$

Because  $F$  is concave and monotonically increasing in the interval  $(-R/B^2, \infty)$ , as shown in the figure, the equation  $P = F(P)$  has one positive solution  $P^*$  and one negative solution  $\tilde{P}$ . The Riccati iteration  $P_{k+1} = F(P_k)$  converges to  $P^*$  starting anywhere in the interval  $(\tilde{P}, \infty)$  as shown in the figure.

where both minimizations are subject to the system equation constraint  $x_{i+1} = Ax_i + Bu_i$ . Furthermore, for a fixed  $x_0$  and for every  $k$ ,  $x_0' P_k(0) x_0$  is bounded from above by the cost corresponding to a control sequence that forces  $x_0$  to the origin in  $n$  steps and applies zero control after that. Such a sequence exists by the controllability assumption. Thus the sequence  $\{x_0' P_k(0) x_0\}$  is nondecreasing with respect to  $k$  and bounded from above, and therefore converges to some real number for every  $x_0 \in \mathbb{R}^n$ . It follows that the sequence  $\{P_k(0)\}$  converges to some matrix  $P$  in the sense that each of the sequences of the elements of  $P_k(0)$  converges to the correspond-

ing elements of  $P$ . To see this, take  $x_0 = (1, 0, \dots, 0)$ . Then  $x_0' P_k(0) x_0$  is equal to the first diagonal element of  $P_k(0)$ , so it follows that the sequence of first diagonal elements of  $P_k(0)$  converges; the limit of this sequence is the first diagonal element of  $P$ . Similarly, by taking  $x_0 = (0, \dots, 0, 1, 0, \dots, 0)$  with the 1 in the  $i$ th coordinate, for  $i = 2, \dots, n$ , it follows that all the diagonal elements of  $P_k(0)$  converge to the corresponding diagonal elements of  $P$ . Next take  $x_0 = (1, 1, 0, \dots, 0)$  to show that the second elements of the first row converge. Continuing similarly, we obtain

$$\lim_{k \rightarrow \infty} P_k(0) = P,$$

where  $P_k(0)$  are generated by Eq. (4.8) with  $P_0 = 0$ . Furthermore, since  $P_k(0)$  is positive semidefinite and symmetric, so is the limit matrix  $P$ . Now by taking the limit in Eq. (4.8) it follows that  $P$  satisfies

$$P = A'(P - PB(B'PB + R)^{-1}B'P)A + Q.$$

In addition, by direct calculation we can verify the following useful equality

$$P = D'PD + Q + L'RL, \quad (4.12)$$

where  $D$  and  $L$  are given by Eqs. (4.10) and (4.11). An alternative way to derive this equality is to observe that from the DP algorithm corresponding to a finite horizon  $N$  we have for all states  $x_{N-k}$

$$\begin{aligned} x_{N-k}' P_{k+1}(0) x_{N-k} &= x_{N-k}' Q x_{N-k} + \mu_{N-k}^*(x_{N-k})' R \mu_{N-k}^*(x_{N-k}) \\ &\quad + x_{N-k+1}' P_k(0) x_{N-k+1}. \end{aligned}$$

By using the optimal controller expression  $\mu_{N-k}^*(x_{N-k}) = L_{N-k} x_{N-k}$  and the closed-loop system equation  $x_{N-k+1} = (A + BL_{N-k})x_{N-k}$ , we thus obtain

$$P_{k+1}(0) = Q + L_{N-k}' RL_{N-k} + (A + BL_{N-k})' P_k(0)(A + BL_{N-k}). \quad (4.13)$$

Equation (4.12) then follows by taking the limit as  $k \rightarrow \infty$  in Eq. (4.13).

*Stability of the Closed-Loop System.* Consider the system

$$x_{k+1} = (A + BL)x_k = Dx_k \quad (4.14)$$

for an arbitrary initial state  $x_0$ . We will show that  $x_k \rightarrow 0$  as  $k \rightarrow \infty$ . We have for all  $k$ , by using Eq. (4.12),

$$x_{k+1}' Px_{k+1} - x_k' Px_k = x_k' (D'PD - P)x_k = -x_k' (Q + L'RL)x_k.$$

Hence

$$x_{k+1}' Px_{k+1} = x_0' Px_0 - \sum_{i=0}^k x_i' (Q + L'RL)x_i. \quad (4.15)$$

The left-hand side of this equation is bounded below by zero, so it follows that

$$\lim_{k \rightarrow \infty} x_k' (Q + L'RL)x_k = 0.$$

Since  $R$  is positive definite and  $Q$  may be written as  $C'C$ , we obtain

$$\lim_{k \rightarrow \infty} Cx_k = 0, \quad \lim_{k \rightarrow \infty} Lx_k = \lim_{k \rightarrow \infty} \mu^*(x_k) = 0. \quad (4.16)$$

The preceding relations imply that as the control asymptotically becomes negligible, we have  $\lim_{k \rightarrow \infty} Cx_k = 0$ , and in view of the observability assumption, this implies that  $x_k \rightarrow 0$ . To express this argument more precisely, let us use the relation  $x_{k+1} = (A + BL)x_k$  [cf. Eq. (4.14)], to write

$$\begin{pmatrix} C(x_{k+n-1} - \sum_{i=1}^{n-1} A^{i-1} BLx_{k+n-i-1}) \\ C(x_{k+n-2} - \sum_{i=1}^{n-2} A^{i-1} BLx_{k+n-i-2}) \\ \vdots \\ C(x_{k+1} - BLx_k) \\ Cx_k \end{pmatrix} = \begin{pmatrix} CA^{n-1} \\ CA^{n-2} \\ \vdots \\ CA \\ C \end{pmatrix} x_k. \quad (4.17)$$

Since  $Lx_k \rightarrow 0$  by Eq. (4.16), the left-hand side tends to zero and hence the right-hand side tends to zero also. By the observability assumption, however, the matrix multiplying  $x_k$  on the right side of (4.17) has full rank. It follows that  $x_k \rightarrow 0$ .

*Positive Definiteness of  $P$ .* Assume the contrary, i.e., there exists some  $x_0 \neq 0$  such that  $x_0' Px_0 = 0$ . Since  $P$  is positive semidefinite, from Eq. (4.15) we obtain

$$x_k' (Q + L'RL)x_k = 0, \quad k = 0, 1, \dots$$

Since  $x_k \rightarrow 0$ , we obtain  $x_k' Q x_k = x_k' C'C x_k = 0$  and  $x_k' L'RL x_k = 0$ , or

$$Cx_k = 0, \quad Lx_k = 0, \quad k = 0, 1, \dots$$

Thus all the controls  $\mu^*(x_k) = Lx_k$  of the closed-loop system are zero while we have  $Cx_k = 0$  for all  $k$ . Based on the observability assumption, we will show that this implies  $x_0 = 0$ , thereby reaching a contradiction. Indeed, consider Eq. (4.17) for  $k = 0$ . By the preceding equalities, the left hand side is zero and hence

$$0 = \begin{pmatrix} CA^{n-1} \\ \vdots \\ CA \\ C \end{pmatrix} x_0.$$

Since the matrix multiplying  $x_0$  above has full rank by the observability assumption, we obtain  $x_0 = 0$ , which contradicts the hypothesis  $x_0 \neq 0$  and proves that  $P$  is positive definite.

*Arbitrary Initial Matrix  $P_0$ .* Next we show that the sequence of matrices  $\{P_k(P_0)\}$ , defined by Eq. (4.8) when the starting matrix is an arbitrary positive semidefinite symmetric matrix  $P_0$ , converges to  $P = \lim_{k \rightarrow \infty} P_k(0)$ . Indeed, the optimal cost of the problem of minimizing

$$x'_k P_0 x_k + \sum_{i=0}^{k-1} (x'_i Q x_i + u'_i R u_i) \quad (4.18)$$

subject to the system equation  $x_{i+1} = Ax_i + Bu_i$  is equal to  $x'_0 P_k(P_0) x_0$ . Hence we have for every  $x_0 \in \mathbb{R}^n$

$$x'_0 P_k(0) x_0 \leq x'_0 P_k(P_0) x_0.$$

Consider now the cost (4.18) corresponding to the controller  $\mu(x_k) = u_k = Lx_k$ , where  $L$  is defined by Eq. (4.11). This cost is

$$x'_0 \left( D^{k'} P_0 D^k + \sum_{i=0}^{k-1} D^{i'} (Q + L' RL) D^i \right) x_0$$

and is greater or equal to  $x'_0 P_k(P_0) x_0$ , which is the optimal value of the cost (4.18). Hence we have for all  $k$  and  $x \in \mathbb{R}^n$

$$x' P_k(0) x \leq x' P_k(P_0) x \leq x' \left( D^{k'} P_0 D^k + \sum_{i=0}^{k-1} D^{i'} (Q + L' RL) D^i \right) x.$$

We have proved that

$$\lim_{k \rightarrow \infty} P_k(0) = P,$$

and we also have, using the fact  $\lim_{k \rightarrow \infty} D^{k'} P_0 D^k = 0$ , and the relation  $Q + L' RL = P - D' PD$  [cf. Eq. (4.12)],

$$\begin{aligned} \lim_{k \rightarrow \infty} & \left\{ D^{k'} P_0 D^k + \sum_{i=0}^{k-1} D^{i'} (Q + L' RL) D^i \right\} \\ &= \lim_{k \rightarrow \infty} \left\{ \sum_{i=0}^{k-1} D^{i'} (Q + L' RL) D^i \right\} \\ &= \lim_{k \rightarrow \infty} \left\{ \sum_{i=0}^{k-1} D^{i'} (P - D' PD) D^i \right\} \\ &= P. \end{aligned} \quad (4.19)$$

Combining the preceding three equations, we obtain

$$\lim_{k \rightarrow \infty} P_k(P_0) = P,$$

for an arbitrary positive semidefinite symmetric initial matrix  $P_0$ .

*Uniqueness of Solution.* If  $\tilde{P}$  is another positive semidefinite symmetric solution of the algebraic Riccati equation (4.9), we have  $P_k(\tilde{P}) = \tilde{P}$  for all  $k = 0, 1, \dots$ . From the convergence result just proved, we then obtain

$$\lim_{k \rightarrow \infty} P_k(\tilde{P}) = P,$$

implying that  $\tilde{P} = P$ . Q.E.D.

The assumptions of the preceding proposition can be relaxed somewhat. Suppose that, instead of controllability of the pair  $(A, B)$ , we assume that the system is *stabilizable* in the sense that there exists an  $m \times n$  feedback gain matrix  $G$  such that the closed-loop system  $x_{k+1} = (A + BG)x_k$  is stable. Then the proof of convergence of  $P_k(0)$  to some positive semidefinite  $P$  given previously carries through. [We use the stationary control law  $\mu(x) = Gx$  for which the closed-loop system is stable to ensure that  $x'_0 P_k(0) x_0$  is bounded.] Suppose that, instead of observability of the pair  $(A, C)$ , the system is assumed *detectable* in the sense that  $A$  is such that if  $u_k \rightarrow 0$  and  $Cx_k \rightarrow 0$  then it follows that  $x_k \rightarrow 0$ . (This essentially means that instability of the system can be detected by looking at the measurement sequence  $\{z_k\}$  with  $z_k = Cx_k$ .) Then Eq. (4.16) implies that  $x_k \rightarrow 0$  and that the system  $x_{k+1} = (A + BL)x_k$  is stable. The other parts of the proof of the proposition follow similarly, with the exception of positive definiteness of  $P$ , which cannot be guaranteed anymore. (As an example, take  $A = 0$ ,  $B = 0$ ,  $C = 0$ ,  $R > 0$ . Then both the stabilizability and the detectability assumptions are satisfied, but  $P = 0$ .)

To summarize, if the controllability and observability assumptions of the proposition are replaced by the preceding stabilizability and detectability assumptions, the conclusions of the proposition hold with the exception of positive definiteness of the limit matrix  $P$ , which can now only be guaranteed to be positive semidefinite.

### Random System Matrices

We consider now the case where  $\{A_0, B_0\}, \dots, \{A_{N-1}, B_{N-1}\}$  are not known but rather are independent random matrices that are also independent of  $w_0, w_1, \dots, w_{N-1}$ . Their probability distributions are given, and they are assumed to have finite second moments. This problem falls again within the framework of the basic problem by considering as disturbance at each time  $k$  the triplet  $(A_k, B_k, w_k)$ . The DP algorithm is written as

$$J_N(x_N) = x'_N Q_N x_N,$$

$$J_k(x_k) = \min_{u_k, w_k, A_k, B_k} E_{u_k, w_k, A_k, B_k} \{ x'_k Q_k x_k + u'_k R_k u_k + J_{k+1}(A_k x_k + B_k u_k + w_k) \}.$$

Calculations very similar to those for the case where  $A_k, B_k$  are not random show that the optimal control law has the form

$$\mu_k^*(x_k) = L_k x_k,$$

where the gain matrices  $L_k$  are given by

$$L_k = -(R_k + E\{B'_k K_{k+1} B_k\})^{-1} E\{B'_k K_{k+1} A_k\},$$

and where the matrices  $K_k$  are given by the recursive equation

$$K_N = Q_N,$$

$$\begin{aligned} K_k &= E\{A'_k K_{k+1} A_k\} \\ &\quad - E\{A'_k K_{k+1} B_k\}(R_k + E\{B'_k K_{k+1} B_k\})^{-1} E\{B'_k K_{k+1} A_k\} + Q_k. \end{aligned} \quad (4.20)$$

In the case of a stationary system and constant matrices  $Q_k$  and  $R_k$  it is not necessarily true that the above equation converges to a steady-state solution. This is demonstrated in Fig. 4.1.3 for a scalar system, where it is shown that if the expression

$$T = E\{A^2\}E\{B^2\} - (E\{A\})^2(E\{B\})^2$$

exceeds a certain threshold, the matrices  $K_k$  diverge to  $\infty$  starting from any nonnegative initial condition. A possible interpretation is that if there is a lot of uncertainty about the system, as quantified by  $T$ , optimization over a long horizon is meaningless. This phenomenon has been called the *uncertainty threshold principle*; see Athans, Ku, and Gershwin [AGK77], and Ku and Athans [KuA77].

### On Certainty Equivalence

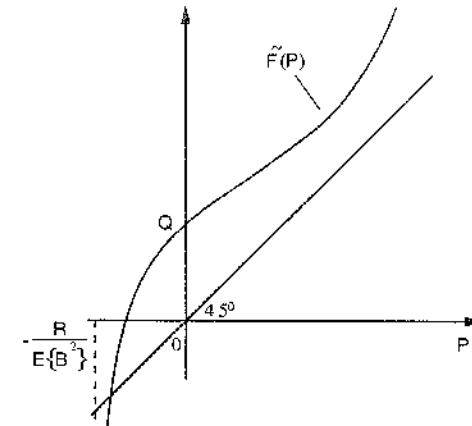
We close this section by making an observation about the simplifications that arise when the cost is quadratic. Consider the minimization over  $u$  of

$$E_w\{(ax + bu + w)^2\},$$

where  $a$  and  $b$  are given scalars,  $x$  is known, and  $w$  is a random variable. The optimum is attained for

$$u^* = -\left(\frac{a}{b}\right)x - \left(\frac{1}{b}\right)E\{w\}.$$

Thus  $u^*$  depends on the probability distribution of  $w$  only through the mean  $E\{w\}$ . In particular, the result of the optimization is the same as



**Figure 4.1.3** Graphical illustration of the asymptotic behavior of the generalized Riccati equation (4.20) in the case of a scalar stationary system (one-dimensional state and control). Using  $P_k$  in place of  $K_{N-k}$ , this equation is written as

$$P_{k+1} = \tilde{F}(P_k),$$

where the function  $\tilde{F}$  is given by

$$\tilde{F}(P) = \frac{E\{A^2\}RP}{E\{B^2\}P + R} + Q + \frac{TP^2}{E\{B^2\}P + R},$$

$$T = E\{A^2\}E\{B^2\} - (E\{A\})^2(E\{B\})^2.$$

If  $T = 0$ , as in the case where  $A$  and  $B$  are not random, the Riccati equation becomes identical with the one of Fig. 4.1.2 and converges to a steady-state. Convergence also occurs when  $T$  has a small positive value. However, as illustrated in the figure, for  $T$  large enough, the graph of the function  $\tilde{F}$  and the 45-degree line that passes through the origin do not intersect at a positive value of  $P$ , and the Riccati equation diverges to infinity.

for the corresponding deterministic problem where  $w$  is replaced by  $E\{w\}$ . This property is called the *certainty equivalence principle* and appears in various forms in many (but not all) stochastic control problems involving linear systems and quadratic cost. For the first problem of this section, where  $A_k, B_k$  are known, certainty equivalence holds because the optimal control law (4.2) is the same as the one that would be obtained from the corresponding deterministic problem where  $w_k$  is not random but rather is known and is equal to zero (its expected value). However, for the problem where  $A_k, B_k$  are random, the certainty equivalence principle does not hold, since if one replaces  $A_k, B_k$  with their expected values in Eq. (4.20) the resulting control law need not be optimal.

## 4.2 INVENTORY CONTROL

We consider now the inventory control problem discussed in Sections 1.1 and 1.2. We assume that excess demand at each period is backlogged and is filled when additional inventory becomes available. This is represented by negative inventory in the system equation

$$x_{k+1} = x_k + u_k - w_k, \quad k = 0, 1, \dots, N-1.$$

We assume that the demands  $w_k$  take values within some bounded interval and are independent. We will analyze the problem for the case of a holding/shortage cost of the form

$$r(x) = p \max(0, -x) + h \max(0, x),$$

where  $p$  and  $h$  are given nonnegative scalars. Thus the total expected cost to be minimized is

$$E \left\{ \sum_{k=0}^{N-1} (cu_k + r(x_k + u_k - w_k)) \right\}.$$

We assume that the purchase cost per unit stock  $c$  is positive and that  $p > c$ . The last assumption is necessary for the problem to be well posed; if  $c$ , the purchase cost per unit, were greater than  $p$ , the depletion cost per unit, it would never be optimal to buy new stock at the last period and possibly in earlier periods. Much of the subsequent analysis generalizes to the case where  $r$  is a convex function that grows to infinity with asymptotic slopes  $p$  and  $h$  as its argument tends to  $-\infty$  and  $\infty$ , respectively.

By applying the DP algorithm, we have

$$J_N(x_N) = 0,$$

$$J_k(x_k) = \min_{u_k \geq 0} [cu_k + H(x_k + u_k) + E\{J_{k+1}(x_k + u_k - w_k)\}], \quad (4.21)$$

where the function  $H$  is defined by

$$H(y) = E\{r(y - w_k)\} = pE\{\max(0, w_k - y)\} + hE\{\max(0, y - w_k)\}.$$

Actually,  $H$  depends on  $k$  whenever the probability distribution of  $w_k$  depends on  $k$ . To simplify notation, we do not show this dependence and assume that all demands are identically distributed, but the following analysis carries through even when the demand distribution is time-varying. The function  $H$  can be seen to be convex, since  $r(y - w_k)$  is convex in  $y$  for each fixed  $w_k$ , and taking expectation over  $w_k$  preserves convexity.

By introducing the variable  $y_k = x_k + u_k$ , we can write the DP Eq. (4.21) as

$$J_k(x_k) = \min_{y_k \geq x_k} G_k(y_k) - cx_k, \quad (4.22)$$

where

$$G_k(y) = cy + H(y) + E\{J_{k+1}(y - w)\}. \quad (4.23)$$

We will prove shortly that the function  $G_k$  is convex, but for the moment let us assume this convexity. Suppose that  $G_k$  has an unconstrained minimum with respect to  $y$ , denoted by  $S_k$ :

$$S_k = \arg \min_{y \in \mathbb{R}} G_k(y).$$

Then, in view of the constraint  $y_k \geq x_k$ , it is seen that a minimizing  $y_k$  in Eq. (4.22) equals  $S_k$  if  $x_k < S_k$ , and equals  $x_k$  otherwise [since by convexity,  $G_k(y)$  cannot decrease as  $y$  increases beyond  $S_k$ ]. Using the reverse transformation  $u_k = y_k - x_k$ , we see that the minimum in the DP equation (4.21) is attained at  $u_k = S_k - x_k$  if  $x_k < S_k$ , and at  $u_k = 0$  otherwise. An optimal policy is determined by the sequence of scalars  $\{S_0, S_1, \dots, S_{N-1}\}$  and has the form

$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < S_k, \\ 0 & \text{if } x_k \geq S_k. \end{cases} \quad (4.24)$$

Thus, the optimality of the policy (4.24) will be proved if we can show that the cost-to-go functions  $J_k$  [and hence also the functions  $G_k$  of Eq. (4.23)] are convex, and furthermore  $\lim_{|y| \rightarrow \infty} G_k(y) = \infty$ , so that the minimizing scalars  $S_k$  exist. We proceed to show these properties inductively.

We have that  $J_N$  is the zero function, so it is convex. Since  $c < p$  and the derivative of  $H(y)$  tends to  $-p$  as  $y \rightarrow -\infty$ , we see that  $G_{N-1}(y)$  [which is  $cy + H(y)$ ] has a derivative that becomes negative as  $y \rightarrow -\infty$  and becomes positive as  $y \rightarrow \infty$  (see Fig. 4.2.1). Therefore

$$\lim_{|y| \rightarrow \infty} G_{N-1}(y) = \infty.$$

Thus, as shown above, an optimal policy at time  $N-1$  is given by

$$\mu_{N-1}^*(x_{N-1}) = \begin{cases} S_{N-1} - x_{N-1} & \text{if } x_{N-1} < S_{N-1}, \\ 0 & \text{if } x_{N-1} \geq S_{N-1}. \end{cases}$$

Furthermore, from the DP equation (4.21) we have

$$J_{N-1}(x_{N-1}) = \begin{cases} c(S_{N-1} - x_{N-1}) + H(S_{N-1}) & \text{if } x_{N-1} < S_{N-1}, \\ H(x_{N-1}) & \text{if } x_{N-1} \geq S_{N-1}, \end{cases}$$

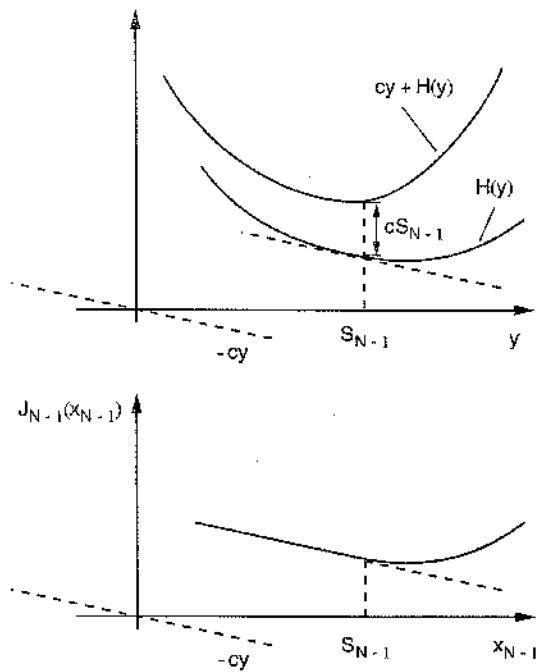


Figure 4.2.1 Structure of the cost-to-go functions when the fixed cost is zero.

which is a convex function because  $H$  is convex and  $S_{N-1}$  minimizes  $cy + H(y)$  (see Fig. 4.2.1). Thus, given the convexity of  $J_N$ , we were able to prove the convexity of  $J_{N-1}$ . Furthermore,

$$\lim_{|y| \rightarrow \infty} J_{N-1}(y) = \infty.$$

This argument can be repeated to show that for all  $k = N-2, \dots, 0$ , if  $J_{k+1}$  is convex,  $\lim_{|y| \rightarrow \infty} J_{k+1}(y) = \infty$ , and  $\lim_{|y| \rightarrow \infty} G_k(y) = \infty$ , then we have

$$J_k(x_k) = \begin{cases} c(S_k - x_k) + H(S_k) + E\{J_{k+1}(S_k - w_k)\} & \text{if } x_k < S_k, \\ H(x_k) + E\{J_{k+1}(x_k - w_k)\} & \text{if } x_k \geq S_k, \end{cases}$$

where  $S_k$  is an unconstrained minimum of  $G_k$ . Furthermore,  $J_k$  is convex,  $\lim_{|y| \rightarrow \infty} J_k(y) = \infty$ , and  $\lim_{|y| \rightarrow \infty} G_{k-1}(y) = \infty$ . Thus, the optimality proof of the policy (4.24) is completed.

#### Positive Fixed Cost and $(s, S)$ Policies

We now turn to the more complicated case where there is a positive fixed cost  $K$  associated with a positive inventory order. Thus the cost for order-

ing inventory  $u \geq 0$  is

$$C(u) = \begin{cases} K + cu & \text{if } u > 0, \\ 0 & \text{if } u = 0. \end{cases}$$

The DP algorithm takes the form

$$J_N(x_N) = 0,$$

$$J_k(x_k) = \min_{u_k \geq 0} [C(u_k) + H(x_k + u_k) + E\{J_{k+1}(x_k + u_k - w_k)\}],$$

with  $H$  defined as earlier by

$$H(y) = E\{r(y - w)\} = pE\{\max(0, w - y)\} + hE\{\max(0, y - w)\}.$$

Consider again the functions

$$G_k(y) = cy + H(y) + E\{J_{k+1}(y - w)\}.$$

Then  $J_k$  is written as

$$J_k(x_k) = \min \left[ G_k(x_k), \min_{u_k > 0} [K + G_k(x_k + u_k)] \right] - cx_k,$$

or equivalently, through the change of variable  $y_k = x_k + u_k$ ,

$$J_k(x_k) = \min \left[ G_k(x_k), \min_{y_k > x_k} [K + G_k(y_k)] \right] - cx_k.$$

If, as in the case where  $K = 0$ , we could prove that the functions  $G_k$  are convex, then it would not be difficult to verify [see part (d) of the following Lemma 4.2.1] that the policy

$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < s_k, \\ 0 & \text{if } x_k \geq s_k, \end{cases} \quad (4.25)$$

is optimal, where  $S_k$  is a value of  $y$  that minimizes  $G_k(y)$  and  $s_k$  is the smallest value of  $y$  for which  $G_k(y) = K + G_k(S_k)$ . A policy of the form (4.25) is known as a *multiperiod  $(s, S)$  policy*.

Unfortunately, when  $K > 0$  it is not necessarily true that the functions  $G_k$  are convex. This opens the possibility of  $G_k$  having the form shown in Fig. 4.2.2, where the optimal policy is to order  $(S - x)$  in interval I, zero in intervals II and IV, and  $(\tilde{S} - x)$  in interval III. However, we will show that even though the functions  $G_k$  may not be convex, they have the property

$$K + G_k(z + y) \geq G_k(y) + z \left( \frac{G_k(y) - G_k(y - b)}{b} \right), \quad \text{for all } z \geq 0, b > 0, y. \quad (4.26)$$

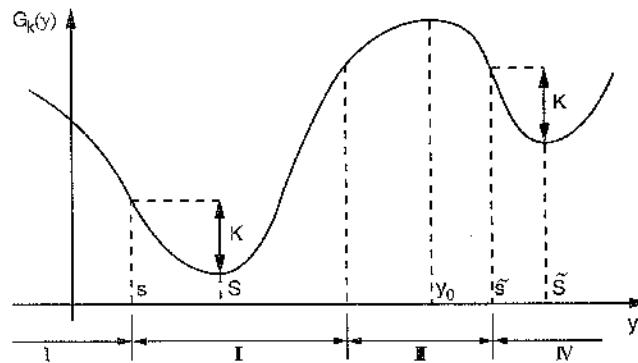
This property is called *K-convexity* and was first used by Scarf [Sca60] to show the optimality of multiperiod  $(s, S)$  policies. Now if *K*-convexity holds, the situation shown in Fig. 4.2.2 is impossible; if  $y_0$  is the local maximum in the interval III, then we must have, for sufficiently small  $b > 0$ ,

$$\frac{G_k(y_0) - G_k(y_0 - b)}{b} \geq 0,$$

and from Eq. (4.26) it follows that

$$K + G_k(\tilde{S}) \geq G_k(y_0),$$

which contradicts the construction shown in Fig. 4.2.2. More generally, it will be shown by using part (d) of the following Lemma 4.2.1 that if the *K*-convexity Eq. (4.26) holds, then an optimal policy takes the  $(s, S)$  form (4.25).



**Figure 4.2.2** Potential form of the function  $G_k$  when the fixed cost is nonzero. If  $G_k$  had the form shown in the figure, the optimal policy would be to order  $(S - x)$  in interval I, zero in intervals II and IV, and  $(\tilde{S} - x)$  in interval III. The use of *K*-convexity allows us to show that the form of  $G_k$  shown in the figure is impossible.

**Definition 4.2.1:** We say that a real-valued function  $g$  is *K*-convex, where  $K \geq 0$ , if

$$K + g(z + y) \geq g(y) + z \left( \frac{g(y) - g(y - b)}{b} \right), \quad \text{for all } z \geq 0, b > 0, y.$$

Some properties of *K*-convex functions are provided in the following lemma. Part (d) of the lemma essentially proves the optimality of the  $(s, S)$  policy (4.25) when the functions  $G_k$  are *K*-convex.

**Lemma 4.2.1:**

- (a) A real-valued convex function  $g$  is also 0-convex and hence also *K*-convex for all  $K \geq 0$ .
- (b) If  $g_1(y)$  and  $g_2(y)$  are *K*-convex and *L*-convex ( $K \geq 0, L \geq 0$ ), respectively, then  $\alpha g_1(y) + \beta g_2(y)$  is  $(\alpha K + \beta L)$ -convex for all  $\alpha > 0$  and  $\beta > 0$ .
- (c) If  $g(y)$  is *K*-convex and  $w$  is a random variable, then  $E_w\{g(y - w)\}$  is also *K*-convex, provided  $E_w\{|g(y - w)|\} < \infty$  for all  $y$ .
- (d) If  $g$  is a continuous *K*-convex function and  $g(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ , then there exist scalars  $s$  and  $S$  with  $s \leq S$  such that
  - (i)  $g(S) \leq g(y)$ , for all scalars  $y$ ;
  - (ii)  $g(S) + K = g(s) < g(y)$ , for all  $y < s$ ;
  - (iii)  $g(y)$  is a decreasing function on  $(-\infty, s)$ ;
  - (iv)  $g(y) \leq g(z) + K$  for all  $y, z$  with  $s \leq y \leq z$ .

**Proof:** Part (a) follows from elementary properties of convex functions, and parts (b) and (c) follow from the definition of a *K*-convex function. We will thus concentrate on proving part (d).

Since  $g$  is continuous and  $g(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ , there exists a minimizing point of  $g$ . Let  $S$  be such a point. Also let  $s$  be the smallest scalar  $z$  for which  $z \leq S$  and  $g(S) + K = g(z)$ . For all  $y$  with  $y < s$ , we have from the definition of *K*-convexity

$$K + g(S) \geq g(s) + \frac{S - s}{s - y} (g(s) - g(y)).$$

Since  $K + g(S) - g(s) = 0$ , we obtain  $g(s) - g(y) \leq 0$ . Since  $y < s$  and  $s$  is the smallest scalar for which  $g(S) + K = g(s)$ , we must have  $g(s) < g(y)$  and (ii) is proved. To prove (iii), note that for  $y_1 < y_2 < s$ , we have

$$K + g(S) \geq g(y_2) + \frac{S - y_2}{y_2 - y_1} (g(y_2) - g(y_1)).$$

Also from (ii),

$$g(y_2) > g(S) + K,$$

and by adding these two inequalities we have

$$0 > \frac{S - y_2}{y_2 - y_1} (g(y_2) - g(y_1)).$$

from which we obtain  $g(y_1) > g(y_2)$ , thus proving (iii). To prove (iv), we note that it holds for  $y = z$  as well as for either  $y = S$  or  $y = s$ . There remain two other possibilities:  $S < y < z$  and  $s < y < S$ . If  $S < y < z$ , then by  $K$ -convexity

$$K + g(z) \geq g(y) + \frac{z-y}{y-S}(g(y) - g(S)) \geq g(y),$$

and (iv) is proved. If  $s < y < S$ , then by  $K$ -convexity

$$g(s) = K + g(S) \geq g(y) + \frac{S-y}{y-s}(g(y) - g(s)),$$

from which

$$\left(1 + \frac{S-y}{y-s}\right)g(s) \geq \left(1 + \frac{S-y}{y-s}\right)g(y),$$

and  $g(s) \geq g(y)$ . Noting that

$$g(z) + K \geq g(S) + K = g(s),$$

it follows that  $g(z) + K \geq g(y)$ . Thus (iv) is proved for this case as well. Q.E.D.

Consider now the function  $G_{N-1}$ :

$$G_{N-1}(y) = cy + H(y).$$

Clearly,  $G_{N-1}$  is convex and hence by part (a) of Lemma 4.2.1, it is also  $K$ -convex. We have

$$J_{N-1}(x) = \min \left[ G_{N-1}(x), \min_{y>x} [K + G_{N-1}(y)] \right] - cx,$$

and it can be seen that

$$J_{N-1}(x) = \begin{cases} K + G_{N-1}(S_{N-1}) - cx & \text{for } x < s_{N-1}, \\ G_{N-1}(x) - cx & \text{for } x \geq s_{N-1}, \end{cases} \quad (4.27)$$

where  $S_{N-1}$  minimizes  $G_{N-1}(y)$  and  $s_{N-1}$  is the smallest value of  $y$  for which  $G_{N-1}(y) = K + G_{N-1}(S_{N-1})$ . Note that since  $K > 0$ , we have  $s_{N-1} \neq S_{N-1}$  and furthermore the derivative of  $G_{N-1}$  at  $s_{N-1}$  is negative. As a result the left derivative of  $J_{N-1}$  at  $s_{N-1}$  is greater than the right derivative, as shown in Fig. 4.2.3, and  $J_{N-1}$  is not convex. However, we will show that  $J_{N-1}$  is  $K$ -convex based on the fact that  $G_{N-1}$  is  $K$ -convex. To this end we must verify that for all  $z \geq 0$ ,  $b > 0$ , and  $y$ , we have

$$K + J_{N-1}(y+z) \geq J_{N-1}(y) + z \left( \frac{J_{N-1}(y) - J_{N-1}(y-b)}{b} \right). \quad (4.28)$$

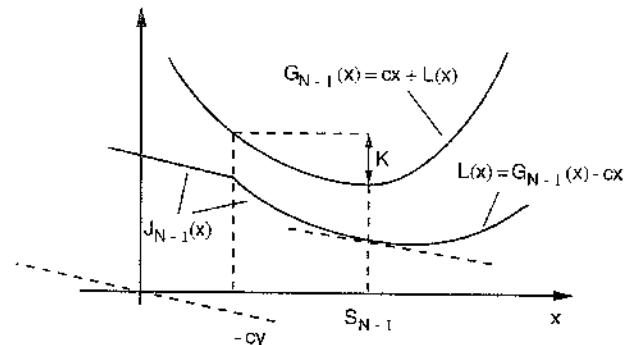


Figure 4.2.3 Structure of the cost-to-go function when fixed cost is nonzero.

We distinguish three cases:

*Case 1:*  $y \geq s_{N-1}$ . If  $y - b \geq s_{N-1}$ , then in this region of values of  $z$ ,  $b$ , and  $y$ , the function  $J_{N-1}$ , by Eq. (4.27), is the sum of a  $K$ -convex function and a linear function. Hence by part (b) of Lemma 4.2.1, it is  $K$ -convex and Eq. (4.28) holds. If  $y - b < s_{N-1}$ , then in view of Eq. (4.27) we can write Eq. (4.28) as

$$K + G_{N-1}(y+z) - c(y+z) \geq G_{N-1}(y) - cy + z \left( \frac{G_{N-1}(y) - cy - G_{N-1}(s_{N-1}) + c(y-b)}{b} \right)$$

or equivalently

$$K + G_{N-1}(y+z) \geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right). \quad (4.29)$$

Now if  $y$  is such that  $G_{N-1}(y) \geq G_{N-1}(s_{N-1})$ , then by  $K$ -convexity of  $G_{N-1}$  we have

$$\begin{aligned} K + G_{N-1}(y+z) &\geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{y - s_{N-1}} \right) \\ &\geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right). \end{aligned}$$

Thus Eq. (4.29) and hence also Eq. (4.28) hold. If  $y$  is such that  $G_{N-1}(y) < G_{N-1}(s_{N-1})$ , then we have

$$\begin{aligned} K + G_{N-1}(y+z) &\geq K + G_{N-1}(S_{N-1}) \\ &= G_{N-1}(s_{N-1}) \\ &> G_{N-1}(y) \\ &\geq G_{N-1}(y) + z \left( \frac{G_{N-1}(y) - G_{N-1}(s_{N-1})}{b} \right). \end{aligned}$$

So for this case, Eq. (4.29) holds, and hence also the desired  $K$ -convexity Eq. (4.28) holds.

*Case 2:*  $y \leq y + z \leq s_{N-1}$ . In this region, by Eq. (4.27), the function  $J_{N-1}$  is linear and hence the  $K$ -convexity Eq. (4.28) holds.

*Case 3:*  $y < s_{N-1} < y + z$ . For this case, in view of Eq. (4.28), we can write the  $K$ -convexity Eq. (4.28) as

$$K + G_{N-1}(y + z) - c(y + z) \geq G_{N-1}(s_{N-1}) - cy \\ + z \left( \frac{G_{N-1}(s_{N-1}) - cy - G_{N-1}(s_{N-1}) + c(y - b)}{b} \right),$$

or equivalently

$$K + G_{N-1}(y + z) \geq G_{N-1}(s_{N-1}),$$

which holds by the definition of  $s_{N-1}$ .

We have thus proved that  $K$ -convexity and continuity of  $G_{N-1}$ , together with the fact that  $G_{N-1}(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ , imply  $K$ -convexity of  $J_{N-1}$ . In addition,  $J_{N-1}$  can be seen to be continuous. Now using Lemma 4.2.1, it follows from Eq. (4.23) that  $G_{N-2}$  is a  $K$ -convex function. Furthermore, by using the boundedness of  $w_{N-2}$ , it follows that  $G_{N-2}$  is continuous and, in addition,  $G_{N-2}(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ . Repeating the preceding argument, we obtain that  $J_{N-2}$  is  $K$ -convex, and proceeding similarly, we prove  $K$ -convexity and continuity of the functions  $G_k$  for all  $k$ , as well as that  $G_k(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ . At the same time [by using part (d) of Lemma 4.2.1] we prove optimality of the multiperiod  $(s, S)$  policy of Eq. (4.25).

The optimality of policies of the  $(s, S)$  type can be proved for several other inventory problems (see Exercises 4.3 to 4.10).

### 4.3 DYNAMIC PORTFOLIO ANALYSIS

Portfolio theory deals with the question of how to invest a certain amount of wealth among a collection of assets, perhaps over a long time interval. One approach, to be discussed in this section, is to assume that an investor makes a decision in each of several successive time periods with the objective of maximizing final wealth. We will start with an analysis of a single-period model and then extend the results to the multiperiod case.

Let  $x_0$  denote the initial wealth of the investor and assume that there are  $n$  risky assets, with corresponding random rates of return  $e_1, \dots, e_n$  among which the investor can allocate his wealth. The investor can also invest in a riskless asset offering a sure rate of return  $s$ . If we denote by  $u_1, \dots, u_n$  the corresponding amounts invested in the  $n$  risky assets and by

$(x_0 - u_1 - \dots - u_n)$  the amount invested in the riskless asset, the wealth at the end of the first period is given by

$$x_1 = s(x_0 - u_1 - \dots - u_n) + \sum_{i=1}^n e_i u_i,$$

or equivalently

$$x_1 = sx_0 + \sum_{i=1}^n (e_i - s) u_i.$$

The objective is to maximize over  $u_1, \dots, u_n$ ,

$$E\{U(x_1)\},$$

where  $U$  is a known function, referred to as the *utility function* for the investor (Appendix G contains a discussion of utility functions and their significance in the formulation of optimization problems under uncertainty). We assume that  $U$  is concave and twice continuously differentiable, and that the given expected value is well defined and finite for all  $x_0$  and  $u_i$ . We will not impose constraints on  $u_1, \dots, u_n$ . This is necessary in order to obtain the results in convenient form. A few additional assumptions will be made later.

Let us denote by  $u_i^* = \mu^{i*}(x_0)$ ,  $i = 1, \dots, n$ , the optimal amounts to be invested in the  $n$  risky assets when the initial wealth is  $x_0$ . We will show that when the utility function satisfies

$$-\frac{U'(x)}{U''(x)} = a + bx, \quad \text{for all } x, \quad (4.30)$$

where  $U'$  and  $U''$  denote the first and second derivatives of  $U$ , respectively, and  $a$  and  $b$  are some scalars, then the optimal portfolio is given by the linear policy

$$\mu^{i*}(x_0) = \alpha^i(a + bx_0), \quad i = 1, \dots, n, \quad (4.31)$$

where  $\alpha^i$  are some constant scalars. Furthermore, if  $J(x_0)$  is the optimal value of the problem

$$J(x_0) = \max_{u_i} E\{U(x_1)\},$$

then we have

$$-\frac{J'(x_0)}{J''(x_0)} = \frac{a}{s} + bx_0, \quad \text{for all } x_0. \quad (4.32)$$

It can be verified that the following utility functions  $U(x)$  satisfy condition (4.30):

exponential :  $e^{-x/a}$ , for  $b = 0$ ,  $a > 0$ ,

logarithmic :  $\ln(x + a)$ , for  $b = 1$ ,

power :  $(1/(b-1))(a + bx)^{1-(1/b)}$ , for  $b \neq 0$ ,  $b \neq 1$ .

Only concave utility functions from this class are admissible for our problem. Furthermore, if a utility function that is not defined over the whole real line is used, the problem should be formulated in a way that ensures that all possible values of the resulting final wealth are within the domain of definition of the utility function.

To show the desired relations, let us hypothesize that an optimal portfolio exists and is of the form

$$\mu^{i*}(x_0) = \alpha^i(x_0)(a + bsx_0),$$

where  $\alpha^i(x_0)$ ,  $i = 1, \dots, n$ , are some differentiable functions. We will prove that  $d\alpha^i(x_0)/dx_0 = 0$  for all  $x_0$ , implying that the functions  $\alpha^i$  must be constant, so the optimal portfolio has the linear form (4.31).

We have for every  $x_0$  and  $i = 1, \dots, n$ , by the optimality of  $\mu^{i*}(x_0)$ ,

$$\begin{aligned} \frac{dE\{U(x_1)\}}{du_i} &= E \left\{ U' \left( sx_0 + \sum_{j=1}^n (e_j - s) \alpha^j(x_0)(a + bsx_0) \right) (e_i - s) \right\} \\ &= 0. \end{aligned} \quad (4.33)$$

Differentiating each of the  $n$  equations above with respect to  $x_0$  yields

$$\begin{aligned} E \left\{ \begin{pmatrix} (e_1 - s)^2 \cdots (e_1 - s)(e_n - s) \\ \vdots \\ (e_n - s)(e_1 - s) \cdots (e_n - s)^2 \end{pmatrix} U''(x_1)(a + bsx_0) \right\} \begin{pmatrix} \frac{d\alpha^1(x_0)}{dx_0} \\ \vdots \\ \frac{d\alpha^n(x_0)}{dx_0} \end{pmatrix} \\ = - \begin{pmatrix} E\{U''(x_1)(e_1 - s)s(1 + \sum_{i=1}^n (e_i - s)\alpha^i(x_0)b)\} \\ \vdots \\ E\{U''(x_1)(e_n - s)s(1 + \sum_{i=1}^n (e_i - s)\alpha^i(x_0)b)\} \end{pmatrix}. \end{aligned} \quad (4.34)$$

Using relation (4.31), we have

$$\begin{aligned} U''(x_1) &= - \frac{U'(x_1)}{a + b(sx_0 + \sum_{i=1}^n (e_i - s)\alpha^i(x_0)(a + bsx_0))} \\ &= - \frac{U'(x_1)}{(a + bsx_0)(1 + \sum_{i=1}^n (e_i - s)\alpha^i(x_0)b)}. \end{aligned} \quad (4.35)$$

Substituting in Eq. (4.34) and using Eq. (4.33), we have that the right-hand side of Eq. (4.34) is the zero vector. The matrix on the left in Eq. (4.34), except for degenerate cases, can be shown to be nonsingular. Assuming that it is indeed nonsingular, we obtain

$$\frac{d\alpha^i(x_0)}{dx_0} = 0, \quad i = 1, \dots, n,$$

and  $\alpha^i(x_0) \rightarrow \alpha^i$ , where  $\alpha^i$  are some constants, thus proving the optimality of the linear policy (4.31).

We now prove Eq. (4.32). We have

$$\begin{aligned} J(x_0) &= E\{U(x_1)\} \\ &= E \left\{ U \left( s \left( 1 + \sum_{i=1}^n (e_i - s)\alpha^i b \right) x_0 + \sum_{i=1}^n (e_i - s)\alpha^i a \right) \right\} \end{aligned}$$

and hence

$$\begin{aligned} J'(x_0) &= E \left\{ U'(x_1)s \left( 1 + \sum_{i=1}^n (e_i - s)\alpha^i b \right) \right\}, \\ J''(x_0) &= E \left\{ U''(x_1)s^2 \left( 1 + \sum_{i=1}^n (e_i - s)\alpha^i b \right)^2 \right\}. \end{aligned} \quad (4.36)$$

The last relation after some calculation using Eq. (4.35) yields

$$J''(x_0) = - \frac{E\{U'(x_1)s(1 + \sum_{i=1}^n (e_i - s)\alpha^i b)\}s}{a + bsx_0}. \quad (4.37)$$

By combining Eqs. (4.36) and (4.37), we obtain the desired result:

$$-\frac{J'(x_0)}{J''(x_0)} = \frac{a}{s} + bx_0.$$

### The Multiperiod Problem

We now extend the preceding one-period analysis to the multiperiod case. We will assume that the current wealth can be reinvested at the beginning of each of  $N$  consecutive time periods. We denote

$x_k$ : the wealth of the investor at the start of the  $k$ th period,

$u_i^k$ : the amount invested at the start of the  $k$ th period in the  $i$ th risky asset,

$e_i^k$ : the rate of return of the  $i$ th risky asset during the  $k$ th period,

$s_k$ : the rate of return of the riskless asset during the  $k$ th period.

We have the system equation

$$x_{k+1} = s_k x_k + \sum_{i=1}^n (e_i^k - s_k) u_i^k, \quad k = 0, 1, \dots, N-1.$$

We assume that the vectors  $e^k = (e_1^k, \dots, e_n^k)$ ,  $k = 0, \dots, N - 1$ , are independent with given probability distributions that yield finite expected values throughout the following analysis. The objective is to maximize  $E\{U(x_N)\}$ , the expected utility of the terminal wealth  $x_N$ , where we assume that  $U$  satisfies

$$-\frac{U'(x)}{U''(x)} = a + bx, \quad \text{for all } x.$$

Applying the DP algorithm to this problem, we have

$$J_N(x_N) = U(x_N),$$

$$J_k(x_k) = \max_{u_1^k, \dots, u_n^k} E \left\{ J_{k+1} \left( s_k x_k + \sum_{i=1}^n (e_i^k - s_k) u_i^k \right) \right\}. \quad (4.38)$$

From the solution of the one-period problem, we have that the optimal policy at period  $N - 1$  has the form

$$\mu_{N-1}^*(x_{N-1}) = \alpha_{N-1}(a + b s_{N-1} x_{N-1}),$$

where  $\alpha_{N-1}$  is an appropriate  $n$ -dimensional vector. Furthermore, we have

$$-\frac{J'_{N-1}(x)}{J''_{N-1}(x)} = \frac{a}{s_{N-1}} + bx.$$

Hence, applying the one-period result in the DP Eq. (4.38) for the next to the last period, we obtain the optimal policy

$$\mu_{N-2}^*(x_{N-2}) = \alpha_{N-2} \left( \frac{a}{s_{N-1}} + b s_{N-2} x_{N-2} \right),$$

where  $\alpha_{N-2}$  is again an appropriate  $n$ -dimensional vector.

Proceeding similarly, we have for the  $k$ th period

$$\mu_k^*(x_k) = \alpha_k \left( \frac{a}{s_{N-1} \cdots s_{k+1}} + b s_k x_k \right), \quad (4.39)$$

where  $\alpha_k$  is an  $n$ -dimensional vector that depends on the probability distributions of the rates of return  $e_i^k$  of the risky assets and are determined by maximization in the DP Eq. (4.38). The corresponding cost-to-go functions satisfy

$$-\frac{J'_k(x)}{J''_k(x)} = \frac{a}{s_{N-1} \cdots s_k} + bx, \quad k = 0, 1, \dots, N - 1. \quad (4.40)$$

Thus it is seen that the investor, when faced with the opportunity to sequentially reinvest his wealth, uses a policy similar to that of the single-period case. Carrying the analysis one step further, it is seen that if the utility function  $U$  is such that  $a = 0$ , that is,  $U$  has one of the forms

$$\begin{aligned} \ln x, & \quad \text{for } b = 1, \\ \left( \frac{1}{b-1} \right) (bx)^{1-(1/b)}, & \quad \text{for } b \neq 0, b \neq 1, \end{aligned}$$

then it follows from Eq. (4.39) that the investor acts at each stage  $k$  as if he were faced with a *single-period* investment characterized by the rates of return  $s_k$  and  $e_i^k$ , and the objective function  $E\{U(x_{k+1})\}$ . This policy whereby the investor can ignore the fact that he will have the opportunity to reinvest his wealth is called a *myopic policy*.

Note that a myopic policy is also optimal when  $s_k = 1$  for all  $k$ , which means that wealth is discounted at the rate of return of the riskless asset. Furthermore, it has been shown by Mossin [Mos68] that when  $a = 0$  a myopic policy is optimal even in the more general case where the rates of return  $s_k$  are independent random variables, and for the case where forecasts on the probability distributions of the rates of return  $e_i^k$  of the risky assets become available during the investment process (see Exercise 4.14).

It turns out that even in the more general case where  $a \neq 0$  only a small amount of foresight is required on the part of the decision maker. It can be seen [compare Eqs. (4.38)-(4.40)] that the optimal policy (4.39) at period  $k$  is the one that the investor would use in a single-period problem to maximize over  $u_i^k$ ,  $i = 1, \dots, n$ ,

$$E\{U(s_{N-1} \cdots s_{k+1} x_{k+1})\}$$

subject to

$$x_{k+1} = s_k x_k + \sum_{i=1}^n (e_i^k - s_k) u_i^k.$$

In other words, the investor at period  $k$  should maximize the expected utility of wealth that results if amounts  $u_i^k$  are invested in the risky assets in period  $k$  and the resulting wealth  $x_{k+1}$  is subsequently invested *exclusively* in the riskless asset during the remaining periods  $k + 1, \dots, N - 1$ . This is known as a *partially myopic policy*. Such a policy can also be shown to be optimal when forecasts on the probability distributions of the rates of return of the risky assets become available during the investment process (see Exercise 4.14).

Another interesting aspect of the case where  $a \neq 0$  is that if  $s_k > 1$  for all  $k$ , then as the horizon becomes increasingly long ( $N \rightarrow \infty$ ), the policy in the initial stages approaches a myopic policy [compare Eqs. (4.39) and (4.40)]. Thus, for  $s_k > 1$ , a partially myopic policy becomes asymptotically myopic as the horizon tends to infinity.

#### 4.4 OPTIMAL STOPPING PROBLEMS

Optimal stopping problems of the type that we will consider in this and subsequent sections are characterized by the availability, at each state, of a control that stops the evolution of the system. Thus at each stage the decision maker observes the current state of the system and decides whether to continue the process (perhaps at a certain cost) or stop the process and incur a certain loss. If the decision is to continue, a control must be selected from a given set of available choices. If there is only one choice other than stopping, then each policy is characterized at each period by the *stopping set*, that is, the set of states where the policy stops the system.

##### Asset Selling

As a first example, consider a person having an asset (say a piece of land) for which he is offered an amount of money from period to period. We assume that the offers, denoted  $w_0, w_1, \dots, w_{N-1}$ , are random and independent, and take values within some bounded interval of nonnegative numbers ( $w_k = 0$  could correspond to no offer received during the period). If the person accepts an offer, he can invest the money at a fixed rate of interest  $r > 0$ , and if he rejects the offer, he waits until the next period to consider the next offer. Offers rejected are not renewed, and we assume that the last offer  $w_{N-1}$  must be accepted if every prior offer has been rejected. The objective is to find a policy for accepting and rejecting offers that maximizes the revenue of the person at the  $N$ th period.

The DP algorithm for this problem can be derived by elementary reasoning. As a modeling exercise, however, we will embed the problem in the framework of the basic problem by specifying the system and cost. We define the state space to be the real line, augmented with an additional state (call it  $T$ ), which is a *termination state*. By writing that the system is at state  $x_k = T$  at some time  $k \leq N - 1$ , we mean that the asset has already been sold. By writing that the system is at a state  $x_k \neq T$  at some time  $k \leq N - 1$ , we mean that the asset has not been sold as yet and the offer under consideration is equal to  $x_k$  (and also equal to the  $k$ th offer  $w_{k-1}$ ). We take  $x_0 = 0$  (a fictitious "null" offer). The control space consists of two elements  $u^1$  and  $u^2$ , corresponding to the decisions "sell" and "do not sell," respectively. We view  $w_k$  as the disturbance at time  $k$ .

With these conventions, we may write a system equation of the form

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N - 1,$$

where the function  $f_k$  is defined via the relation

$$x_{k+1} = \begin{cases} T & \text{if } x_k = T, \text{ or if } x_k \neq T \text{ and } u_k = u^1 \text{ (sell),} \\ w_k & \text{otherwise.} \end{cases}$$

Note that a sell decision at time  $k$  ( $u_k = u^1$ ) accepts the offer  $w_{k-1}$ , and that no explicit sell decision is required to accept the last offer  $w_{N-1}$ , as it must be accepted by assumption if the asset has not yet been sold. The corresponding reward function may be written as

$$E_{w_k} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\}$$

where

$$g_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T, \\ 0 & \text{otherwise,} \end{cases}$$

$$g_k(x_k, u_k, w_k) = \begin{cases} (1+r)^{N-k} x_k & \text{if } x_k \neq T \text{ and } u_k = u^1 \text{ (sell),} \\ 0 & \text{otherwise.} \end{cases}$$

Based on this formulation we can write the corresponding DP algorithm:

$$J_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq T, \\ 0 & \text{if } x_N = T, \end{cases} \quad (4.41)$$

$$J_k(x_k) = \begin{cases} \max \left[ (1+r)^{N-k} x_k, E \{ J_{k+1}(w_k) \} \right] & \text{if } x_k \neq T, \\ 0 & \text{if } x_k = T. \end{cases} \quad (4.42)$$

In the above equation,  $(1+r)^{N-k} x_k$  is the revenue resulting from decision  $u^1$  (sell) when the offer is  $x_k$ , and  $E \{ J_{k+1}(w_k) \}$  represents the expected revenue corresponding to the decision  $u^2$  (do not sell). Thus, the optimal policy is to accept an offer if it is greater than  $E \{ J_{k+1}(w_k) \} / (1+r)^{N-k}$ , which represents expected revenue discounted to the present time:

$$\text{accept the offer } x_k \quad \text{if } x_k > \alpha_k,$$

$$\text{reject the offer } x_k \quad \text{if } x_k < \alpha_k,$$

where

$$\alpha_k = \frac{E \{ J_{k+1}(w_k) \}}{(1+r)^{N-k}}.$$

When  $x_k = \alpha_k$ , both acceptance and rejection are optimal. Thus the optimal policy is determined by the scalar sequence  $\{\alpha_k\}$  (see Fig. 4.4.1).

##### Properties of the Optimal Policy

We will now derive some properties of the optimal policy with some further analysis. Let us assume that the offers  $w_k$  are identically distributed, and to simplify notation, let us drop the time index  $k$  and denote by  $E_w \{\cdot\}$  the expected value of the corresponding expression over  $w_k$ , for all  $k$ . We will then show that

$$\alpha_k \geq \alpha_{k+1}, \quad \text{for all } k,$$

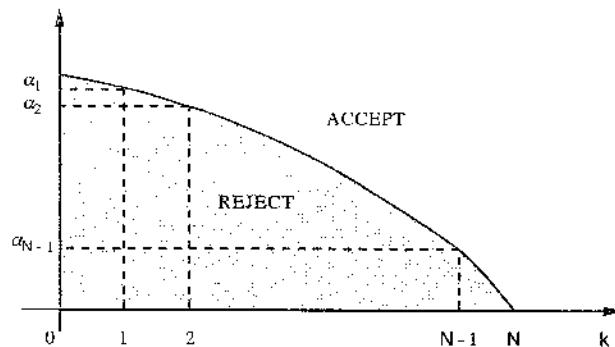


Figure 4.4.1 Optimal policy for accepting offers in the asset selling problem.

which expresses the intuitive fact that if an offer is good enough to be acceptable at time  $k$ , it should also be acceptable at time  $k+1$  when there will be one less chance for improvement. We will also obtain an equation for the limit of  $\alpha_k$  as  $k \rightarrow -\infty$ .

Let us introduce the functions

$$V_k(x_k) = \frac{J_k(x_k)}{(1+r)^{N-k}}, \quad x_k \neq T.$$

It can be seen from Eqs. (4.41) and (4.42) that

$$V_N(x_N) = x_N, \quad (4.43)$$

$$V_k(x_k) = \max \left[ x_k, (1+r)^{-1} \underset{w}{E} \{ V_{k+1}(w) \} \right], \quad k = 0, 1, \dots, N-1, \quad (4.44)$$

and that

$$\alpha_k = \frac{\underset{w}{E} \{ V_{k+1}(w) \}}{1+r}.$$

To prove that  $\alpha_k \geq \alpha_{k+1}$ , note that from Eqs. (4.43) and (4.44), we have

$$V_{N-1}(x) \geq V_N(x), \quad \text{for all } x \geq 0.$$

Applying Eq. (4.44) for  $k = N-2$  and  $k = N-1$ , and using the preceding inequality, we obtain for all  $x \geq 0$

$$\begin{aligned} V_{N-2}(x) &= \max \left[ x, (1+r)^{-1} \underset{w}{E} \{ V_{N-1}(w) \} \right] \\ &\geq \max \left[ x, (1+r)^{-1} \underset{w}{E} \{ V_N(w) \} \right] \\ &= V_{N-1}(x). \end{aligned}$$

Continuing in the same manner, we see that

$$V_k(x) \geq V_{k+1}(x), \quad \text{for all } x \geq 0 \text{ and } k.$$

Since  $\alpha_k = \underset{w}{E} \{ V_{k+1}(w) \} / (1+r)$ , we obtain  $\alpha_k \geq \alpha_{k+1}$ , as desired.

Let us now see what happens when the horizon  $N$  is very large. From the algorithm (4.43) and (4.44) we have

$$V_k(x_k) = \max(x_k, \alpha_k). \quad (4.45)$$

Hence we obtain

$$\begin{aligned} \alpha_k &= \frac{1}{1+r} \underset{w}{E} \{ V_{k+1}(w) \} \\ &= \frac{1}{1+r} \int_0^{\alpha_{k+1}} \alpha_{k+1} dP(w) + \frac{1}{1+r} \int_{\alpha_{k+1}}^{\infty} wdP(w), \end{aligned}$$

where the function  $P$  is defined for all scalars  $\lambda$  by

$$P(\lambda) = \text{Prob}\{w < \lambda\}.$$

The difference equation for  $\alpha_k$  may also be written as

$$\alpha_k = \frac{P(\alpha_{k+1})}{1+r} \alpha_{k+1} + \frac{1}{1+r} \int_{\alpha_{k+1}}^{\infty} wdP(w), \quad \text{for all } k, \quad (4.46)$$

with  $\alpha_N = 0$ .

Now since we have

$$0 \leq \frac{P(\alpha)}{1+r} \leq \frac{1}{1+r} < 1, \quad \text{for all } \alpha \geq 0,$$

$$0 \leq \frac{1}{1+r} \int_{\alpha_{k+1}}^{\infty} wdP(w) \leq \frac{E\{w\}}{1+r}, \quad \text{for all } k,$$

it can be seen, using the property  $\alpha_k \geq \alpha_{k+1}$ , that the sequence  $\{\alpha_k\}$  generated (backward) by the difference equation (4.46) converges (as  $k \rightarrow -\infty$ ) to a constant  $\bar{\alpha}$  satisfying

$$(1+r)\bar{\alpha} = P(\bar{\alpha})\bar{\alpha} + \int_{\bar{\alpha}}^{\infty} wdP(w).$$

This equation is obtained from Eq. (4.46) by taking limits as  $k \rightarrow -\infty$  and by using the fact that  $P$  is continuous from the left.

Thus, when the horizon  $N$  tends to become longer, the optimal policy for every fixed  $k \geq 1$  can be approximated by the stationary policy

$$\begin{aligned} \text{accept the offer } x_k &\quad \text{if } x_k > \bar{\alpha}, \\ \text{reject the offer } x_k &\quad \text{if } x_k < \bar{\alpha}. \end{aligned}$$

The optimality of this policy for the corresponding infinite horizon problem will be shown in Section 7.3.

### Purchasing with a Deadline

Let us consider another stopping problem that has similar nature. Assume that a certain quantity of raw material is required by a certain time. If the price of this material fluctuates, there arises the problem of deciding whether to purchase at the current price or wait a further period, during which the price may go up or down. We thus want to minimize the expected price of purchase. We assume that successive prices  $w_k$  are independent and identically distributed with distribution  $P(w_k)$ , and that the purchase must be made within  $N$  time periods.

This problem and the earlier asset selling problem have obvious similarities. Let us denote by

$$x_{k+1} = w_k$$

the price prevailing at the beginning of period  $k+1$ . We have similar to the earlier problem the DP algorithm

$$J_N(x_N) = x_N,$$

$$J_k(x_k) = \min [x_k, E\{J_{k+1}(w_k)\}].$$

Note that  $J_k(x_k)$  is the optimal cost-to-go when the current price is  $x_k$  and the material has not been purchased yet. To be strictly formal, we should introduce a termination state  $T$ , to which the system moves following a purchasing decision and at which the system subsequently stays at no cost. A nonzero cost is incurred only when moving from  $x_k$  to  $T$ ; this cost is equal to  $x_k$ . Thus the cost-to-go from the termination state  $T$  is 0, and for this reason it was neglected in the preceding DP equation.

The optimal policy is given by

purchase if  $x_k < \alpha_k$ ,

do not purchase if  $x_k > \alpha_k$ ,

where

$$\alpha_k = E\{J_{k+1}(w_k)\}.$$

Similar to the asset selling problem, the thresholds  $\alpha_1, \alpha_2, \dots, \alpha_{N-1}$  can be obtained from the discrete-time equation

$$\alpha_k = \alpha_{k+1}(1 - P(\alpha_{k+1})) + \int_0^{\alpha_{k+1}} wdP(w),$$

with the terminal condition

$$\alpha_{N-1} = \int_0^\infty wdP(w) = E\{w\}.$$

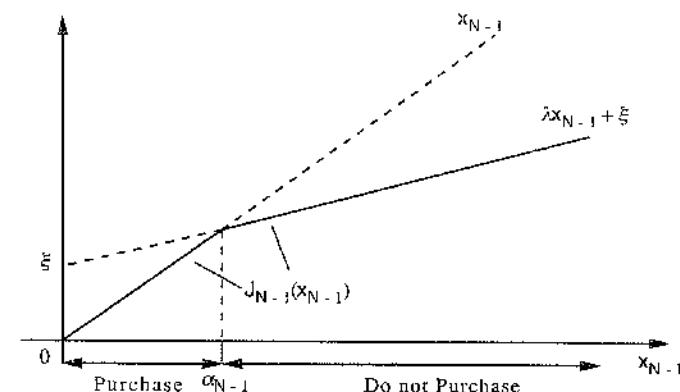


Figure 4.4.2 Structure of the cost-to-go function  $J_{N-1}(x_{N-1})$  when prices are correlated.

### The Case of Correlated Prices

Consider now a variation of the purchasing problem where we do not assume that the successive prices  $w_0, \dots, w_{N-1}$  are independent. Instead, we assume that they are correlated and can be represented as the state of a linear system driven by independent disturbances (cf. Section 1.4). In particular, we have

$$w_k = x_{k+1}, \quad k = 0, 1, \dots, N-1,$$

with

$$x_{k+1} = \lambda x_k + \xi_k, \quad x_0 = 0,$$

where  $\lambda$  is a scalar with  $0 \leq \lambda < 1$  and  $\xi_0, \xi_1, \dots, \xi_{N-1}$  are independent identically distributed random variables taking positive values with given probability distribution. As discussed in Section 1.4, the DP algorithm under these circumstances takes the form

$$J_N(x_N) = x_N,$$

$$J_k(x_k) = \min [x_k, E\{J_{k+1}(\lambda x_k + \xi_k)\}],$$

where the cost associated with the purchasing decision is  $x_k$  and the cost associated with the waiting decision is  $E\{J_{k+1}(\lambda x_k + \xi_k)\}$ .

We will show that in this case the optimal policy is of the same type as the one for independent prices. Indeed, we have

$$J_{N-1}(x_{N-1}) = \min[x_{N-1}, \lambda x_{N-1} + \bar{\xi}],$$

where  $\bar{\xi} = E\{\xi_{N-1}\}$ . As shown in Fig. 4.4.2, an optimal policy at time  $N - 1$  is given by

purchase if  $x_{N-1} < \alpha_{N-1}$ ,

do not purchase if  $x_{N-1} > \alpha_{N-1}$ ,

where  $\alpha_{N-1}$  is defined from the equation  $\alpha_{N-1} = \lambda\alpha_{N-1} + \bar{\xi}$ :

$$\alpha_{N-1} = \frac{\bar{\xi}}{1 - \lambda}.$$

Note that

$$J_{N-1}(x) \leq J_N(x), \quad \text{for all } x,$$

and that  $J_{N-1}$  is concave and increasing in  $x$ . Using this fact in the DP algorithm, one may show (the monotonicity property of DP; Exercise 1.23 in Chapter 1) that

$$J_k(x) \leq J_{k+1}(x), \quad \text{for all } x \text{ and } k,$$

and that  $J_k$  is concave and increasing in  $x$  for all  $k$ . Furthermore, since  $\bar{\xi} = E\{\xi_k\} > 0$  for all  $k$ , one can show that

$$E\{J_{k+1}(\xi_k)\} > 0, \quad \text{for all } k.$$

These facts imply (as illustrated in Fig. 4.4.3) that the optimal policy for every period  $k$  is of the form

purchase if  $x_k < \alpha_k$ ,

do not purchase if  $x_k > \alpha_k$ ,

where the scalar  $\alpha_k$  is the unique positive solution of the equation

$$x = E\{J_{k+1}(\lambda x + \xi_k)\}.$$

Note that the relation  $J_k(x) \leq J_{k+1}(x)$  for all  $x$  and  $k$  implies that

$$\alpha_{k-1} \leq \alpha_k \leq \alpha_{k+1}, \quad \text{for all } k,$$

and hence (as one would expect) the threshold price to purchase increases as the deadline gets closer.

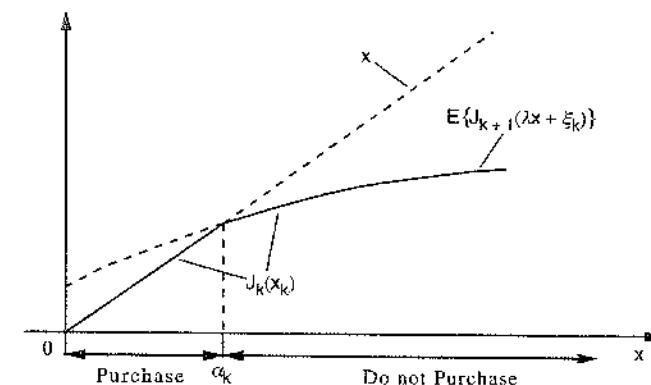


Figure 4.4.3 Determining the optimal policy when prices are correlated.

#### General Stopping Problems and the One-Step-Lookahead Rule

We now formulate a general type of  $N$ -stage problem where stopping is mandatory at or before stage  $N$ . Consider the stationary version of the basic problem of Chapter 1 (state, control, and disturbance spaces, disturbance distribution, control constraint set, and cost per stage are the same for all times). Assume that at each state  $x_k$  and at time  $k$  there is available, in addition to the controls  $u_k \in U(x_k)$ , a stopping action that forces the system to enter a termination state at a cost  $t(x_k)$  and subsequently remain there at no cost. The terminal cost, assuming stopping has not occurred by the last stage, is  $t(x_N)$ . Thus, in effect, we assume that the termination cost will always be incurred either at the end of the horizon or earlier.

The DP algorithm is given by

$$J_N(x_N) = t(x_N), \quad (4.47)$$

$$J_k(x_k) = \min \left[ t(x_k), \min_{u_k \in U(x_k)} E\{g(x_k, u_k, w_k) + J_{k+1}(f(x_k, u_k, w_k))\} \right], \quad (4.48)$$

and it is optimal to stop at time  $k$  for states  $x$  in the set

$$T_k = \left\{ x \mid t(x) \leq \min_{u \in U(x)} E\{g(x, u, w) + J_{k+1}(f(x, u, w))\} \right\}.$$

We have from Eqs. (4.47) and (4.48)

$$J_{N-1}(x) \leq J_N(x), \quad \text{for all } x,$$

and using this fact in the DP equation (4.48), we obtain inductively

$$J_k(x) \leq J_{k+1}(x), \quad \text{for all } x \text{ and } k.$$

We are making use here of the stationarity of the problem and the monotonicity property of DP (Exercise 1.23 in Chapter 1).] Using this fact and the definition of  $T_k$  we see that

$$T_0 \subset \cdots \subset T_k \subset T_{k+1} \subset \cdots \subset T_{N-1}. \quad (4.49)$$

We will now consider a condition guaranteeing that all the stopping sets  $T_k$  are equal. Suppose that the set  $T_{N-1}$  is *absorbing* in the sense that if a state belongs to  $T_{N-1}$  and termination is not selected, the next state will also be in  $T_{N-1}$ :

$$f(x, u, w) \in T_{N-1}, \quad \text{for all } x \in T_{N-1}, u \in U(x), w. \quad (4.50)$$

We will show that equality holds in Eq. (4.49) and for all  $k$  we have

$$T_k = T_{N-1} = \left\{ x \in S \mid t(x) \leq \min_{u \in U(x)} E \left\{ g(x, u, w) + t(f(x, u, w)) \right\} \right\}.$$

To see this, note that by the definition of  $T_{N-1}$ , we have

$$J_{N-1}(x) = t(x), \quad \text{for all } x \in T_{N-1},$$

and using Eq. (4.50) we obtain for  $x \in T_{N-1}$

$$\begin{aligned} \min_{u \in U(x)} E \left\{ g(x, u, w) + J_{N-1}(f(x, u, w)) \right\} \\ = \min_{u \in U(x)} E \left\{ g(x, u, w) + t(f(x, u, w)) \right\} \\ \geq t(x). \end{aligned}$$

Therefore, stopping is optimal for all  $x_{N-2} \in T_{N-1}$  or equivalently  $T_{N-1} \subset T_{N-2}$ . This together with Eq. (4.49) implies  $T_{N-2} = T_{N-1}$ . Proceeding similarly, we obtain  $T_k = T_{N-1}$  for all  $k$ .

In conclusion, if condition (4.50) holds (the one-step stopping set  $T_{N-1}$  is absorbing), then the stopping sets  $T_k$  are all equal to the set of states for which it is better to stop rather than continue for one more stage and then stop. A policy of this type is known as a *one-step lookahead policy*. Such a policy turns out to be optimal in several types of applications. We provide next some examples. Additional examples are given in the exercises, and in Vol. II, Chapter 3.

#### Example 4.4.1 (Asset Selling with Past Offers Retained)

Consider the asset selling problem discussed earlier in this section with the difference that rejected offers can be accepted at a later time. Then if the asset is not sold at time  $k$  the state evolves according to

$$x_{k+1} = \max(x_k, w_k)$$

instead of  $x_{k+1} = w_k$ . The DP equations (4.43) and (4.44) then become

$$V_N(x_N) = x_N,$$

$$V_k(x_k) = \max[x_k, (1+r)^{-1} E \{ V_{k+1}(\max(x_k, w_k)) \}].$$

The one-step stopping set is

$$T_{N-1} = \{x \mid x \geq (1+r)^{-1} E \{ \max(x, w) \}\}.$$

It is seen [compare with Eqs. (4.45) and (4.46)] that an alternative characterization is

$$T_{N-1} = \{x \mid x \geq \bar{\alpha}\}, \quad (4.51)$$

where  $\bar{\alpha}$  is obtained from the equation

$$(1+r)\bar{\alpha} = P(\bar{\alpha})\bar{\alpha} + \int_{\bar{\alpha}}^{\infty} wdP(w).$$

Since past offers can be accepted at a later date, the effective offer available cannot decrease with time, and it follows that the one-step stopping set (4.51) is absorbing in the sense of Eq. (4.50). Therefore, the one-step lookahead stopping rule that accepts the first offer that equals or exceeds  $\bar{\alpha}$  is optimal. Note that this policy is independent of the horizon length  $N$ .

#### Example 4.4.2 (The Rational Burglar [Whi82])

A burglar may at any night  $k$  choose to retire with his accumulated earnings  $x_k$  or enter a house and bring home an amount  $w_k$ . However, in the latter case he gets caught with probability  $p$ , and then he is forced to terminate his activities and forfeit his earnings thus far. The amounts  $w_k$  are independent, identically distributed with mean  $\bar{w}$ . The problem is to find a policy that maximizes the burglar's expected earnings over  $N$  nights.

We can formulate this problem as a stopping problem with two actions (retire or continue) and a state space consisting of the real line, the retirement state, and a special state corresponding to the burglar getting caught. The DP algorithm is given by

$$J_N(x_N) = x_N$$

$$J_k(x_k) = \max \left[ x_k, (1-p)E \{ J_{k+1}(x_k + w_k) \} \right].$$

The one-step stopping set is

$$T_{N-1} = \{x \mid x \geq (1-p)(x + \bar{w})\} = \left\{x \mid x \geq \frac{(1-p)\bar{w}}{p}\right\},$$

(more accurately this set together with the special state corresponding to the burglar's arrest). Since this set is absorbing in the sense of Eq. (4.50), we see that the one-step lookahead policy by which the burglar retires when his earnings reach or exceed  $(1-p)\bar{w}/p$  is optimal. The optimality of this policy for the corresponding infinite horizon problem will be demonstrated in Vol. II, Chapter 3.

## 4.5 SCHEDULING AND THE INTERCHANGE ARGUMENT

Suppose one has a collection of tasks to perform but the ordering of the tasks is subject to optimal choice. As examples, consider the ordering of operations in a construction project so as to minimize construction time or the scheduling of jobs in a workshop so as to minimize machine idle time. In such problems a useful technique is to start with some schedule and then to interchange two adjacent tasks and see what happens. We first provide some examples, and we then formalize mathematically the technique.

### Example 4.5.1 (The Quiz Problem)

Consider a quiz contest where a person is given a list of  $N$  questions and can answer these questions in any order he chooses. Question  $i$  will be answered correctly with probability  $p_i$ , and the person will then receive a reward  $R_i$ . At the first incorrect answer, the quiz terminates and the person is allowed to keep his previous rewards. The problem is to choose the ordering of questions so as to maximize expected rewards.

Let  $i$  and  $j$  be the  $k$ th and  $(k+1)$ st questions in an optimally ordered list

$$L = (i_0, \dots, i_{k-1}, i, j, i_{k+2}, \dots, i_{N-1}).$$

Consider the list

$$L' = (i_0, \dots, i_{k-1}, j, i, i_{k+2}, \dots, i_{N-1})$$

obtained from  $L$  by interchanging the order of questions  $i$  and  $j$ . We compare the expected rewards of  $L$  and  $L'$ . We have

$$\begin{aligned} E\{\text{reward of } L\} &= E\{\text{reward of } \{i_0, \dots, i_{k-1}\}\} \\ &\quad + p_{i_0} \cdots p_{i_{k-1}} (p_i R_i + p_j p_i R_j) \\ &\quad + p_{i_0} \cdots p_{i_{k-1}} p_i p_j E\{\text{reward of } \{i_{k+2}, \dots, i_{N-1}\}\} \end{aligned}$$

$$\begin{aligned} E\{\text{reward of } L'\} &= E\{\text{reward of } \{i_0, \dots, i_{k-1}\}\} \\ &\quad + p_{i_0} \cdots p_{i_{k-1}} (p_j R_j + p_i p_i R_i) \\ &\quad + p_{i_0} \cdots p_{i_{k-1}} p_j p_i E\{\text{reward of } \{i_{k+2}, \dots, i_{N-1}\}\}. \end{aligned}$$

Since  $L$  is optimally ordered, we have

$$E\{\text{reward of } L\} \geq E\{\text{reward of } L'\},$$

so it follows from these equations that

$$p_i R_i + p_i p_j R_j \geq p_j R_j + p_j p_i R_i$$

or equivalently

$$\frac{p_i R_i}{1 - p_i} \geq \frac{p_j R_j}{1 - p_j}.$$

Therefore, to maximize expected rewards, questions should be answered in decreasing order of  $p_i R_i / (1 - p_i)$ .

### Example 4.5.2 (Job Scheduling on a Single Processor)

Suppose we have  $N$  jobs to process in sequential order with the  $i$ th job requiring a random time  $T_i$  for its execution. The times  $T_1, \dots, T_N$  are independent. If job  $i$  is completed at time  $t$ , the reward is  $\alpha^t R_i$ , where  $\alpha$  is a discount factor with  $0 < \alpha < 1$ . The problem is to find a schedule that maximizes the total expected reward.

It can be seen that the state for this problem is just the collection of jobs yet to be processed. Indeed, because the execution times  $T_i$  are independent, and also because future costs are multiplicatively affected through discounting by the completion times of preceding jobs, the optimization of the scheduling of future jobs is unaffected by the completion times of preceding jobs. As a result these times need not be included in the state; this would not be so if either the times  $T_i$  were correlated or if the reward for completing job  $i$  at time  $t$  were not  $\alpha^t R_i$  but instead had a general dependence on  $t$ . Now, given that the state is the collection of jobs yet to be processed, it is clear that an optimal policy can be mapped into an optimal job schedule  $(i_0, \dots, i_{N-1})$ .

Suppose that  $L = (i_0, \dots, i_{k-1}, i, j, i_{k+2}, \dots, i_{N-1})$  is an optimal job schedule, and consider the schedule  $L' = (i_0, \dots, i_{k-1}, j, i, i_{k+2}, \dots, i_{N-1})$  obtained by interchanging  $i$  and  $j$ . Let  $t_k$  be the time of completion of job  $i_{k-1}$ . We compare the rewards of the schedules  $L$  and  $L'$ , similar to the preceding example. Since the reward for completing the remaining jobs  $i_{k+2}, \dots, i_{N-1}$  is independent of the order in which jobs  $i$  and  $j$  are executed, we obtain

$$E\{\alpha^{t_k+T_i} R_i + \alpha^{t_k+T_i+T_j} R_j\} \geq E\{\alpha^{t_k+T_j} R_j + \alpha^{t_k+T_j+T_i} R_i\}.$$

Since  $t_k$ ,  $T_i$ , and  $T_j$  are independent, this relation can be written as

$$\begin{aligned} E\{\alpha^{t_k}\}(E\{\alpha^{T_i}\}R_i + E\{\alpha^{T_i}\}E\{\alpha^{T_j}\}R_j) \\ \geq E\{\alpha^{t_k}\}(E\{\alpha^{T_j}\}R_j + E\{\alpha^{T_j}\}E\{\alpha^{T_i}\}R_i), \end{aligned}$$

from which we finally obtain

$$\frac{E\{\alpha^{T_i}\}R_i}{1 - E\{\alpha^{T_i}\}} \geq \frac{E\{\alpha^{T_j}\}R_j}{1 - E\{\alpha^{T_j}\}}.$$

It follows that scheduling jobs in order of decreasing  $E\{\alpha^{T_i}\}R_i/(1 - E\{\alpha^{T_i}\})$  maximizes expected rewards. The structure of the optimal policy is identical with the one we derived for the preceding quiz contest example (identify  $E\{\alpha^{T_i}\}$  with the probability  $p_i$  of answering correctly question  $i$ ).

### Example 4.5.3 (Job Scheduling on Two Processors in Series)

Consider the scheduling of  $N$  jobs on two processors  $A$  and  $B$ , such that  $B$  accepts the output of  $A$  as input. Job  $i$  requires known times  $a_i$  and  $b_i$  for processing in  $A$  and  $B$ , respectively. The problem is to find a schedule that minimizes the total processing time.

To formulate the problem into the form of the basic problem, we increment discrete time at the moments when processing of a job is completed at machine  $A$  and the next job is started. We take as state at time  $k$  the collection of jobs  $X_k$  that remain to be processed at  $A$ , together with the backlog of work  $\tau_k$  at machine  $B$ , that is, the amount of time that the jobs currently at  $B$  need to clear  $B$ . Thus if  $(X_k, \tau_k)$  is the state at stage  $k$  and job  $i$  is completed at machine  $A$ , the state changes to  $(X_{k+1}, \tau_{k+1})$  given by

$$X_{k+1} = X_k - \{i\}, \quad \tau_{k+1} = b_i + \max(0, \tau_k - a_i).$$

The corresponding DP algorithm is

$$J_k(X_k, \tau_k) = \min_{i \in X_k} [a_i + J_{k+1}(X_k - \{i\}, b_i + \max(0, \tau_k - a_i))]$$

with the terminal condition

$$J_N(\emptyset, \tau_N) = \tau_N,$$

where  $\emptyset$  is the empty set.

Since the problem is deterministic, there exists an optimal open-loop schedule

$$\{i_0, \dots, i_{k-1}, i, j, i_{k+2}, \dots, i_{N-1}\}.$$

By arguing that the cost of this schedule is no worse than the cost of the schedule

$$\{i_0, \dots, i_{k-1}, j, i, i_{k+2}, \dots, i_{N-1}\},$$

obtained by interchanging  $i$  and  $j$ , it can be verified that

$$J_{k-2}(X_k - \{i\} - \{j\}, \tau_{ij}) \leq J_{k+2}(X_k - \{i\} - \{j\}, \tau_{ji}), \quad (4.52)$$

where  $\tau_{ij}$  and  $\tau_{ji}$  are the backlogs at machine  $B$  at time  $k+2$  when  $i$  is processed before  $j$  and  $j$  is processed before  $i$ , respectively, and the backlog at time  $k$  was  $\tau_k$ . A straightforward calculation shows that

$$\tau_{ij} = b_i + b_j - a_i - a_j + \max(\tau_k, a_i, a_i + a_j - b_i), \quad (4.53)$$

$$\tau_{ji} = b_j + b_i - a_j - a_i + \max(\tau_k, a_j, a_j + a_i - b_j). \quad (4.54)$$

Clearly,  $J_{k+2}$  is monotonically increasing in  $\tau$ , so from Eq. (4.52) we obtain

$$\tau_{ij} \leq \tau_{ji}.$$

In view of Eqs. (4.53) and (4.54), this relation implies two possibilities. The first is

$$\tau_k \geq \max(a_i, a_i + a_j - b_i),$$

$$\tau_k \geq \max(a_j, a_j + a_i - b_j),$$

in which case  $\tau_{ij} = \tau_{ji}$  and the order of  $i$  and  $j$  makes no difference. (This is the case where the backlog at time  $k$  is so large that both jobs  $i$  and  $j$  will find  $B$  working on an earlier job.) The second possibility is that

$$\max(a_i, a_i + a_j - b_i) \leq \max(a_j, a_j + a_i - b_j),$$

which can be seen to be equivalent to

$$\min(a_i, b_j) \leq \min(a_j, b_i).$$

A schedule satisfying these necessary conditions for optimality can be constructed by the following procedure:

1. Find  $\min_i \min(a_i, b_i)$ .
2. If the minimizing value is an  $a$  take the corresponding job first; if it is a  $b$ , take the corresponding job last.
3. Repeat the procedure with the remaining jobs until a complete schedule is constructed.

To show that this schedule is indeed optimal, we start with an optimal schedule. We consider the job  $i_0$  that minimizes  $\min(a_i, b_i)$  and by successive interchanges we move it to the same position as in the schedule constructed previously. It is seen from the preceding analysis that the resulting schedule is still optimal. Similarly, continuing through successive interchanges and maintaining optimality throughout, we can transform the optimal schedule into the schedule constructed earlier. We leave the details to the reader.

### The Interchange Argument

Let us now consider the basic problem of Chapter 1 and formalize the interchange argument used in the preceding examples. The main requirement is that the problem has structure such that *there exists an open-loop policy that is optimal*, that is, a sequence of controls that performs as well or better than any sequence of control functions. This is certainly true in deterministic problems as discussed in Chapter 1, but it is also true in some stochastic problems such as those of Examples 4.5.1 and 4.5.2.

To apply the interchange argument, we start with an optimal sequence

$$\{u_0, \dots, u_{k-1}, \bar{u}, \tilde{u}, u_{k+2}, \dots, u_{N-1}\}$$

and focus attention on the controls  $\bar{u}$  and  $\tilde{u}$  applied at times  $k$  and  $k+1$ , respectively. We then argue that if the order of  $\bar{u}$  and  $\tilde{u}$  is interchanged the expected cost cannot decrease. In particular, if  $X_k$  is the set of states that can occur with positive probability starting from the given initial state  $x_0$  and using the control subsequence  $\{u_0, \dots, u_{k-1}\}$ , we must have for all  $x_k \in X_k$

$$\begin{aligned} & E\{g_k(x_k, \bar{u}, w_k) + g_{k+1}(\bar{x}_{k+1}, \tilde{u}, w_{k+1}) + J_{k+2}^*(\bar{x}_{k+2})\} \\ & \leq E\{g_k(x_k, \tilde{u}, w_k) + g_{k+1}(\tilde{x}_{k+1}, \bar{u}, w_{k+1}) + J_{k+2}^*(\tilde{x}_{k+2})\}, \end{aligned} \quad (4.55)$$

where  $\bar{x}_{k+1}$  and  $\bar{x}_{k+2}$  (or  $\tilde{x}_{k+1}$  and  $\tilde{x}_{k+2}$ ) are the states subsequent to  $x_k$  when  $u_k = \bar{u}$  and  $u_{k+1} = \tilde{u}$  (or  $u_k = \tilde{u}$  and  $u_{k+1} = \bar{u}$ , respectively) are applied, and  $J_{k+2}^*(\cdot)$  is the optimal cost-to-go function for time  $k+2$ .

Relation (4.55) is a *necessary* condition for optimality. It holds for every  $k$  and every optimal policy that is open-loop. There is no guarantee that this necessary condition is powerful enough to lead to an optimal solution, but it is worth considering in some specially structured problems. Generally in scheduling problems, algorithms that aim to improve a sub-optimal schedule through a sequence of interchanges, may not provide an optimal solution, but are often the basis for successful heuristics.

## 4.6 SET-MEMBERSHIP DESCRIPTION OF UNCERTAINTY

In this section, we focus on problems where the uncertain quantities are described by their membership in given sets rather than probability distributions. This type of description is appropriate in minimax control problems, as discussed in Section 1.6. Our purpose in this section is to analyze some basic problems of estimation and control involving uncertainty with a set-membership description. These problems are conceptually important, and arise in several contexts, including the model predictive control methodology discussed in Section 6.5. However, their general solution can be computationally difficult. We will discuss some easily implementable approximations, involving linear systems and ellipsoidal descriptions.

### 4.6.1 Set-Membership Estimation

Suppose that we are given a linear dynamic system of the type considered in Section 4.1 but without a control ( $u_k \equiv 0$ ):

$$x_{k+1} = A_k x_k + w_k, \quad k = 0, 1, \dots, N-1,$$

where  $x_k \in \mathbb{R}^n$  and  $w_k \in \mathbb{R}^n$  denote the state and disturbance vectors, respectively, and the matrices  $A_k$  are known. Suppose also that at each time  $k$ , we receive a measurement  $z_k \in \mathbb{R}^s$  of the form

$$z_k = C_k x_k + v_k,$$

where  $v_k \in \mathbb{R}^s$  is an (unknown) observation noise vector, and the matrix  $C_k$  is given.

An important and generic problem is to estimate the value of  $x_k$ , given the observations  $z_1, \dots, z_k$ , accumulated up to time  $k$ . The uncertain quantities here are the initial state  $x_0$ , the system disturbances  $w_0, \dots, w_{N-1}$ , and the observation noise vectors  $v_1, \dots, v_N$ . When the joint probability distribution for these vectors is given, one may calculate the conditional distribution of  $x_k$  given  $z_1, \dots, z_k$ , and from this, obtain estimates such as for example the conditional expectation  $E\{x_k | z_1, \dots, z_k\}$ . This approach leads to a rich theory, centered around the Kalman filtering algorithm, which is described in detail in Appendix E.

Suppose now that, instead of a probability distribution, we have a set  $\mathcal{R}$  within which the vector of unknown quantities

$$r = (x_0, w_0, \dots, w_{N-1}, v_1, \dots, v_N)$$

is known to belong. The state  $x_k$  can be expressed in terms of  $r$  using the system equation as

$$x_k = A_{k-1} \cdots A_0 x_0 + \sum_{i=0}^{k-1} A_{k-2-i} \cdots A_{i+1} w_i,$$

or more abstractly as

$$x_k = L_k r,$$

where  $L_k$  is an appropriate matrix. Thus, knowing that  $r \in \mathcal{R}$  and before any measurements are received, the state  $x_k$  is known to belong to the set

$$\mathcal{X}_k = L_k \mathcal{R} = \{L_k r \mid r \in \mathcal{R}\}.$$

Each measurement  $z_i$ , when received, restricts the set of possible values of  $r$  to be such that  $z_i = C_i L_i r + v_i$  or

$$z_i = E_i r$$

for an appropriate matrix  $E_i$ . Thus, with each new measurement, the set of possible vectors  $r$  is further restricted, and so is the set of possible states. In particular, given the measurements  $z_1, \dots, z_k$ , the set of possible vectors  $r$  is given by

$$\mathcal{R}_k(z_1, \dots, z_k) = \mathcal{R} \cap \{r \mid z_1 = E_1 r\} \cap \dots \cap \{r \mid z_k = E_k r\}, \quad (4.56)$$

and by a linear transformation, yields the set of possible states  $x_k$  as

$$\mathcal{X}_k(z_1, \dots, z_k) = L_k \mathcal{R}_k(z_1, \dots, z_k). \quad (4.57)$$

The procedure just described is straightforward, and can easily be extended to systems and measurements that are nonlinear. The difficulty, however, is to specify conveniently the sets  $\mathcal{R}_k(z_1, \dots, z_k)$  and/or  $\mathcal{X}_k(z_1, \dots, z_k)$ . There are only few special cases where these sets admit a simple description, e.g., one that involves a finite set of numbers. The most interesting of these are:

- (a) The *polyhedral* case, where the set  $\mathcal{R}$  is a polyhedron (a set specified by a finite number of linear inequalities). Then, it can be seen that the sets  $\mathcal{R}_k(z_1, \dots, z_k)$  and/or  $\mathcal{X}_k(z_1, \dots, z_k)$  are also polyhedra. The reason is that the intersection of a polyhedron with a linear manifold (a translated subspace) is a polyhedron, and a linear transformation of a polyhedron yields another polyhedron [cf. Eqs. (4.56) and (4.57)].
- (b) The *ellipsoidal* case, where the set  $\mathcal{R}$  is an ellipsoid (a linearly transformed sphere – a more specific description is given later). Then, it can be shown that the sets  $\mathcal{R}_k(z_1, \dots, z_k)$  and/or  $\mathcal{X}_k(z_1, \dots, z_k)$  are also ellipsoids. Similar to the polyhedral case, the reason is that the intersection of an ellipsoid with a linear manifold is an ellipsoid, and a linear transformation of this ellipsoid yields another ellipsoid.

The polyhedral case is interesting in some cases, but suffers from a quick explosion of the computational requirements to describe the associated polyhedra, as  $k$  increases. We will focus instead on the ellipsoidal case, and we will use DP methods to derive easily implementable algorithms that resemble the Kalman filtering algorithm described in Appendix E.

### Energy Constraints

We first consider the most favorable case where the set of possible states  $\mathcal{X}_k(z_1, \dots, z_k)$  turns out to be an ellipsoid. Suppose that the vector of unknown quantities  $r$  is known to belong to a set of the form

$$\mathcal{R} = \left\{ r \mid (x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0) + \sum_{i=0}^{N-1} (w_i' M_i^{-1} w_i + v_{i+1}' N_{i+1}^{-1} v_{i+1}) \leq 1 \right\},$$

where  $S$ ,  $M_i$ , and  $N_i$  are positive definite symmetric matrices, and  $\hat{x}_0$  is a given vector. This set is a bounded ellipsoid. Much of the analysis that follows carries through in the case of more general ellipsoids, but for simplicity, we restrict attention to bounded ellipsoids.

Let us use DP to derive the set of possible states and show that it is an ellipsoid of the form

$$\mathcal{X}_k(z_1, \dots, z_k) = \{x_k \mid (x_k - \hat{x}_k)' \Sigma_k^{-1} (x_k - \hat{x}_k) \leq 1 - \delta_k\},$$

where

$\Sigma_k$  is a positive definite symmetric matrix that is independent of the observations  $z_1, \dots, z_k$ ,

$\hat{x}_k$  is a vector that depends on  $z_1, \dots, z_k$ ,

$\delta_k$  is a positive scalar that depends on  $z_1, \dots, z_k$ .

We observe that a vector  $\xi$  belongs to  $\mathcal{X}_k(z_1, \dots, z_k)$  if and only if there exist  $x_0$  and  $w_0, \dots, w_{k-1}$  such that

$$(x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0) + \sum_{i=0}^{k-1} w_i' M_i^{-1} w_i + \sum_{i=0}^{k-1} (z_{i+1} - C_{i+1} x_{i+1})' N_{i+1}^{-1} (z_{i+1} - C_{i+1} x_{i+1}) \leq 1$$

while  $\xi$  is equal to the vector  $x_k$  that is generated at the  $k$ th stage by the system

$$x_{i+1} = A_i x_i + w_i, \quad i = 0, \dots, k-1. \quad (4.58)$$

Thus, we have  $\xi \in \mathcal{X}_k(z_1, \dots, z_k)$  if and only if  $V_k(\xi) \leq 1$ , where  $V_k(\xi)$  is the optimal cost of the problem of minimizing the quadratic cost

$$(x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0) + \sum_{i=0}^{k-1} w_i' M_i^{-1} w_i + \sum_{i=0}^{k-1} (z_{i+1} - C_{i+1} x_{i+1})' N_{i+1}^{-1} (z_{i+1} - C_{i+1} x_{i+1})$$

subject to the system equation constraint (4.58) and the terminal condition  $x_k = \xi$  (here the  $w_i$  are viewed as the controls/minimization variables). Thus

$$\mathcal{X}_k(z_1, \dots, z_k) = \{\xi \mid V_k(\xi) \leq 1\}. \quad (4.59)$$

As the analysis of Section 4.1 suggests, the function  $V_k$  is quadratic in  $\xi$ , and can be calculated by a DP recursion. This is because in the above problem the system is linear and the cost is quadratic. Thus the set of

possible states  $\mathcal{X}_k(z_1, \dots, z_k)$  of Eq. (4.59) is an ellipsoid. To calculate the matrix and center of this ellipsoid, we can use DP. Since here the terminal state  $x_k$  is specified to be equal to the given  $\xi$ , we should use a *forward* DP algorithm and view  $V_k(\xi)$  as an *optimal cost to arrive at  $\xi$*  by optimal choice of  $x_0$  and  $w_0, \dots, w_{k-1}$  in the system (4.58). Using the reasoning employed in Section 2.1, for  $i = 1, \dots, k$ , we have the forward recursion

$$\begin{aligned} V_i(x_i) &= \min_{\substack{w_{i-1}, x_{i-1} \\ x_i = A_{i-1}x_{i-1} + w_{i-1}}} \{V_{i-1}(x_{i-1}) + w_{i-1}' M_{i-1}^{-1} w_{i-1} \\ &\quad + (z_i - C_i x_i)' N_i^{-1} (z_i - C_i x_i)\} \\ &= \min_{x_{i-1}} \{V_{i-1}(x_{i-1}) + (x_i - A_{i-1}x_{i-1})' M_{i-1}^{-1} (x_i - A_{i-1}x_{i-1}) \\ &\quad + (z_i - C_i x_i)' N_i^{-1} (z_i - C_i x_i)\} \end{aligned}$$

starting with the initial condition

$$V_0(x_0) = (x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0).$$

At the  $k$ th step of the recursion, we obtain the set of possible states  $\mathcal{X}_k(z_1, \dots, z_k)$  of Eq. (4.59).

Rather than provide the detailed derivation, we leave it for the reader to verify by induction the formula

$$V_k(x_k) = (x_k - \hat{x}_k)' \Sigma_k^{-1} (x_k - \hat{x}_k) + \delta_k,$$

where  $\hat{x}_k$  and  $\Sigma_k$  are generated by the recursions

$$\hat{x}_k = A_{k-1} \hat{x}_{k-1} + \Sigma_k C'_k N_k^{-1} (z_k - C_k A_{k-1} \hat{x}_{k-1}), \quad (4.60)$$

$$\Sigma_k = (\hat{\Sigma}_k^{-1} + C'_k N_k^{-1} C_k)^{-1}, \quad (4.61)$$

$$\hat{\Sigma}_k = A_{k-1} \Sigma_{k-1} A'_{k-1} + M_{k-1}, \quad (4.62)$$

with the initial condition

$$\Sigma_0 = S,$$

and  $\delta_k$  is given by

$$\delta_k = \sum_{i=1}^k (z_i - C_i A_{i-1} \hat{x}_{i-1})' (C_i \hat{\Sigma}_i C'_i + N_i)^{-1} (z_i - C_i A_{i-1} \hat{x}_{i-1}). \quad (4.63)$$

There are several variations of the estimation problem discussed above, for which we refer to the sources given at the end of the chapter.

### Instantaneous Constraints

We now consider a different type of set description of the uncertainty. In particular, we assume that the initial state, the system disturbances, and the observation noise vectors are independently constrained to lie in ellipsoids. In other words, we know that

$$x_0' S^{-1} x_0 \leq 1, \quad (4.64)$$

$$w_i' M_i^{-1} w_i \leq 1, \quad i = 0, \dots, N-1, \quad (4.65)$$

$$v_{i+1}' N_{i+1}^{-1} v_{i+1} \leq 1, \quad i = 0, \dots, N-1, \quad (4.66)$$

where  $S$ ,  $M_i$ , and  $N_i$  are given symmetric positive definite matrices. Thus the vector

$$r = (x_0, w_0, \dots, w_{N-1}, v_1, \dots, v_N)$$

is known to belong to the set

$$\mathcal{R} = \{r \mid (x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0) \leq 1, w_i' M_i^{-1} w_i \leq 1, v_{i+1}' N_{i+1}^{-1} v_{i+1} \leq 1, i = 0, \dots, N-1\}.$$

For this case, the set of possible states  $\mathcal{X}_k(z_1, \dots, z_k)$  is not an ellipsoid, but can be bounded by an ellipsoid, by bounding the set  $\mathcal{R}$  with an ellipsoid  $\bar{\mathcal{R}}$ , and by bounding  $\mathcal{X}_k(z_1, \dots, z_k)$  with the ellipsoid  $\bar{\mathcal{X}}_k(z_1, \dots, z_k)$  that corresponds to  $\bar{\mathcal{R}}$  as in the preceding case of energy constraints.

In particular, we observe that if  $x_0, w_0, \dots, w_{N-1}, v_1, \dots, v_N$  satisfy the instantaneous constraints of Eqs. (4.64), (4.65), (4.66), then they also satisfy the energy constraint

$$\sigma(x_0 - \hat{x}_0)' S^{-1} (x_0 - \hat{x}_0) + \sum_{i=0}^{N-1} (\mu_i w_i' M_i^{-1} w_i + \nu_{i+1} v_{i+1}' N_{i+1}^{-1} v_{i+1}) \leq 1, \quad (4.67)$$

where  $\sigma, \mu_i, \nu_{i+1}$  are any positive scalars satisfying

$$\sigma + \sum_{i=0}^{N-1} (\mu_i + \nu_{i+1}) = 1.$$

We thus replace the instantaneous constraints of Eqs. (4.64), (4.65), (4.66) with the energy constraint (4.67), and we obtain a bounding ellipsoid of the form

$$\bar{\mathcal{X}}_k(z_1, \dots, z_k) = \{x_k \mid (x_k - \hat{x}_k)' \Sigma_k^{-1} (x_k - \hat{x}_k) \leq 1 - \delta_k\},$$

where  $\hat{x}_k$  and  $\Sigma_k$  are generated by the recursions given earlier for the energy constraint case, after we replace  $S$  with  $S/\sigma$ ,  $M_i$  with  $M_i/\mu_i$ , and  $N_i$

with  $N_i/\nu_i$ . The formulas obtained in this way are simplified if we write  $\sigma, \mu_i, \nu_{i+1}$  in the following form:

$$\begin{aligned}\sigma &= (1 - \beta_0)(1 - \gamma_1)(1 - \beta_1)(1 - \gamma_2) \cdots (1 - \beta_{k-1})(1 - \gamma_k), \\ \mu_0 &= \beta_0(1 - \gamma_1)(1 - \beta_1)(1 - \gamma_2) \cdots (1 - \beta_{k-1})(1 - \gamma_k), \\ \nu_1 &= \gamma_1(1 - \beta_1)(1 - \gamma_2) \cdots (1 - \beta_{k-1})(1 - \gamma_k), \\ &\vdots \\ \mu_{k-1} &= \beta_{k-1}(1 - \gamma_k), \\ \nu_k &= \gamma_k,\end{aligned}$$

where  $\beta_{i-1}, \gamma_i, i = 1, \dots, k$  are any scalars with

$$0 < \beta_{i-1} < 1, \quad 0 < \gamma_i < 1.$$

It is easy to see that for the scalars  $\sigma, \mu_i, \nu_{i+1}$  given by the above equations, we have  $\sigma + \sum_{i=0}^{N-1} (\mu_i + \nu_{i+1}) = 1$ .

Now, by writing the estimator equations (4.60)-(4.63), with  $S, M_i$ , and  $N_i$  replaced by  $S/\sigma, M_i/\mu_i$ , and  $N_i/\nu_i$ , respectively, we can obtain after straightforward manipulation a bounding ellipsoid of the form

$$\bar{\mathcal{X}}_k(z_1, \dots, z_k) = \{x_k \mid (x_k - \hat{x}_k)' \Sigma_k^{-1} (x_k - \hat{x}_k) \leq 1 - \delta_k\},$$

where

$$\hat{x}_k = A_{k-1} \hat{x}_{k-1} + \gamma_k \Sigma_k C_k' N_k^{-1} (z_k - C_k A_{k-1} \hat{x}_{k-1}),$$

$$\Sigma_k = ((1 - \gamma_k) \hat{\Sigma}_k^{-1} + \gamma_k C_k' N_k^{-1} C_k)^{-1},$$

$$\hat{\Sigma}_k = (1 - \beta_{k-1})^{-1} A_{k-1} \Sigma_{k-1} A_{k-1}' + \beta_{k-1}^{-1} M_{k-1},$$

with the initial condition

$$\Sigma_0 = S,$$

and  $\delta_k$  is generated by the equation

$$\begin{aligned}\delta_k &= (1 - \beta_{k-1})(1 - \gamma_k) \delta_{k-1} + (z_k - C_k A_{k-1} \hat{x}_{k-1})' \\ &\quad ((1 - \gamma_k)^{-1} C_k \hat{\Sigma}_k C_k' + \gamma_k^{-1} N_k)^{-1} (z_k - C_k A_{k-1} \hat{x}_{k-1}),\end{aligned}$$

with the initial condition

$$\delta_0 = 0.$$

We omit the verification of the above equations because it is tedious, and we refer to the cited references. Note that the estimators for both cases of energy and instantaneous constraints bear close resemblance to the Kalman filtering algorithm described in Appendix E. An interesting problem variant is when the system equation has the form  $x_{k+1} = x_k$ . In this case, the problem is to use linear measurements to estimate the initial state  $x_0$ , which can be viewed as an unknown parameter vector. It can then be shown under mild assumptions that  $\Sigma_k \rightarrow 0$  as  $k \rightarrow \infty$ , so that the parameter vector is identified with arbitrary accuracy as the number of measurements increases.

#### 4.6.2 Control with Unknown-but-Bounded Disturbances

We now consider a problem of control when the uncertain quantities are described by their membership in given sets. We consider the system

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

where as usual  $x_k$  is the state,  $u_k$  is the control to be selected from a set  $U_k(x_k)$ , and  $w_k$  is a disturbance. However, instead of probability distributions, we only know that  $w_k$  belongs to a given set  $W_k(x_k, u_k)$ , which may depend on the current state  $x_k$  and control  $u_k$ .

Often in control problems one is interested in keeping the state of the system close to a desired trajectory, in spite of the effects of the disturbances. We can formulate such a problem as one of finding a policy  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$  with  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k$  and  $k$ , such that for each  $k = 1, 2, \dots, N$ , the state  $x_k$  of the closed-loop system

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k)$$

belongs to a given set  $X_k$ , called the *target set at time k*.

We may view the set sequence  $\{X_1, X_2, \dots, X_N\}$  as a "tube" within which the state must stay, even under the worst possible choice of the disturbances  $w_k$  from within the corresponding sets  $W_k(x_k, \mu_k(x_k))$ . Accordingly we refer to this problem as the problem of *reachability of a target tube*.

One may formulate this problem as a minimax control problem (cf. Section 1.6), where the cost at stage  $k$  is

$$g_k(x_k) = \begin{cases} 0 & \text{if } x_k \in X_k, \\ 1 & \text{if } x_k \notin X_k. \end{cases}$$

With this choice, the optimal cost-to-go from a given initial state  $x_0$  is the minimum number of violations of the target tube constraints  $x_k \in X_k$  that can occur when the  $w_k$  are optimally chosen, subject to the constraint  $w_k \in W_k(x_k, u_k)$ , by an adversary wishing to maximize the number of violations. In particular, if  $J_k(x_k) = 0$  for some  $x_k \in X_k$ , there exists a policy such that starting from  $x_k$ , the subsequent system states  $x_i, i = k+1, \dots, N$ , are guaranteed to be within the corresponding sets  $X_i$ .

It can be seen that the set

$$\bar{X}_k = \{x_k \mid J_k(x_k) = 0\}$$

is the set that we *must* reach at time  $k$  in order to be able to maintain the state within the subsequent target sets. Accordingly, we refer to  $\bar{X}_k$  as the *effective target set at time k*. We can generate the sets  $\bar{X}_k$  with a backwards recursion, which is derived from the DP algorithm for minimax problems

(see Section 1.6) but can also be easily justified from first principles. In particular, we start with

$$\bar{X}_N = X_N, \quad (4.68)$$

and for  $k = 0, 1, \dots, N-1$ , we have

$\bar{X}_k = \{x_k \in X_k \mid \text{there exists } u_k \in U_k(x_k) \text{ such that}$

$$f_k(x_k, u_k, w_k) \in \bar{X}_{k+1}, \text{ for all } w_k \in W_k(x_k, u_k)\}. \quad (4.69)$$

### Example 4.6.1

Consider the scalar linear system

$$x_{k+1} = 2x_k + u_k + w_k,$$

and the target tube  $\{X_1, X_2, \dots, X_N\}$ , where for all  $k$ ,

$$X_k = [-1, 1].$$

We want to keep the state within this tube by using controls  $u_k$  that belong to the set  $U_k = [-1, 1]$ , and in spite of the effects of the disturbances  $W_k$  that can take any values in the set  $[-1/2, 1/2]$ .

Let us construct the effective target sets  $\bar{X}_k$  by using the DP recursion of Eqs. (4.68) and (4.69). We have  $\bar{X}_N = [-1, 1]$ , and

$$\begin{aligned} \bar{X}_{N-1} &= \{x \mid \text{for some } u \in [-1, 1] \text{ we have} \\ &\quad -1 \leq 2x + u + w \leq 1 \text{ for all } w \in [-1/2, 1/2]\}. \end{aligned}$$

We see that for  $x$  and  $u$  to satisfy  $-1 \leq 2x + u + w \leq 1$  for all  $w \in [-1/2, 1/2]$ , it is necessary and sufficient that

$$-\frac{1}{2} \leq 2x + u \leq \frac{1}{2},$$

so  $u$  must be chosen (with knowledge of  $x$ ) to satisfy

$$-1 \leq u \leq 1, \quad -\frac{1}{2} - 2x \leq u \leq \frac{1}{2} - 2x.$$

For existence of such a  $u$ , the intervals  $[-1, 1]$  and  $[-1/2 - 2x, 1/2 - 2x]$  must have nonempty intersection, which can be seen to be true if and only if

$$-\frac{3}{4} \leq x \leq \frac{3}{4}.$$

Thus, to be able to reach the set  $X_N$  at time  $N$ , the state  $x_{N-1}$  must belong to the (effective target) set

$$\bar{X}_{N-1} = \left[ -\frac{3}{4}, \frac{3}{4} \right].$$

We similarly proceed to construct  $\bar{X}_{N-2}$ . We have

$$\bar{X}_{N-2} = \{x \mid \text{for some } u \in [-1, 1] \text{ we have}$$

$$-3/4 \leq 2x + u + w \leq 3/4 \text{ for all } w \in [-1/2, 1/2]\},$$

and we see that  $u$  must be chosen so that

$$-1 \leq u \leq 1, \quad -\frac{1}{4} - 2x \leq u \leq \frac{1}{4} - 2x.$$

For existence of such a  $u$ , the intervals  $[-1, 1]$  and  $[-1/4 - 2x, 1/4 - 2x]$  must have nonempty intersection, which can be seen to be true if and only if

$$-\frac{5}{8} \leq x \leq \frac{5}{8}.$$

Thus, to be able to reach the effective target set  $\bar{X}_{N-1}$  at time  $N-1$ , the state  $x_{N-2}$  must belong to the set

$$\bar{X}_{N-2} = \left[ -\frac{5}{8}, \frac{5}{8} \right].$$

The above calculations illustrate the form of the algorithm that yields the effective target set  $\bar{X}_k$  for every  $k$ . We have

$$\bar{X}_k = [-\alpha_k, \alpha_k],$$

where the scalars  $\alpha_k$  satisfy the recursion

$$\alpha_k = \frac{\alpha_{k+1}}{2} + \frac{1}{4}, \quad k = 0, 1, \dots, N-1,$$

with the starting condition

$$\alpha_N = 1.$$

In order to guarantee reachability of the given target tube, the initial state  $x_0$  should belong to the interval  $[-\alpha_0, \alpha_0]$ . Note that the scalars  $\alpha_k$  are monotonically decreasing as  $k \rightarrow -\infty$ , and we have  $\alpha_k \rightarrow 1/2$ . Thus, if the initial state  $x_0$  is in the interval  $[-1/2, 1/2]$ , then given any horizon length  $N$ , there is a policy that keeps the state of the system within the set  $[-1, 1]$ . In fact, it can be seen that the *linear stationary* policy  $\{\mu, \mu, \dots\}$ , where

$$\mu(x) = -2x,$$

keeps the state of the system in the interval  $[-1/2, 1/2]$ , provides the initial state belongs to that interval. It can also be seen that if the initial state does not belong to the interval  $[-1/2, 1/2]$ , then there is a large enough horizon length  $N$ , such that for every admissible policy, a sequence of feasible disturbances exists that will force the state to be outside the target set  $[-1, 1]$  at some time  $k \leq N$ . These observations can be generalized for the case of linear

systems and ellipsoidal constraint sets (see the discussion and the references given below).

In general, it is not easy to characterize the effective target sets  $\bar{X}_k$ . However, similar to the estimation problem of the preceding subsection, a few special cases involving the linear system

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

where  $A_k$  and  $B_k$  are given matrices, are amenable to exact or approximate computational solution. One such case is when the sets  $X_k$  are ellipsoids, and the sets  $U_k(x_k)$  and  $W_k(x_k, u_k)$  are also ellipsoids that do not depend on  $x_k$  and  $(x_k, u_k)$ , respectively. In this case, the effective target sets  $\bar{X}_k$  are not ellipsoids, but can be approximated by inner ellipsoids  $\tilde{X}_k \subset \bar{X}_k$  (this requires that the ellipsoids  $U_k$  have sufficiently large size, for otherwise the target tube may not be reachable and the problem may not have a solution). Furthermore, the state trajectory  $\{x_1, x_2, \dots, x_N\}$  can be maintained within the ellipsoidal tube

$$\{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_N\}$$

by using a *linear control law* (compare with the preceding example). We outline the main algorithm in Exercise 4.31, and refer to the author's thesis work, [Ber71], [BeR71b], for a more detailed analysis.

Another case of interest is when the sets  $X_k$  are polyhedral, and the sets  $U_k(x_k)$  and  $W_k(x_k, u_k)$  are also polyhedral, and independent of  $x_k$  and  $u_k$ . Then the effective target sets are polyhedral and can be computed by linear programming methods.

An important special case is when the problem is stationary and  $f_k$ ,  $X_k$ ,  $U_k$ , and  $W_k$  do not depend on  $k$ . Then it can be shown that the effective target sets satisfy

$$\bar{X}_k \subset \bar{X}_{k+1}, \quad \text{for all } k.$$

The intersection  $\cap_{k=0}^N \bar{X}_k$  depends on the size of the horizon  $N$  and "decreases" to the set

$$X_\infty = \cap_{N=1}^\infty \cap_{k=0}^N \bar{X}_k$$

as  $N$  increases to  $\infty$ . We may view  $X_\infty$  as the set within which the state can be kept for an arbitrarily large (but finite) number of time periods. Paradoxically, under some unusual circumstances, there may be states in  $X_\infty$  starting from which it may be impossible to remain within  $X_\infty$  for an indefinitely long horizon, i.e., an infinite number of time periods. Conditions that preclude this possibility have been investigated by the author in [Ber72a]. References [Ber71] and [Ber72a] contain a methodology for constructing ellipsoidal inner approximations to  $X_\infty$  and an associated linear control law for the case where the system is linear, and the sets  $X_k$ ,  $U_k$ , and  $W_k$  are ellipsoids.

#### 4.7 NOTES, SOURCES, AND EXERCISES

The certainty equivalence principle for dynamic linear-quadratic problems was first discussed by Simon [Sim56]. His work was preceded by Theil [The54], who considered a single-period case, and Holt, Modigliani, and Simon [HMS55], who considered a deterministic case. Similar problems were independently considered by Kalman and Koepcke [KaK58], Joseph and Tou [JoT61], and Gunckel and Franklin [GuF63]. The linear-quadratic problem is central in control theory; see the special issue [IEE71], which contains hundreds of references.

The literature on inventory control stimulated by the pioneering paper of Arrow et al. [AHM51] is also voluminous. An important work summarizing most of the research up to 1958 is Arrow, Karlin, and Scarf [AKS58]. Veinott [Vei66] also surveys the early work on the subject. The proof of optimality of  $(s, S)$  policies in the case of nonzero fixed costs is due to Scarf [Sca60].

Most of the material in Section 4.3 is taken from Mossin [Mos68]; see also Hakansson [Hak70], [Hak71], and Samuelson [Sam69]. Many applications of DP in economics are described in Sargent [Sar87], and Stokey and Lucas [StL89].

The material of Section 4.4 is largely drawn from White [Whi69]. Example 4.5.1 is given by Ross [Ros70], Example 4.5.2 is given by Ross [Ros83], and Example 4.5.3 is due to Weiss and Pinedo [WeP80]. An extensive reference on scheduling is Pinedo [Pin95].

The problem of state estimation with a set-membership description of the uncertainty was first formulated and addressed by Witsenhausen in his Ph.D. work at MIT [Wit66], and also in the paper [Wit68]. The material given here follows closely the author's Ph.D. thesis [Ber71], where the problem of estimation for the case of an energy constraint was first formulated and solved. The estimator given in Section 4.6.1 for the case of instantaneous constraints was also first derived in the author's thesis using the method given here, and a steady-state analysis was also given. A state estimator using ellipsoidal approximations for the case of instantaneous constraints was first proposed by Schwerpke [Sch68], [Sch74]. This estimator, however, has several drawbacks relative to the estimator given here. In particular, the associated matrix  $\Sigma_k$  depends on the observations  $z_1, \dots, z_k$ , and need not converge to a steady state as  $k \rightarrow \infty$ .

Continuous-time versions of the estimators of Section 4.6, as well and other variants of the estimation problem (the prediction and smoothing problems) with a set-membership description of the uncertainty were first given by Bertsekas and Rhodes [BeR71a]. Kurzhanski and Valyi [KuV97] provide an account of set-membership estimation. Deller [Del89] surveys applications in signal processing. Kosut, Lau, and Boyd [KLB92] discuss applications in system identification. The state estimation problem can also be addressed as a minimax problem that involves in part a probabilis-

tic description of the uncertainty. Basar [Bas91] describes the relations between this approach and the set-membership approach.

The target tube reachability problem was first formulated by the author in his Ph.D. thesis [Ber71]; see also the papers [BeR71b] and [Ber72a], and Exercises 3.23 and 3.24 of Vol. II. The associated recursion of Eqs. (4.68)-(4.69) for the effective target sets, and methods for approximating this recursion were also given in these references (see Exercise 4.31). The target tube reachability problem arises within several contexts in control system design, including model predictive control, which is described in Section 6.5 (for a recent discussion, see the paper by Mayne [May01]). For a survey of the associated issues, including extensions to continuous-time systems and additional references, see Blanchini [Bla99].

There has been considerable recent research on minimax formulations of general optimization problems under uncertainty, such as linear programming problems. This approach, which is known as *robust optimization*, is also based on a set-membership description of the uncertainty; for some representative works, see Ben-Tal and Nemirovski [BeN98], [BeN01], and Bertsimas and Sim [BeS03].

## EXERCISES

### 4.1 (Linear-Quadratic Problems with Forecasts)

Consider the linear-quadratic problem first examined in Section 4.1 ( $A_k, B_k$ : known) for the case where at the beginning of period  $k$  we have a forecast  $y_k \in \{1, 2, \dots, n\}$  consisting of an accurate prediction that  $w_k$  will be selected in accordance with a particular probability distribution  $P_{k|y_k}$  (cf. Section 1.4). The vectors  $w_k$  need not have zero mean under the distribution  $P_{k|y_k}$ . Show that the optimal control law is of the form

$$\mu_k(x_k, y_k) = -(B'_k K_{k+1} B_k + R_k)^{-1} B'_k K_{k+1} (A_k x_k + E\{w_k | y_k\}) + \alpha_k,$$

where the matrices  $K_k$  are given by the Riccati equation (4.3) and (4.4) and  $\alpha_k$  are appropriate vectors.

### 4.2

Consider a scalar linear system

$$x_{k+1} = a_k x_k + b_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

### Sec. 4.7 Notes, Sources, and Exercises

where  $a_k, b_k \in \mathbb{R}$ , and each  $w_k$  is a Gaussian random variable with zero mean and variance  $\sigma^2$ . We assume no control constraints and independent disturbances. Show that the control law  $\{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$  that minimizes the cost function

$$E \left\{ \exp \left[ x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + r u_k^2) \right] \right\}, \quad r > 0,$$

is linear in the state variable, assuming that the optimal cost is finite for every  $x_0$ . Show by example that the Gaussian assumption is essential for the result to hold. (For analyses of multidimensional versions of this exercise, see Jacobson [Jac73], Whittle [Whi82], [Whi90], and Basar [Bas00].)

### 4.3

Consider an inventory problem similar to the problem of Section 4.2 (zero fixed cost). The only difference is that at the beginning of each period  $k$  the decision maker, in addition to knowing the current inventory level  $x_k$ , receives an accurate forecast that the demand  $w_k$  will be selected in accordance with one out of two possible probability distributions  $P_L, P_S$  (large demand, small demand). The a priori probability of a large demand forecast is known (cf. Section 1.4).

- (a) Obtain the optimal ordering policy for the case of a single-period problem.
- (b) Extend the result to the  $N$ -period case.
- (c) Extend the result to the case of any finite number of possible distributions.

### 4.4

Consider the inventory problem of Section 4.2 (zero fixed cost), where the purchase costs  $c_k$ ,  $k = 0, 1, \dots, N-1$ , are not initially known, but instead they are independent random variables with a priori known probability distributions. The exact value of the cost  $c_k$ , however, becomes known to the decision maker at the beginning of the  $k$ th period, so that the inventory purchasing decision at time  $k$  is made with exact knowledge of the cost  $c_k$ . Characterize the optimal ordering policy assuming that  $p$  is greater than all possible values of  $c_k$ .

### 4.5

Consider the inventory problem of Section 4.2 for the case where the cost has the general form

$$E \left\{ \sum_{k=0}^N r_k(x_k) \right\}.$$

The functions  $r_k$  are convex and differentiable and

$$\lim_{x \rightarrow -\infty} \frac{dr_k(x)}{dx} = -\infty, \quad \lim_{x \rightarrow \infty} \frac{dr_k(x)}{dx} = \infty, \quad k = 0, \dots, N.$$

- (a) Assume that the fixed cost is zero. Write the DP algorithm for this problem and show that the optimal ordering policy has the same form as the one derived in Section 4.2.
- (b) Suppose there is a one-period time lag between the order and the delivery of inventory; that is, the system equation is of the form

$$x_{k+1} = x_k + u_{k-1} - w_k, \quad k = 0, 1, \dots, N-1,$$

where  $u_{-1}$  is given. Reformulate the problem so that it has the form of the problem of part (a). Hint: Make a change of variables  $y_k = x_k + u_{k-1}$ .

#### 4.6 (Inventory Control for Nonzero Fixed Cost)

Consider the inventory problem of Section 4.2 (nonzero fixed cost) under the assumption that unfilled demand at each stage is not backlogged but rather is lost; that is, the system equation is  $x_{k+1} = \max(0, x_k + u_k - w_k)$  instead of  $x_{k+1} = x_k + u_k - w_k$ . Complete the details of the following argument, which shows that a multiperiod  $(s, S)$  policy is optimal.

*Abbreviated Proof:* (due to S. Shreve) Let  $J_N(x) = 0$  and for all  $k$

$$G_k(y) = cy + E\left\{ h \max(0, y - w_k) + p \max(0, w_k - y) + J_{k+1}\left(\max(0, y - w_k)\right) \right\},$$

$$J_k(x) = -cx + \min_{u \geq 0} [K\delta(u) + G_k(x+u)],$$

where  $\delta(0) = 0$ ,  $\delta(u) = 1$  for  $u > 0$ . The result will follow if we can show that  $G_k$  is  $K$ -convex, continuous, and  $G_k(y) \rightarrow \infty$  as  $|y| \rightarrow \infty$ . The difficult part is to show  $K$ -convexity, since  $K$ -convexity of  $G_{k-1}$  does not imply  $K$ -convexity of  $E\{J_{k+1}(\max(0, y - w))\}$ . It will be sufficient to show that  $K$ -convexity of  $G_{k+1}$  implies  $K$ -convexity of

$$H(y) = p \max(0, -y) + J_{k+1}\left(\max(0, y)\right), \quad (4.70)$$

or equivalently that

$$K + H(y+z) \geq H(y) + z \left( \frac{H(y) - H(y-b)}{b} \right), \quad z \geq 0, b > 0, y \in \mathbb{R}. \quad (4.71)$$

By  $K$ -convexity of  $G_{k+1}$  we have for appropriate scalars  $s_{k+1}$  and  $S_{k+1}$  such that  $G_{k+1}(S_{k+1}) = \min_y G_{k+1}(y)$  and  $K + G_{k+1}(S_{k+1}) = G_{k+1}(s_{k+1})$ :

$$J_{k+1}(x) = \begin{cases} K + G_{k+1}(S_{k+1}) - cx & \text{if } x < s_{k+1}, \\ G_{k+1}(x) - cx & \text{if } x \geq s_{k+1}, \end{cases} \quad (4.72)$$

and  $J_{k+1}$  is  $K$ -convex by the theory of Section 4.2.

*Case 1:*  $0 \leq y - b < y \leq y + z$ . In this region, Eq. (4.71) follows from  $K$ -convexity of  $J_{k+1}$ .

*Case 2:*  $y - b < y \leq y + z \leq 0$ . In this region,  $H$  is linear and hence  $K$ -convex.

*Case 3:*  $y - b < y \leq 0 \leq y + z$ . In this region, Eq. (4.71) may be written [in view of Eq. (6.1)] as

$$K + J_{k+1}(y+z) \geq J_{k+1}(0) - p(y+z).$$

We will show that

$$K + J_{k+1}(z) \geq J_{k+1}(0) = pz, \quad z \geq 0. \quad (4.73)$$

If  $0 < s_{k+1} \leq z$ , then using Eq. (4.72) and the fact  $p > c$ , we have

$$K + J_{k+1}(z) = K - cz + G_{k+1}(z) \geq K - pz + G_{k+1}(S_{k+1}) = J_{k+1}(0) = pz.$$

If  $0 \leq z \leq s_{k+1}$ , then using Eq. (4.72) and the fact  $p > c$ , we have

$$K + J_{k+1}(z) = 2K - cz + G_{k+1}(S_{k+1}) \geq K - pz + G_{k+1}(S_{k+1}) = J_{k+1}(0) - pz.$$

If  $s_{k+1} \leq 0 \leq z$ , then using Eq. (4.72), the fact  $p > c$ , and part (iv) of the lemma in Section 4.2, we have

$$K + J_{k+1}(z) = K - cz + G_{k+1}(z) \geq G_{k+1}(0) - pz = J_{k-1}(0) - pz.$$

Thus Eq. (4.73) is proved and Eq. (4.71) follows for the case under consideration.

*Case 4:*  $y - b < 0 < y \leq y + z$ . Then  $0 < y < b$ . If

$$\frac{H(y) - H(0)}{y} \geq \frac{H(y) - H(y-b)}{b}, \quad (4.74)$$

then since  $H$  agrees with  $J_{k+1}$  on  $[0, \infty)$  and  $J_{k+1}$  is  $K$ -convex,

$$\begin{aligned} K + H(y+z) &\geq H(y) + z \left( \frac{H(y) - H(0)}{y} \right) \\ &\geq H(y) + z \left( \frac{H(y) - H(y-b)}{b} \right), \end{aligned}$$

where the last step follows from Eq. (4.74). If

$$\frac{H(y) - H(0)}{y} < \frac{H(y) - H(y-b)}{b},$$

then we have

$$H(y) - H(0) < \frac{y}{b} (H(y) - H(y-b)) = \frac{y}{b} (H(y) - H(0) + p(y-b)).$$

It follows that

$$\left(1 - \frac{y}{b}\right) (H(y) - H(0)) < \left(\frac{y}{b}\right) p(y-b) = -py \left(1 - \frac{y}{b}\right),$$

and since  $b > y$ ,

$$H(y) - H(0) < -py. \quad (4.75)$$

Now we have, using the definition of  $H$ , Eqs. (4.73) and (4.75),

$$\begin{aligned} H(y) + z \frac{H(y) - H(y-b)}{b} &= H(y) + z \left( \frac{H(0) - py - H(0) + p(y-b)}{b} \right) \\ &= H(y) - pz \\ &< H(0) - p(y+z) \\ &\leq K + H(y+z). \end{aligned}$$

Hence Eq. (4.73) is proved for this case as well. Q.E.D.

#### 4.7

Consider the inventory problem of Section 4.2 (zero fixed cost) with the difference that successive demands are correlated and satisfy a relation of the form

$$w_k = e_k - \gamma e_{k-1}, \quad k = 0, 1, \dots,$$

where  $\gamma$  is a given scalar,  $e_k$  are independent random variables, and  $e_{-1}$  is given.

- (a) Show that this problem can be converted into an inventory problem with independent demands. Hint: Given  $w_0, w_1, \dots, w_{k-1}$ , we can determine  $e_{k-1}$  in view of the relation

$$e_{k-1} = \gamma^k e_{-1} + \sum_{i=0}^{k-1} \gamma^i w_{k-1-i}.$$

Define  $z_k = x_k + \gamma e_{k-1}$  as a new state variable.

- (b) Show that the same is true when in addition there is a one-period delay in the delivery of inventory (cf. Exercise 4.5).

#### 4.8

Consider the inventory problem of Section 4.2 (zero fixed cost), the only difference being that there is an upper bound  $\bar{b}$  and a lower bound  $\underline{b}$  to the allowable values of the stock  $x_k$ . This imposes the additional constraint on  $u_k$

$$\underline{b} + d \leq u_k + x_k \leq \bar{b},$$

where  $d > 0$  is the maximum value that the demand  $w_k$  can take (we assume  $\underline{b} + d < \bar{b}$ ). Show that an optimal policy  $\{\mu_0^*, \dots, \mu_{N-1}^*\}$  is of the form

$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < S_k, \\ 0 & \text{if } x_k \geq S_k, \end{cases}$$

where  $S_0, S_1, \dots, S_{N-1}$  are some scalars.

#### 4.9

Consider the inventory problem of Section 4.2 (nonzero fixed cost) with the difference that demand is deterministic and must be met at each time period (i.e., the shortage cost per unit is  $\infty$ ). Show that it is optimal to order a positive amount at period  $k$  if and only if the stock  $x_k$  is insufficient to meet the demand  $w_k$ . Furthermore, when a positive amount is ordered, it should bring up stock to a level that will satisfy demand for an integral number of periods.

#### 4.10 [Vci65], [Tsi84b] [www](#)

Consider the inventory control problem of Section 4.2 (zero fixed cost) with the only difference that the orders  $u_k$  are constrained to be nonnegative integers. Let  $J_k$  be the optimal cost-to-go function. Show that:

- (a)  $J_k$  is continuous.
- (b)  $J_k(x+1) - J_k(x)$  is a nondecreasing function of  $x$ .
- (c) There exists a sequence  $\{S_k\}$  of numbers such that the policy given by

$$\mu_k(x_k) = \begin{cases} n & \text{if } S_k - n \leq x_k < S_k - n + 1, \quad n = 1, 2, \dots, \\ 0 & \text{if } x_k \geq S_k \end{cases}$$

is optimal.

#### 4.11 (Capacity Expansion Problem)

Consider a problem of expanding over  $N$  time periods the capacity of a production facility. Let us denote by  $x_k$  the production capacity at the beginning of the  $k$ th period and by  $u_k \geq 0$  the addition to capacity during the  $k$ th period. Thus capacity evolves according to

$$x_{k+1} = x_k + u_k, \quad k = 0, 1, \dots, N-1.$$

The demand at the  $k$ th period is denoted  $w_k$  and has a known probability distribution that does not depend on either  $x_k$  or  $u_k$ . Also, successive demands are assumed to be independent and bounded. We denote:

$C_k(u_k)$ : expansion cost associated with adding capacity  $u_k$ ,

$P_k(x_k + u_k - w_k)$ : penalty associated with capacity  $x_k + u_k$  and demand  $w_k$ ,

$S(x_N)$ : salvage value of final capacity  $x_N$ .

Thus the cost function has the form

$$\min_{\substack{x_k \\ u_k \\ \vdots \\ x_N}} \left\{ -S(x_N) + \sum_{k=0}^{N-1} (C_k(u_k) + P_k(x_k + u_k - w_k)) \right\}.$$

- (a) Derive the DP algorithm for this problem.  
 (b) Assume that  $S$  is a concave function with  $\lim_{x \rightarrow -\infty} dS(x)/dx = 0$ ,  $P_k$  are convex functions, and the expansion cost  $C_k$  is of the form

$$C_k(u) = \begin{cases} K + c_k u & \text{if } u > 0, \\ 0 & \text{if } u = 0, \end{cases}$$

where  $K \geq 0$ ,  $c_k > 0$  for all  $k$ . Show that the optimal policy is of the  $(s, S)$  type assuming  $c_k y + E\{P_k(y - w_k)\} \rightarrow \infty$  as  $|y| \rightarrow \infty$ .

#### 4.12

We want to use a machine to produce a certain item in quantities that meet as closely as possible a known (nonrandom) sequence of demands  $d_k$  over  $N$  periods. The machine can be in one of two states: good ( $G$ ) or bad ( $B$ ). The state of the machine is perfectly observed and evolves from one period to the next according to

$P(G | G) = \lambda_G$ ,  $P(B | G) = 1 - \lambda_G$ ,  $P(B | B) = \lambda_B$ ,  $P(G | B) = 1 - \lambda_B$ , where  $\lambda_G$  and  $\lambda_B$  are given probabilities. Let  $x_k$  be the stock at the beginning of the  $k$ th period. If the machine is in good state at period  $k$ , it can produce  $u_k$ , where  $u_k \in [0, \bar{u}]$ , and the stock evolves according to

$$x_{k+1} = x_k + u_k - d_k;$$

otherwise the stock evolves according to

$$x_{k+1} = x_k - d_k.$$

There is a cost  $g(x_k)$  for having stock  $x_k$  in period  $k$ , and the terminal cost is also  $g(x_N)$ . We assume that the cost per stage  $g$  is a convex function such that  $g(x) \rightarrow \infty$  as  $|x| \rightarrow \infty$ . The objective is to find a production policy that minimizes the total expected cost.

- (a) Prove inductively a convexity property of the cost-to-go functions, and show that for each  $k$  there is a target stock level  $S_{k+1}$  such that if the machine is in the good state, it is optimal to produce  $u_k^* \in [0, \bar{u}]$  that will bring  $x_{k+1}$  as close as possible to  $S_{k+1}$ .
- (b) Generalize part (a) for the case where each demand  $d_k$  is random and takes values in an interval  $[0, \bar{d}]$  with given probability distribution. The stock and the state of the machine are still perfectly observable.

#### 4.13 (A Gambling Problem)

A gambler enters a game whereby he may at time  $k$  stake any amount  $u_k \geq 0$  that does not exceed his current fortune  $x_k$  (defined to be his initial capital plus his gain or minus his loss thus far). He wins his stake back and as much more with probability  $p$ , where  $\frac{1}{2} < p < 1$ , and he loses his stake with probability  $(1 - p)$ . Show that the gambling strategy that maximizes  $E\{\ln x_N\}$ , where  $x_N$  denotes his fortune after  $N$  plays, is to stake at each time  $k$  an amount  $u_k = (2p - 1)x_k$ . Hint: The problem is related to the portfolio problem of Section 4.3.

#### 4.14

Consider the dynamic portfolio problem of Section 4.3 for the case where at each period  $k$  there is a forecast that the rates of return of the risky assets for that period will be selected in accordance with a particular probability distribution as in Section 1.4. Show that a partially myopic policy is optimal.

#### 4.15

Consider a problem involving the linear system

$$x_{k+1} = A_k x_k + B_k u_k, \quad k = 0, 1, \dots, N - 1,$$

where the  $n \times n$  matrices  $A_k$  are given, and the  $n \times m$  matrices  $B_k$  are random and independent with given probability distributions that do not depend on  $x_k$ ,  $u_k$ . The problem is to find a policy that maximizes  $E\{U(c' x_N)\}$ , where  $c$  is a given  $n$ -dimensional vector. We assume that  $U$  is a concave twice continuously differentiable utility function satisfying for all  $y$

$$-\frac{U'(y)}{U''(y)} = a + b y,$$

and that the control is unconstrained. Show that the optimal policy consists of linear functions of the current state. Hint: Reduce the problem to a one-dimensional problem and use the results of Section 4.3.

#### 4.16

Suppose that a person wants to sell a house and an offer comes at the beginning of each day. We assume that successive offers are independent and an offer is  $w_j$  with probability  $p_j$ ,  $j = 1, \dots, n$ , where  $w_j$  are given nonnegative scalars. Any offer not immediately accepted is not lost but may be accepted at any later date. Also, a maintenance cost  $c$  is incurred for each day that the house remains unsold. The objective is to maximize the price at which the house is sold minus the maintenance costs. Consider the problem when there is a deadline to sell the house within  $N$  days and characterize the optimal policy.

#### 4.17

Assume that we have  $x_0$  items of a certain type that we want to sell over a period of  $N$  days. At each day we may sell at most one item. At the  $k$ th day, knowing the current number  $x_k$  of remaining unsold items, we can set the selling price  $u_k$  of a unit item to a nonnegative number of our choice; then, the probability  $\lambda_k(u_k)$  of selling an item on the  $k$ th day depends on  $u_k$  as follows:

$$\lambda_k(u_k) = \alpha e^{-u_k},$$

where  $\alpha$  is a given scalar with  $0 < \alpha \leq 1$ . The objective is to find the optimal price setting policy so as to maximize the total expected revenue over  $N$  days.

- (a) Assuming that, for all  $k$ , the cost-to-go function  $J_k(x_k)$  is monotonically nondecreasing as a function of  $x_k$ , prove that for  $x_k > 0$ , the optimal prices have the form

$$\mu_k^*(x_k) = 1 + J_{k+1}(x_k) - J_{k+1}(x_k - 1),$$

and that

$$J_k(x_k) = \alpha e^{-\mu_k^*(x_k)} + J_{k+1}(x_k).$$

- (b) Prove simultaneously by induction that, for all  $k$ , the cost-to-go function  $J_k(x_k)$  is indeed monotonically nondecreasing as a function of  $x_k$ , that the optimal price  $\mu_k^*(x_k)$  is monotonically nonincreasing as a function of  $x_k$ , and that  $J_k(x_k)$  is given in closed form by

$$J_k(x_k) = \begin{cases} (N-k)\alpha e^{-1} & \text{if } x_k \geq N-k, \\ \sum_{i=k}^{N-x_k} \alpha e^{-\mu_i^*(x_k)} + x_k \alpha e^{-1} & \text{if } 0 < x_k < N-k, \\ 0 & \text{if } x_k = 0. \end{cases}$$

#### 4.18 (Optimal Termination of Sampling) [www](#)

This is a classical problem, which when appropriately paraphrased, is known as the job selection, or as the secretary selection, or as the spouse selection problem. A collection of  $N \geq 2$  objects is observed randomly and sequentially one at a time. The observer may either select the current object observed, in which case the selection process is terminated, or reject the object and proceed to observe the next. The observer can rank each object relative to those already observed, and the objective is to maximize the probability of selecting the "best" object according to some criterion. It is assumed that no two objects can be judged to be equal. Let  $r^*$  be the smallest positive integer  $r$  such that

$$\frac{1}{N-1} + \frac{1}{N-2} + \cdots + \frac{1}{r} \leq 1.$$

Show that an optimal policy requires that the first  $r^*$  objects be observed. If the  $r^*$ th object has rank 1 relative to the others already observed, it should be selected; otherwise, the observation process should be continued until an object of rank 1 relative to those already observed is found. *Hint:* We assume that, if the  $r$ th object has rank 1 relative to the previous  $(r-1)$  objects, then the probability that it is best is  $r/N$ . For  $k \geq r^*$ , let  $J_k(0)$  be the maximal probability of finding the best object assuming  $k$  objects have been selected and the  $k$ th object is not best relative to the previous  $(k-1)$  objects. Show that

$$J_k(0) = \frac{k}{N} \left( \frac{1}{N-1} + \cdots + \frac{1}{k} \right).$$

#### 4.19

A driver is looking for parking on the way to his destination. Each parking place is free with probability  $p$  independently of whether other parking places are free or not. The driver cannot observe whether a parking place is free until he reaches it. If he parks  $k$  places from his destination, he incurs a cost  $k$ . If he reaches the destination without having parked the cost is  $C$ .

- (a) Let  $F_k$  be the minimal expected cost if he is  $k$  parking places from his destination, where  $F_0 = C$ . Show that

$$F_k = p \min(k, F_{k-1}) + q F_{k-1}, \quad k = 1, 2, \dots,$$

where  $q = 1 - p$ .

- (b) Show that an optimal policy is of the form: never park if  $k \geq k^*$ , but take the first free place if  $k < k^*$ , where  $k$  is the number of parking places from the destination and  $k^*$  is the smallest integer  $i$  satisfying  $q^{i-1} < (pC+q)^{-1}$ .

#### 4.20 [Whi82]

A person may go hunting for a certain type of animal on a given day or stay home. When the animal population is  $x$ , the probability of capturing one animal is  $p(x)$ , a known increasing function, and the probability of capturing more than one is zero. A captured animal is worth one unit and a day of hunting costs  $c$  units. Assume that  $x$  does not change due to deaths or births, that the hunter knows  $x$  at all times, that the horizon is finite, and that the terminal reward is zero. Show that it is optimal to hunt only when  $p(x) \geq c$ .

#### 4.21

Consider the scalar linear system  $x_{k+1} = ax_k + bu_k$ , where  $a$  and  $b$  are known. At each period  $k$  we have the option of using a control  $u_k$  and incurring a cost  $qx_k^2 + ru_k^2$ , or else stopping and incurring a stopping cost  $tx_k^2$ . If we have not stopped by period  $N$ , the terminal cost is the stopping cost  $tx_N^2$ . We assume that  $q \geq 0$ ,  $r > 0$ ,  $t > 0$ . Show that there is a threshold value for  $t$  below which immediate stopping is optimal at every initial state, and above which continuing at every state  $x_k$  and period  $k$  is optimal.

#### 4.22

Consider a situation involving a blackmailer and his victim. In each period the blackmailer has a choice of: a) Accepting a lump sum payment of  $R$  from the victim and promising not to blackmail again. b) Demanding a payment of  $u$ , where  $u \in [0, 1]$ . If blackmailed, the victim will either: 1) Comply with the demand and pay  $u$  to the blackmailer. This happens with probability  $1-u$ . 2) Refuse to pay and denounce the blackmailer to the police. This happens with probability  $u$ . Once known to the police, the blackmailer cannot ask for any more

money. The blackmailer wants to maximize the expected amount of money he gets over  $N$  periods by optimal choice of the payment demands  $u_k$ . (Note that there is no additional penalty for being denounced to the police.) Write a DP algorithm and find the optimal policy.

#### 4.23 [Whi82]

The Greek mythological hero Theseus is trapped in King Minos' Labyrinth maze. He can try on each day one of  $N$  passages. If he enters passage  $i$  he will escape with probability  $p_i$ , he will be killed with probability  $q_i$ , and he will determine that the passage is a dead end with probability  $(1 - p_i - q_i)$ , in which case he will return to the point from which he started. Use an interchange argument to show that trying passages in order of decreasing  $p_i/q_i$  maximizes the probability of escape within  $N$  days.

#### 4.24 (Hardy's Theorem)

Let  $\{a_1, \dots, a_n\}$  and  $\{b_1, \dots, b_n\}$  be monotonically nondecreasing sequences of numbers. Let us associate with each  $i = 1, \dots, n$  a distinct index  $j_i$ , and consider the expression  $\sum_{i=1}^n a_i b_{j_i}$ . Use an interchange argument to show that this expression is maximized when  $j_i = i$  for all  $i$ , and is minimized when  $j_i = n - i + 1$  for all  $i$ .

#### 4.25

A busy professor has to complete  $N$  projects. Each project  $k$  has a deadline  $d_k$  and the time it takes the professor to complete it is  $t_k$ . The professor can work on only one project at a time and must complete it before moving on to a new project. For a given order of completion of the projects, denote by  $c_k$  the time of completion of project  $k$ , i.e.,

$$c_k = t_k + \sum_{\substack{\text{projects } i \\ \text{completed before } k}} t_i.$$

The professor wants to order the projects so as to minimize the maximum tardiness, given by  $\max_{k \in \{1, \dots, N\}} \max(0, c_k - d_k)$ . Use an interchange argument to show that it is optimal to complete the projects in the order of their deadlines (do the project with the closest deadline first).

#### 4.26

Assume that we have two gold mines, Anaconda and Bonanza, and a gold-mining machine. Let  $x_A$  and  $x_B$  be the current amounts of gold in Anaconda and Bonanza, respectively ( $x_A$  and  $x_B$  are integer). When the machine is used in Anaconda or Bonanza, there is a probability  $p$  that  $[r_A x_A]$  (or  $[r_B x_B]$ , respectively)

of the gold will be mined without damaging the machine, and a probability  $1 - p$  that the machine will be damaged beyond repair and no gold will be mined. We assume that  $0 < r_A < 1$  and  $0 < r_B < 1$ . We want to find a policy that selects the mine in which to use the machine at each period so as to maximize the total expected amount of gold mined.

- (a) Use an interchange argument to show that it is optimal to mine Anaconda if and only if  $r_A x_A \geq r_B x_B$ .
- (b) Solve the problem for  $x_A = 2$ ,  $x_B = 4$ ,  $r_A = 0.4$ ,  $r_B = 0.6$ ,  $p = 0.9$ .

#### 4.27

Consider the quiz contest problem of Example 5.1, where is an order constraint that each question  $i$  may be answered only after a given number  $k_i$  of other questions have been answered. Use an interchange argument to show that an optimal list can be constructed by ordering the questions in decreasing order of  $p_i R_i / (1 - p_i)$  and by sequentially answering the top question in the list out of those that are available (have not yet been answered and satisfy the order constraints).

#### 4.28

Consider the quiz contest problem of Example 5.1, where there is a cost  $F_i \geq 0$  for failing to answer question  $i$  correctly (in addition to losing the reward  $R_i$ ).

- (a) Use an interchange argument to show that it is optimal to answer the questions in order of decreasing  $(p_i R_i - (1 - p_i) F_i) / (1 - p_i)$ .
- (b) Solve the variant of the problem where there is an option to stop answering questions.

#### 4.29

Consider the quiz contest problem of Example 5.1, where there is a maximum number of questions that can be answered, which is smaller than the number of questions that are available.

- (a) Show that it is not necessarily optimal to answer the questions in order of decreasing  $p_i R_i / (1 - p_i)$ . Hint: Try the case where only one out of two available questions can be answered.
- (b) Give a simple algorithm to solve the problem where the number of available questions is one more than the maximum number of questions that can be answered.

#### 4.30 (Reachability of One-Dimensional Linear Systems)

Generalize the analysis of Example 4.6.1 for the case of the one-dimensional linear system

$$x_{k+1} = ax_k + bu_k + w_k, \quad k = 0, \dots, N-1,$$

where  $x_k$  should be kept within an interval  $[-\alpha, \alpha]$ , using controls from an interval  $[\beta, \bar{\beta}]$ , and in spite of the effects of the disturbances that can take values from the interval  $[-\gamma, \gamma]$ . Derive an algorithm to generate the effective target sets, and characterize the set of initial states from which reachability of the target tube is guaranteed. What happens to this set as  $N \rightarrow \infty$ ?

### 4.31 (Reachability of Ellipsoidal Tubes [BeR71b], [Ber72a]) [www](#)

Consider the linear system

$$x_{k+1} = A_k x_k + B_k u_k + w_k,$$

where the controls  $u_k$  and the disturbances  $w_k$  must belong to the ellipsoids

$$U_k = \{x \mid u' R_k u \leq 1\}, \quad W_k = \{x \mid w' D_k w \leq 1\},$$

where  $R_k$  and  $D_k$  are given positive definite symmetric matrices.

(a) Focus on a single period  $k$ , and consider the problem of finding an ellipsoid

$$\bar{X} = \{x \mid x' K x \leq 1\},$$

where  $K$  is a positive definite symmetric matrix, such that  $\bar{X}$  is contained in the intersection of the following two sets: (1) an ellipsoid  $\{x \mid x' \Xi x \leq 1\}$ , where  $\Xi$  is a positive definite symmetric matrix, and (2) the set of all states  $x$  such that there exists a  $u \in U_k$  with the property that for all  $w \in W_k$ , we have  $A_k x + B_k u + w \in X$ , where

$$X = \{x \mid x' \Psi x \leq 1\},$$

and  $\Psi$  is a given positive definite symmetric matrix. Show that if for some scalar  $\beta \in (0, 1)$ , the matrix

$$F^{-1} = (1 - \beta)(\Psi^{-1} - \beta^{-1} D_k^{-1})$$

is well-defined as a positive definite matrix, an appropriate matrix  $K$  is given by

$$K = A'_k (F^{-1} + B_k R_k^{-1} B'_k)^{-1} A_k + \Xi.$$

Furthermore, the linear control law

$$\mu(x) = -(R_k + B'_k F B_k)^{-1} B'_k F A_k x$$

satisfies the constraint  $\mu(x) \in U_k$  for all  $x \in \bar{X}$  and achieves reachability of  $X$  if  $x \in \bar{X}$ , i.e.,  $\mu$  is such that  $A_k x + B_k \mu(x) + w \in X$  for all  $x \in \bar{X}$  and  $w \in W_k$ . Hint: Use the fact that the vector sum of two ellipsoids  $\{x \mid x' E_1 x \leq 1\}$  and  $\{x \mid x' E_2 x \leq 1\}$  (with  $E_1$  and  $E_2$  positive definite symmetric) is contained in the ellipsoid  $\{x \mid x' E x \leq 1\}$ , where

$$E^{-1} = \beta^{-1} E_1^{-1} + (1 - \beta)^{-1} E_2^{-1}$$

and  $\beta$  is any scalar with  $0 < \beta < 1$ .

(b) Consider an ellipsoidal target tube  $\{\hat{X}_0, \hat{X}_1, \dots, \hat{X}_N\}$ , where

$$\hat{X}_k = \{x \mid x' \Xi_k x \leq 1\}$$

and the  $\Xi_k$  are given positive definite symmetric matrices. Let the matrix sequences  $\{F_k\}$  and  $\{K_k\}$  be generated by the algorithm

$$K_N = \Xi_N,$$

$$F_{k+1}^{-1} = (1 - \beta_k)(K_{k+1}^{-1} - \beta_k^{-1} D_k^{-1}), \quad k = 0, 1, \dots, N-1,$$

$$K_k = A'_k (F_{k+1}^{-1} + B_k R_k^{-1} B'_k)^{-1} A_k - \Xi_k, \quad k = 0, 1, \dots, N-1,$$

where  $\beta_k$  are scalars satisfying  $0 < \beta_k < 1$ . Use the procedure of part (a) to show that a linear control law of the form

$$\mu_k(x_k) = -(R_k + B'_k F_{k+1} B_k)^{-1} B'_k F_{k+1} A_k x_k, \quad k = 0, 1, \dots, N-1,$$

achieves reachability of the target tube, provided the matrices  $F_k$  are well-defined as positive definite matrices and  $x_0$  satisfies  $x_0' K_0^{-1} x_0 \leq 1$ .

(c) Suppose that the matrices  $A_k$ ,  $B_k$ ,  $R_k$ ,  $D_k$ , and  $\Xi_k$  do not depend on  $k$ , and that the algebraic matrix equation

$$K = A' ((1 - \beta)(K^{-1} - \beta^{-1} D^{-1}) + B R^{-1} B')^{-1} A + \Xi$$

has a positive definite solution  $\bar{K}$  for some  $\beta \in (0, 1)$  for which the matrix

$$F^{-1} = (1 - \beta)(\bar{K}^{-1} - \beta^{-1} D^{-1})$$

is well defined as a positive definite matrix. Show that if the initial state belongs to the set  $\bar{X} = \{x \mid x' \bar{K} x \leq 1\}$ , then all subsequent states will belong to  $\bar{X}$  when the stationary linear control law

$$\mu(x) = -(R + B' F B)^{-1} B' F A x$$

is used.

### 4.32 (Pursuit-Evasion Games and Reachability [BeR71b])

Consider the linear system

$$x_{k+1} = A_k x_k + B_k u_k + C_k v_k, \quad k = 0, 1, \dots, N-1$$

where the controls  $u_k$  and  $v_k$  are selected by two antagonistic players from sets  $U_k$  and  $V_k$ , respectively, with exact knowledge of  $x_k$  (but without knowledge of the other player's choice at time  $k$ ). The player selecting  $u_k$  aims to bring the state of the system within some given set  $X$  at some time  $k = 1, \dots, N$ , while the player selecting  $v_k$  aims to keep the state of the system outside the set  $X$  at all times  $k = 1, \dots, N$ . Relate this problem to the problem of reachability of a target tube, and characterize the sets of initial conditions  $x_0$  starting from which the two players are guaranteed to achieve their objective with suitable choice of their control laws.

## 4.33

A famous but somewhat vain opera singer is scheduled to sing on  $N$  successive nights. If she is satisfied with her performance on a given night  $k$  (which happens with probability  $p$ , independently of the previous history) she will sing on the following night (i.e., night  $k + 1$ ). If she is not satisfied, however, she sulks and declares that she will not sing further. In this case, the only way to placate her into performing on the following night is for the opera director to send her an expensive gift, costing  $G$  dollars, which successfully placates her with probability  $q$  (independently of the previous history). If the gift does not placate her, the missed performance costs the opera house  $C$  dollars. The opera director may send a gift on any night, regardless of the success he has had with gifts on previous nights. The objective is to find a policy for when to send a gift and when not to, that minimizes the total cost from the  $N$  nights.

- (a) Write a DP algorithm for solving the problem, and characterize as best as you can the optimal policy.
- (b) Repeat part (a) for the case where the probability  $q$  is not constant, but rather is a decreasing function of the current stage.

## 4.34

An enterprising but somewhat foolish graduate student has invested the tuition for next semester in the stock market. As a result, he currently possesses a certain amount of stock that he/she must sell by registration day, which is  $N$  days away. The stock must be sold in its entirety on a single day, and will then be deposited in a bank where it will earn interest at a daily rate  $r$ . The value of the stock on day  $k$  is denoted by  $x_k$  and it evolves according to

$$x_{k+1} = \lambda x_k + w_k, \quad x_0 : \text{given},$$

where  $\lambda$  is a scalar with  $0 < \lambda < 1$ , and  $w_k$  is a random variable taking one of a finite number of positive values. We assume that  $w_0, \dots, w_{N-1}$  are independent and identically distributed. The student wants to maximize the expected value of the money he/she has on registration day.

- (a) Write a DP algorithm for solving the problem, and characterize as best as you can the optimal policy.
- (b) Assume that the student has the option of selling only a portion of his stock on a given day. What if anything would he/she do different?

## 5

## *Problems with Imperfect State Information*

### Contents

5.1. Reduction to the Perfect Information Case . . . . .	p. 218
5.2. Linear Systems and Quadratic Cost . . . . .	p. 229
5.3. Minimum Variance Control of Linear Systems . . . . .	p. 236
5.4. Sufficient Statistics . . . . .	p. 251
5.4.1. The Conditional State Distribution . . . . .	p. 252
5.4.2. Finite-State Systems . . . . .	p. 258
5.5. Notes, Sources, and Exercises . . . . .	p. 270

We have assumed so far that the controller has access to the exact value of the current state, but this assumption is often unrealistic. For example, some state variables may be inaccessible, the sensors used for measuring them may be inaccurate, or the cost of obtaining the exact value of the state may be prohibitive. We model situations of this type by assuming that at each stage the controller receives some observations about the value of the current state, which may be corrupted by stochastic uncertainty.

Problems where the controller uses observations of this type in place of the state are called problems of *imperfect state information*, and are the subject of this chapter. We will find that even though there are DP algorithms for imperfect information problems, these algorithms are far more computationally intensive than in the perfect information case. For this reason, in the absence of an analytical solution, imperfect information problems are typically solved suboptimally in practice. On the other hand, we will see that conceptually, imperfect state information problems are no different than the perfect state information problems we have been studying so far. In fact by various reformulations, we can reduce an imperfect state information problem to one with perfect state information. We will study two different reductions of this type, which will yield two different DP algorithms. The first reduction is the subject of the next section, while the second reduction will be given in Section 5.4.

## 5.1 REDUCTION TO THE PERFECT INFORMATION CASE

We first formulate the imperfect state information counterpart of the basic problem.

### Basic Problem with Imperfect State Information

Consider the basic problem of Section 1.2 where the controller, instead of having perfect knowledge of the state, has access to observations  $z_k$  of the form

$$z_0 = h_0(x_0, v_0), \quad z_k = h_k(x_k, u_{k-1}, v_k), \quad k = 1, 2, \dots, N - 1.$$

The observation  $z_k$  belongs to a given observation space  $Z_k$ . The random observation disturbance  $v_k$  belongs to a given space  $V_k$  and is characterized by a given probability distribution

$$P_{v_k}(\cdot | x_k, \dots, x_0, u_{k-1}, \dots, u_0, w_{k-1}, \dots, w_0, v_{k-1}, \dots, v_0),$$

which depends on the current state and the past states, controls, and disturbances.

The initial state  $x_0$  is also random and characterized by a given probability distribution  $P_{x_0}$ . The probability distribution  $P_{w_k}(\cdot | x_k, u_k)$  of  $w_k$  is given, and it may depend explicitly on  $x_k$  and  $u_k$  but not on the prior disturbances  $w_0, \dots, w_{k-1}, v_0, \dots, v_{k-1}$ . The control  $u_k$  is constrained to take values from a given nonempty subset  $U_k$  of the control space  $C_k$ . It is assumed that this subset does not depend on  $x_k$ .

Let us denote by  $I_k$  the information available to the controller at time  $k$  and call it the *information vector*. We have

$$\begin{aligned} I_k &= (z_0, z_1, \dots, z_k, u_0, u_1, \dots, u_{k-1}), \quad k = 1, 2, \dots, N - 1, \\ I_0 &= z_0. \end{aligned} \quad (5.1)$$

We consider the class of policies consisting of a sequence of functions  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , where each function  $\mu_k$  maps the information vector  $I_k$  into the control space  $C_k$  and

$$\mu_k(I_k) \in U_k, \quad \text{for all } I_k, \quad k = 0, 1, \dots, N - 1.$$

Such policies are called *admissible*. We want to find an admissible policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$  that minimizes the cost function

$$J_\pi = \underset{\substack{x_0, w_k, v_k \\ k=0, \dots, N-1}}{E} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(I_k), w_k) \right\}$$

subject to the system equation

$$x_{k+1} = f_k(x_k, \mu_k(I_k), w_k), \quad k = 0, 1, \dots, N - 1,$$

and the measurement equation

$$\begin{aligned} z_0 &= h_0(x_0, v_0), \\ z_k &= h_k(x_k, \mu_{k-1}(I_{k-1}), v_k), \quad k = 1, 2, \dots, N - 1. \end{aligned}$$

Note the difference from the perfect state information case. Whereas before we were trying to find a rule that would specify the control  $u_k$  to be applied for each state  $x_k$  and time  $k$ , now we are looking for a rule that gives the control to be applied for every possible information vector  $I_k$  (or state of information), that is, for every sequence of observations received and controls employed up to time  $k$ .

### Example 5.1.1 (Multiaccess Communication)

Consider a collection of transmitting stations sharing a common channel, for example, a set of ground stations communicating with a satellite at a common frequency. The stations are synchronized to transmit packets of

data at integer times. Each packet requires one time unit (also called a *slot*) for transmission. The total number  $a_k$  of packet arrivals during slot  $k$  is independent of prior arrivals and has a given probability distribution. The stations do not know the backlog  $x_k$  at the beginning of the  $k$ th slot (the number of packets waiting to be transmitted). Packet transmissions are scheduled using a strategy (known as *slotted Aloha*) whereby each packet residing in the system at the beginning of the  $k$ th slot is transmitted during the  $k$ th slot with probability  $u_k$  (common for all packets). If two or more packets are transmitted simultaneously, they collide and have to rejoin the backlog for retransmission at a later slot. However, the stations can observe the channel and determine whether in any one slot there was a collision (two or more packets), a success (one packet), or an idle (no packets). These observations provide information about the state of the system (the backlog  $x_k$ ) and can be used to select appropriately the control (the transmission probability  $u_k$ ). The objective is to keep the backlog small, so we assume a cost per stage  $g_k(x_k)$ , which is a monotonically increasing function of  $x_k$ .

The state of the system here is the backlog  $x_k$  and evolves according to the equation

$$x_{k+1} = x_k + a_k - t_k,$$

where  $a_k$  is the number of new arrivals and  $t_k$  is the number of packets successfully transmitted during slot  $k$ . Both  $a_k$  and  $t_k$  may be viewed as disturbances, and the distribution of  $t_k$  depends on the state  $x_k$  and the control  $u_k$ . It can be seen that  $t_k = 1$  (a success) with probability  $x_k u_k (1 - u_k)^{x_k - 1}$ , and  $t_k = 0$  (idle or collision) otherwise [the probability of any one of the  $x_k$  waiting packets being transmitted, while all the other packets are not transmitted, is  $u_k (1 - u_k)^{x_k - 1}$ ].

If we had perfect state information (i.e., the backlog  $x_k$  were known at the beginning of slot  $k$ ), the optimal policy would be to select the value of  $u_k$  that maximizes  $x_k u_k (1 - u_k)^{x_k - 1}$ , which is the success probability.† By setting the derivative of this probability to zero, we find the optimal (perfect state information) policy to be

$$\mu_k(x_k) = \frac{1}{x_k}, \quad \text{for all } x_k \geq 1.$$

† For a more detailed derivation, note that the DP algorithm for the perfect state information problem is

$$\begin{aligned} J_k(x_k) = g_k(x_k) + \min_{\substack{0 \leq u_k \leq 1 \\ a_k}} E & \left\{ p(x_k, u_k) J_{k+1}(x_k + a_k - 1) \right. \\ & \left. + (1 - p(x_k, u_k)) J_{k+1}(x_k + a_k) \right\}, \end{aligned}$$

where  $p(x_k, u_k)$  is the success probability  $x_k u_k (1 - u_k)^{x_k - 1}$ . Since the cost per stage  $g_k(x_k)$  is an increasing function of the backlog  $x_k$ , it is clear that each cost-to-go function  $J_k(x_k)$  is an increasing function of  $x_k$  (this can also be proved by induction). Thus  $J_{k+1}(x_k + a_k) \geq J_{k+1}(x_k + a_k - 1)$  for all  $x_k$  and  $a_k$ , based on which the DP algorithm implies that the optimal  $u_k$  maximizes  $p(x_k, u_k)$  over  $[0, 1]$ .

In practice, however,  $x_k$  is not known (imperfect state information), and the optimal control must be chosen on the basis of the available observations (i.e., the entire channel history of successes, idles, and collisions). These observations relate to the backlog history (the past states) and the past transmission probabilities (the past controls), but are corrupted by stochastic uncertainty. Mathematically, we may write an equation  $z_{k+1} = v_{k-1}$ , where  $z_{k+1}$  is the observation obtained at the end of the  $k$ th slot, and the random variable  $v_{k-1}$  yields an idle with probability  $(1 - u_k)^{x_k}$ , a success with probability  $x_k u_k (1 - u_k)^{x_k - 1}$ , and a collision otherwise.

It can be seen that this is a problem that fits the given imperfect state information framework. Unfortunately, the optimal solution to this problem is very complicated and for all practical purposes cannot be computed. A suboptimal solution will be discussed in Section 6.1.

### Reformulation as a Perfect State Information Problem

We now show how to effect the reduction from imperfect to perfect state information. As in the discussion of state augmentation in Section 1.4, it is intuitively clear that we should define a new system whose state at time  $k$  is the set of all variables the knowledge of which can be of benefit to the controller when making the  $k$ th decision. Thus a first candidate as the state of the new system is the information vector  $I_k$ . Indeed we will show that this choice is appropriate.

We have by the definition of the information vector [cf. Eq. (5.1)]

$$I_{k+1} = (I_k, z_{k+1}, u_k), \quad k = 0, 1, \dots, N-2, \quad I_0 = z_0. \quad (5.2)$$

These equations can be viewed as describing the evolution of a system of the same nature as the one considered in the basic problem of Section 1.2. The state of the system is  $I_k$ , the control is  $u_k$ , and  $z_{k+1}$  can be viewed as a random disturbance. Furthermore, we have

$$P(z_{k+1} | I_k, u_k) = P(z_{k+1} | I_k, u_k, z_0, z_1, \dots, z_k),$$

since  $z_0, z_1, \dots, z_k$  are part of the information vector  $I_k$ . Thus the probability distribution of  $z_{k+1}$  depends explicitly only on the state  $I_k$  and control  $u_k$  of the new system (5.2) and not on the prior "disturbances"  $z_0, \dots, z_0$ .

By writing

$$E\{g_k(x_k, u_k, w_k)\} = E \left\{ \underset{x_k, w_k}{E} \{g_k(x_k, u_k, w_k) | I_k, u_k\} \right\},$$

we can similarly reformulate the cost function in terms of the variables of the new system. The cost per stage as a function of the new state  $I_k$  and the control  $u_k$  is

$$\tilde{g}_k(I_k, u_k) = E_{x_k, w_k} \{g_k(x_k, u_k, w_k) | I_k, u_k\}. \quad (5.3)$$

Thus the basic problem with imperfect state information has been reformulated as a problem with perfect state information that involves the system (5.2) and the cost per stage (5.3). By writing the DP algorithm for this latter problem and substituting the expressions (5.2) and (5.3), we obtain

$$J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} \left[ E_{x_{N-1}, w_{N-1}} \left\{ g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \mid I_{N-1}, u_{N-1} \right\} \right], \quad (5.4)$$

and for  $k = 0, 1, \dots, N-2$ ,

$$J_k(I_k) = \min_{u_k \in U_k} \left[ E_{x_k, w_k, z_{k+1}} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(I_k, z_{k+1}, u_k) \mid I_k, u_k \right\} \right]. \quad (5.5)$$

These equations constitute one possible DP algorithm for the imperfect state information problem. An optimal policy  $\{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$  is obtained by first minimizing in the right-hand side of the DP Eq. (5.4) for every possible value of the information vector  $I_{N-1}$  to obtain  $\mu_{N-1}^*(I_{N-1})$ . Simultaneously,  $J_{N-1}(I_{N-1})$  is computed and used in the computation of  $J_{N-2}(I_{N-2})$  via the minimization in the DP Eq. (5.5), which is carried out for every possible value of  $I_{N-2}$ . Proceeding similarly,  $J_{N-3}(I_{N-3})$  and  $\mu_{N-3}^*$  and so on are obtained, until  $J_0(I_0) = J_0(z_0)$  is computed. The optimal cost  $J^*$  is then given by

$$J^* = E_{z_0} \{ J_0(z_0) \}.$$

### Machine Repair Example

A machine can be in one of two states denoted  $P$  and  $\bar{P}$ . State  $P$  corresponds to a machine in proper condition (good state) and state  $\bar{P}$  to a machine in improper condition (bad state). If the machine is operated for one time period, it stays in state  $P$  with probability  $\frac{2}{3}$  if it started in  $P$ , and it stays in state  $\bar{P}$  with probability 1 if it started in  $\bar{P}$ . The machine is operated for a total of three time periods and starts in state  $P$ . At the end of the first and second time periods the machine is inspected and there are two possible inspection outcomes denoted  $G$  (probably good state) and  $B$  (probably bad state). If the machine is in the good state  $P$ , the inspection outcome is  $G$  with probability  $\frac{3}{4}$ ; if the machine is in the bad state  $\bar{P}$ , the inspection outcome is  $B$  with probability  $\frac{3}{4}$ :

$$P(G \mid x = P) = \frac{3}{4}, \quad P(B \mid x = P) = \frac{1}{4},$$

$$P(G \mid x = \bar{P}) = \frac{1}{4}, \quad P(B \mid x = \bar{P}) = \frac{3}{4};$$

see Fig. 5.1.1. After each inspection one of two possible actions can be taken:

$C$ : Continue operation of the machine.

$S$ : Stop the machine, determine its state through an accurate diagnostic test, and if it is in the bad state  $\bar{P}$  bring it back to the good state  $P$ .

At each period there is a cost of 2 and 0 units for starting the period with a machine in the bad state  $\bar{P}$  and the good state  $P$ , respectively. The cost for taking the stop-and-repair action  $S$  is 1 unit and the terminal cost is 0.

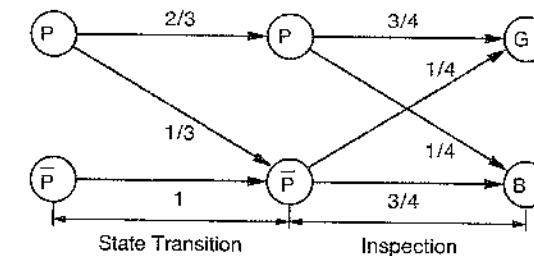


Figure 5.1.1 State transition diagram and probabilities of inspection outcomes in the machine repair example.

The problem is to determine the policy that minimizes the expected costs over the three time periods. In other words, we want to find the optimal action after the result of the first inspection is known, and after the results of the first and second inspections, as well as the action taken after the first inspection, are known.

It can be seen that this example falls within the general framework of the problem of this section. The state space consists of the two states  $P$  and  $\bar{P}$ ,

$$\text{state space} = \{P, \bar{P}\},$$

and the control space consists of the two actions

$$\text{control space} = \{C, S\}.$$

The system evolution may be described by introducing a system equation

$$x_{k+1} = w_k, \quad k = 0, 1,$$

where for  $k = 0, 1$ , the probability distribution of  $w_k$  is given by

$$P(w_k = P \mid x_k = P, u_k = C) = \frac{2}{3}, \quad P(w_k = \bar{P} \mid x_k = P, u_k = C) = \frac{1}{3},$$

$$\begin{aligned} P(w_k = P \mid x_k = P, u_k = C) &= 0, & P(w_k = \bar{P} \mid x_k = P, u_k = C) &= 1, \\ P(w_k = P \mid x_k = P, u_k = S) &= \frac{2}{3}, & P(w_k = \bar{P} \mid x_k = P, u_k = S) &= \frac{1}{3}, \\ P(w_k = P \mid x_k = \bar{P}, u_k = S) &= \frac{2}{3}, & P(w_k = \bar{P} \mid x_k = \bar{P}, u_k = S) &= \frac{1}{3}. \end{aligned}$$

We denote by  $x_0, x_1, x_2$  the state of the machine at the end of the first, second, and third time period, respectively. Also we denote by  $u_0$  the action taken after the first inspection (end of first time period) and by  $u_1$  the action taken after the second inspection (end of second time period). The probability distribution of  $x_0$  is

$$P(x_0 = P) = \frac{2}{3}, \quad P(x_0 = \bar{P}) = \frac{1}{3}.$$

Note that we do not have perfect state information, since the inspections do not reveal the state of the machine with certainty. Rather the result of each inspection may be viewed as a measurement of the system state of the form

$$z_k = v_k, \quad k = 0, 1,$$

where for  $k = 0, 1$ , the probability distribution of  $v_k$  is given by

$$\begin{aligned} P(v_k = G \mid x_k = P) &= \frac{3}{4}, & P(v_k = B \mid x_k = P) &= \frac{1}{4}, \\ P(v_k = G \mid x_k = \bar{P}) &= \frac{1}{4}, & P(v_k = B \mid x_k = \bar{P}) &= \frac{3}{4}. \end{aligned}$$

The cost resulting from a sequence of states  $x_0, x_1$  and actions  $u_0, u_1$  is

$$g(x_0, u_0) + g(x_1, u_1),$$

where

$$g(P, C) = 0, \quad g(P, S) = 1, \quad g(\bar{P}, C) = 2, \quad g(\bar{P}, S) = 1.$$

The information vector at times 0 and 1 is

$$I_0 = z_0, \quad I_1 = (z_0, z_1, u_0),$$

and we seek functions  $\mu_0(I_0), \mu_1(I_1)$  that minimize

$$\begin{aligned} &\underset{x_0, w_0, w_1}{E} \left\{ g(x_0, \mu_0(I_0)) + g(x_1, \mu_1(I_1)) \right\} \\ &= \underset{x_0, w_0, w_1}{E} \left\{ g(x_0, \mu_0(z_0)) + g(x_1, \mu_1(z_0, z_1, \mu_0(z_0))) \right\}. \end{aligned}$$

We now apply the DP algorithm. It involves taking the minimum over the two possible actions, C and S, and it has the form

$$\begin{aligned} J_k(I_k) &= \min \left[ P(x_k = P \mid I_k, C)g(P, C) + P(x_k = \bar{P} \mid I_k, C)g(\bar{P}, C) \right. \\ &\quad + \left. E_{z_{k+1}} \{J_{k+1}(I_k, C, z_{k+1}) \mid I_k, C\}, \right. \\ &\quad P(x_k = P \mid I_k, S)g(P, S) + P(x_k = \bar{P} \mid I_k, S)g(\bar{P}, S) \\ &\quad \left. + E_{z_{k+1}} \{J_{k+1}(I_k, S, z_{k+1}) \mid I_k, S\} \right], \end{aligned}$$

where  $k = 0, 1$ , and the terminal condition is  $J_2(I_2) = 0$ .

*Last Stage:* We use Eq. (5.4) to compute  $J_1(I_1)$  for each of the eight possible information vectors  $I_1 = (z_0, z_1, u_0)$ . As indicated by the above DP algorithm, for each of these vectors, we shall compute the expected cost of the possible actions,  $u_1 = C$  and  $u_1 = S$ , and select as optimal the action with the smallest cost. We have

$$\text{cost of } C = 2 \cdot P(x_1 = \bar{P} \mid I_1), \quad \text{cost of } S = 1,$$

and therefore

$$J_1(I_1) = \min [2P(x_1 = \bar{P} \mid I_1), 1].$$

The probabilities  $P(x_1 = \bar{P} \mid I_1)$  can be computed by using Bayes' rule and the problem data. Some of the details will be omitted. We have:

(1) For  $I_1 = (G, G, S)$

$$\begin{aligned} P(x_1 = \bar{P} \mid G, G, S) &= \frac{P(x_1 = \bar{P}, G, G \mid S)}{P(G, G \mid S)} \\ &= \frac{\frac{1}{3} \cdot \frac{1}{4} \cdot (\frac{2}{3} \cdot \frac{3}{4} + \frac{1}{3} \cdot \frac{1}{4})}{(\frac{2}{3} \cdot \frac{3}{4} + \frac{1}{3} \cdot \frac{1}{4})^2} = \frac{1}{7}. \end{aligned}$$

Hence

$$J_1(G, G, S) = \frac{2}{7}, \quad \mu_1^*(G, G, S) = C.$$

(2) For  $I_1 = (B, G, S)$

$$P(x_1 = \bar{P} \mid B, G, S) = P(x_1 = \bar{P} \mid G, G, S) = \frac{1}{7},$$

$$J_1(B, G, S) = \frac{2}{7}, \quad \mu_1^*(B, G, S) = C.$$

(3) For  $I_1 = (G, B, S)$

$$\begin{aligned} P(x_1 = \bar{P} \mid G, B, S) &= \frac{P(x_1 = \bar{P}, G, B \mid S)}{P(G, B \mid S)} \\ &= \frac{\frac{1}{3} \cdot \frac{3}{4} \cdot \left(\frac{2}{3} \cdot \frac{3}{4} + \frac{1}{3} \cdot \frac{1}{4}\right)}{\left(\frac{2}{3} \cdot \frac{1}{4} + \frac{1}{3} \cdot \frac{3}{4}\right) \left(\frac{2}{3} \cdot \frac{3}{4} + \frac{1}{3} \cdot \frac{1}{4}\right)} = \frac{3}{5}, \end{aligned}$$

$$J_1(G, B, S) = 1, \quad \mu_1^*(G, B, S) = S.$$

(4) For  $I_1 = (B, B, S)$

$$P(x_1 = \bar{P} \mid B, B, S) = P(x_1 = \bar{P} \mid G, B, S) = \frac{3}{5},$$

$$J_1(B, B, S) = 1, \quad \mu_1^*(B, B, S) = S.$$

(5) For  $I_1 = (G, G, C)$

$$P(x_1 = \bar{P} \mid G, G, C) = \frac{P(x_1 = \bar{P}, G, G \mid C)}{P(G, G \mid C)} = \frac{1}{5},$$

$$J_1(G, G, C) = \frac{2}{5}, \quad \mu_1^*(G, G, C) = C.$$

(6) For  $I_1 = (B, G, C)$

$$P(x_1 = \bar{P} \mid B, G, C) = \frac{11}{23},$$

$$J_1(B, G, C) = \frac{22}{23}, \quad \mu_1^*(B, G, C) = C.$$

(7) For  $I_1 = (G, B, C)$

$$P(x_1 = \bar{P} \mid G, B, C) = \frac{9}{13},$$

$$J_1(G, B, C) = 1, \quad \mu_1^*(G, B, C) = S.$$

(8) For  $I_1 = (B, B, C)$

$$P(x_1 = \bar{P} \mid B, B, C) = \frac{33}{37},$$

$$J_1(B, B, C) = 1, \quad \mu_1^*(B, B, C) = S.$$

Summarizing the results for the last stage, the optimal policy is to continue ( $u_1 = C$ ) if the result of the last inspection was  $G$ , and to stop ( $u_1 = S$ ) if the result of the last inspection was  $B$ .

*First Stage:* Here we use the DP Eq. (5.5) to compute  $J_0(I_0)$  for each of the two possible information vectors  $I_0 = (G)$ ,  $I_0 = (B)$ . We have

$$\begin{aligned} \text{cost of } C &= 2P(x_0 = \bar{P} \mid I_0, C) - E_{z_1} \{ J_1(I_0, z_1, C) \mid I_0, C \} \\ &= 2P(x_0 = \bar{P} \mid I_0, C) + P(z_1 = G \mid I_0, C)J_1(I_0, G, C) \\ &\quad + P(z_1 = B \mid I_0, C)J_1(I_0, B, C), \end{aligned}$$

$$\begin{aligned} \text{cost of } S &= 1 + E_{z_1} \{ J_1(I_0, z_1, S) \mid I_0, S \} \\ &= 1 + P(z_1 = G \mid I_0, S)J_1(I_0, G, S) + P(z_1 = B \mid I_0, S)J_1(I_0, B, S), \end{aligned}$$

and

$$\begin{aligned} J_0(I_0) &= \min \left[ 2P(x_0 = \bar{P} \mid I_0, C) + E_{z_1} \{ J_1(I_0, z_1, C) \mid I_0, C \}, \right. \\ &\quad \left. 1 + E_{z_1} \{ J_1(I_0, z_1, S) \mid I_0, S \} \right] \end{aligned}$$

(1) For  $I_0 = (G)$ : Direct calculation yields

$$P(z_1 = G \mid G, C) = \frac{15}{28}, \quad P(z_1 = B \mid G, C) = \frac{13}{28},$$

$$P(z_1 = G \mid G, S) = \frac{7}{12}, \quad P(z_1 = B \mid G, S) = \frac{5}{12},$$

$$P(x_0 = \bar{P} \mid G, C) = \frac{1}{7},$$

and hence

$$\begin{aligned} J_0(G) &= \min \left[ 2 \cdot \frac{1}{7} + \frac{15}{28}J_1(G, G, C) + \frac{13}{28}J_1(G, B, C), \right. \\ &\quad \left. 1 + \frac{7}{12}J_1(G, G, S) + \frac{5}{12}J_1(G, B, S) \right]. \end{aligned}$$

Using the values of  $J_1$  obtained in the previous stage

$$\begin{aligned} J_0(G) &= \min \left[ 2 \cdot \frac{1}{7} + \frac{15}{28} \cdot \frac{2}{5} + \frac{13}{28} \cdot 1, 1 + \frac{7}{12} \cdot \frac{2}{7} + \frac{5}{12} \cdot 1 \right] \\ &= \min \left[ \frac{27}{28}, \frac{19}{12} \right] = \frac{27}{28}, \end{aligned}$$

$$J_0(G) = \frac{27}{28}, \quad \mu_0^*(G) = C.$$

(2) For  $I_0 = (B)$ : Direct calculation yields

$$P(z_1 = G \mid B, C) = \frac{23}{60}, \quad P(z_1 = B \mid B, C) = \frac{37}{60},$$

$$P(z_1 = G \mid B, S) = \frac{7}{12}, \quad P(z_1 = B \mid B, S) = \frac{5}{12},$$

$$P(x_0 = \bar{P} \mid B, C) = \frac{3}{5},$$

and

$$\begin{aligned} J_0(B) = \min & \left[ 2 \cdot \frac{3}{5} + \frac{23}{60} J_1(B, G, C) + \frac{37}{60} J_1(B, B, C), \right. \\ & \left. 1 + \frac{7}{12} J_1(B, G, S) + \frac{5}{12} J_1(B, B, S) \right]. \end{aligned}$$

Using the values of  $J_1$  obtained in the previous state

$$J_0(B) = \min \left[ \frac{131}{60}, \frac{19}{12} \right] = \frac{19}{12},$$

$$J_0(B) = \frac{19}{12}, \quad \mu_0^*(B) = S.$$

Summarizing, the optimal policy for both stages is to continue if the result of the latest inspection is  $G$ , and to stop and repair otherwise.

The optimal cost is

$$J^* = P(G)J_0(G) + P(B)J_0(B).$$

We can verify that  $P(G) = \frac{7}{12}$  and  $P(B) = \frac{5}{12}$ , so that

$$J^* = \frac{7}{12} \cdot \frac{27}{28} + \frac{5}{12} \cdot \frac{19}{12} = \frac{176}{144}.$$

In the above example, the computation of the optimal policy and the optimal cost by means of the DP algorithm (5.4) and (5.5) was possible because the problem was very simple. It is easy to see that for a more complex problem, the computational requirements of the DP algorithm can be prohibitive, particularly if the number of possible information vectors  $I_k$  is large (or infinite). Unfortunately, even if the control and observation spaces are simple (one-dimensional or finite), the space of the information vector  $I_k$  may have large dimension. This makes the application of the algorithm very difficult or computationally impossible in many cases. However, there are some problems where an analytical solution is possible, and the next two sections deal with such problems.

## 5.2 LINEAR SYSTEMS AND QUADRATIC COST

We will show how the DP algorithm of the preceding section can be used to solve the imperfect state information analog of the linear system/quadratic cost problem of Section 4.1. We have the same linear system

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

and quadratic cost

$$E \left\{ x_N' Q_N x_N + \sum_{k=0}^{N-1} (x_k' Q_k x_k + u_k' R_k u_k) \right\},$$

but now the controller does not have access to the current state. Instead it receives at the beginning of each period  $k$  an observation of the form

$$z_k = C_k x_k + v_k, \quad k = 0, 1, \dots, N-1,$$

where  $z_k \in \mathbb{R}^s$ ,  $C_k$  is a given  $s \times n$  matrix, and  $v_k \in \mathbb{R}^s$  is an observation noise vector with given probability distribution. Furthermore, the vectors  $v_k$  are independent, and independent from  $w_k$  and  $x_0$  as well. We make the same assumptions as in Section 4.1 concerning the input disturbances  $w_k$ , i.e., that they are independent, zero mean, and that they have finite variance. The system matrices  $A_k$ ,  $B_k$  are known; there is no analytical solution of the imperfect information counterpart of the model with random system matrices considered in Section 4.1.

From the DP Eq. (5.4) we have

$$\begin{aligned} J_{N-1}(I_{N-1}) = \min_{u_{N-1}} & \left[ E_{x_{N-1}, w_{N-1}} \{ x_{N-1}' Q_{N-1} x_{N-1} + u_{N-1}' R_{N-1} u_{N-1} \right. \\ & + (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1})' \\ & \left. \cdot Q_N (A_{N-1} x_{N-1} + B_{N-1} u_{N-1} + w_{N-1}) \mid I_{N-1} \} \right] \end{aligned}$$

Since  $E\{w_{N-1} \mid I_{N-1}\} = E\{w_{N-1}\} = 0$ , this expression can be written as

$$\begin{aligned} J_{N-1}(I_{N-1}) = & E_{x_{N-1}} \{ x_{N-1}' (A_{N-1}' Q_N A_{N-1} + Q_{N-1}) x_{N-1} \mid I_{N-1} \} \\ & + E_{w_{N-1}} \{ w_{N-1}' Q_N w_{N-1} \} \\ & + \min_{u_{N-1}} \left[ u_{N-1}' (B_{N-1}' Q_N B_{N-1} + R_{N-1}) u_{N-1} \right. \\ & \left. + 2E\{x_{N-1} \mid I_{N-1}\}' A_{N-1}' Q_N B_{N-1} u_{N-1} \right]. \end{aligned} \quad (5.6)$$

The minimization yields the optimal policy for the last stage:

$$\begin{aligned} u_{N-1}^* &= \mu_{N-1}(I_{N-1}) \\ &= -(B'_{N-1}Q_N B_{N-1} + R_{N-1})^{-1} B'_{N-1} Q_N A_{N-1} E\{x_{N-1} | I_{N-1}\}, \end{aligned} \quad (5.7)$$

and upon substitution in Eq. (5.6), we obtain

$$\begin{aligned} J_{N-1}(I_{N-1}) &= \underset{x_{N-1}}{E} \left\{ x'_{N-1} K_{N-1} x_{N-1} | I_{N-1} \right\} \\ &\quad + \underset{x_{N-1}}{E} \left\{ (x_{N-1} - E\{x_{N-1} | I_{N-1}\})' \right. \\ &\quad \cdot P_{N-1}(x_{N-1} - E\{x_{N-1} | I_{N-1}\}) | I_{N-1} \left. \right\} \\ &\quad + \underset{w_{N-1}}{E} \{ w'_{N-1} Q_N w_{N-1} \}, \end{aligned}$$

where the matrices  $K_{N-1}$  and  $P_{N-1}$  are given by

$$\begin{aligned} P_{N-1} &= A'_{N-1} Q_N B_{N-1} (R_{N-1} + B'_{N-1} Q_N B_{N-1})^{-1} B'_{N-1} Q_N A_{N-1}, \\ K_{N-1} &= A'_{N-1} Q_N A_{N-1} - P_{N-1} - Q_{N-1}. \end{aligned}$$

Note that the optimal policy (5.6) is identical to its perfect state information counterpart except that  $x_{N-1}$  is replaced by its conditional expectation  $E\{x_{N-1} | I_{N-1}\}$ . Note also that the cost-to-go  $J_{N-1}(I_{N-1})$  exhibits a corresponding similarity to its perfect state information counterpart except that  $J_{N-1}(I_{N-1})$  contains an additional middle term, which is in effect a penalty for estimation error.

Now the DP equation for period  $N-2$  is

$$\begin{aligned} J_{N-2}(I_{N-2}) &= \min_{u_{N-2}} \left[ \underset{x_{N-2}, w_{N-2}, z_{N-1}}{E} \left\{ x'_{N-2} Q_{N-2} x_{N-2} + u'_{N-2} R_{N-2} u_{N-2} \right. \right. \\ &\quad \left. \left. + J_{N-1}(I_{N-1}) | I_{N-2}, u_{N-2} \right\} \right] \\ &= E\{x'_{N-2} Q_{N-2} x_{N-2} | I_{N-2}\} \\ &\quad + \min_{u_{N-2}} \left[ u'_{N-2} R_{N-2} u_{N-2} + E\{x'_{N-1} K_{N-1} x_{N-1} | I_{N-2}, u_{N-2}\} \right] \\ &\quad + E\left\{ (x_{N-1} - E\{x_{N-1} | I_{N-1}\})' \right. \\ &\quad \cdot P_{N-1}(x_{N-1} - E\{x_{N-1} | I_{N-1}\}) | I_{N-2}, u_{N-2} \left. \right\} \\ &\quad + \underset{w_{N-1}}{E} \{ w'_{N-1} Q_N w_{N-1} \}. \end{aligned} \quad (5.8)$$

Note that we have excluded the next to last term from the minimization with respect to  $u_{N-2}$ . We have done so since this term turns out to be independent of  $u_{N-2}$ . To show this fact, we need the following lemma.

The lemma says essentially that the quality of estimation as expressed by the statistics of the error  $x_k - E\{x_k | I_k\}$  cannot be influenced by the choice of control. This is due to the linearity of both the system and the measurement equation. In particular,  $x_k$  and  $E\{x_k | I_k\}$  contain the same linear terms in  $(u_0, \dots, u_{k-1})$ , which cancel each other out.

**Lemma 5.2.1:** For every  $k$ , there is a function  $M_k$  such that we have

$$x_k - E\{x_k | I_k\} = M_k(x_0, w_0, \dots, w_{k-1}, v_0, \dots, v_k),$$

independently of the policy being used.

**Proof:** Fix a policy and consider the following two systems. In the first system there is control as determined by the policy,

$$x_{k-1} = A_k x_k + B_k u_k + w_k, \quad z_k = C_k x_k + v_k,$$

while in the second system there is no control,

$$\bar{x}_{k+1} = A_k \bar{x}_k + \bar{w}_k, \quad \bar{z}_k = C_k \bar{x}_k + \bar{v}_k.$$

We consider the evolution of these two systems when their initial conditions are identical,

$$x_0 = \bar{x}_0,$$

and when their system disturbance and observation noise vectors are also identical,

$$w_k = \bar{w}_k, \quad v_k = \bar{v}_k, \quad k = 0, 1, \dots, N-1.$$

Consider the vectors

$$\begin{aligned} Z^k &= (z_0, \dots, z_k)', & \bar{Z}^k &= (\bar{z}_0, \dots, \bar{z}_k)', \\ W^k &= (w_0, \dots, w_k)', & V^k &= (v_0, \dots, v_k)', & U^k &= (u_0, \dots, u_k)'. \end{aligned}$$

Linearity implies the existence of matrices  $F_k$ ,  $G_k$ , and  $H_k$  such that

$$\begin{aligned} x_k &= F_k x_0 + G_k U^{k-1} + H_k W^{k-1}, \\ \bar{x}_k &= F_k x_0 + H_k V^{k-1}. \end{aligned}$$

Since the vector  $U^{k-1} = (u_0, \dots, u_{k-1})'$  is part of the information vector  $I_k$ , we have  $U^{k-1} = E\{U^{k-1} | I_k\}$ , so

$$\begin{aligned} E\{x_k | I_k\} &= F_k E\{x_0 | I_k\} + G_k U^{k-1} + H_k E\{W^{k-1} | I_k\}, \\ E\{\bar{x}_k | I_k\} &= F_k E\{x_0 | I_k\} + H_k E\{V^{k-1} | I_k\}. \end{aligned}$$



We thus obtain

$$x_k - E\{x_k \mid I_k\} = \bar{x}_k - E\{\bar{x}_k \mid I_k\}.$$

From the equations for  $z_k$  and  $\bar{z}_k$ , we see that

$$\bar{Z}^k = Z^k - R_k U^{k-1} = S_k W^{k-1} + T_k V^k,$$

where  $R_k$ ,  $S_k$ , and  $T_k$  are some matrices of appropriate dimension. Thus, the information provided by  $I_k = (Z^k, U^{k-1})$  regarding  $\bar{x}_k$  is summarized in  $\bar{Z}^k$ , and we have  $E\{\bar{x}_k \mid I_k\} = E\{\bar{x}_k \mid \bar{Z}^k\}$ , so that

$$x_k - E\{x_k \mid I_k\} = \bar{x}_k - E\{\bar{x}_k \mid \bar{Z}^k\}.$$

The function  $M_k$  given by

$$M_k(x_0, w_0, \dots, w_{k-1}, v_0, \dots, v_k) = \bar{x}_k - E\{\bar{x}_k \mid \bar{Z}^k\}$$

serves the purpose stated in the lemma. **Q.E.D.**

We can now justify excluding the term

$$E\{(x_{N-1} - E\{x_{N-1} \mid I_{N-1}\})' P_{N-1} (x_{N-1} - E\{x_{N-1} \mid I_{N-1}\}) \mid I_{N-2}, u_{N-2}\}$$

from the minimization in Eq. (5.8), as being independent of  $u_{N-2}$ . Indeed, by using the lemma, we see that

$$x_{N-1} - E\{x_{N-1} \mid I_{N-1}\} = \xi_{N-1},$$

where  $\xi_{N-1}$  is a function of  $x_0, w_0, \dots, w_{N-2}, v_0, \dots, v_{N-1}$ . Since  $\xi_{N-1}$  is independent of  $u_{N-2}$ , the conditional expectation of  $\xi_{N-1}' P_{N-1} \xi_{N-1}$  satisfies

$$E\{\xi_{N-1}' P_{N-1} \xi_{N-1} \mid I_{N-2}, u_{N-2}\} = E\{\xi_{N-1}' P_{N-1} \xi_{N-1} \mid I_{N-2}\}.$$

Returning now to our problem, the minimization in Eq. (5.8) yields, using an argument similar to the one for the last stage,

$$\begin{aligned} u_{N-2}^* &= \mu_{N-2}^*(I_{N-2}) \\ &= -(R_{N-2} + B_{N-2}' K_{N-1} B_{N-2})^{-1} B_{N-2}' K_{N-1} A_{N-2} E\{x_{N-2} \mid I_{N-2}\}. \end{aligned}$$

We can proceed similarly to obtain the optimal policy for every stage:

$$\mu_k^*(I_k) = L_k E\{x_k \mid I_k\}, \quad (5.9)$$

where the matrix  $L_k$  is given by

$$L_k = -(R_k + B_k' K_{k+1} B_k)^{-1} B_k' K_{k+1} A_k,$$

with the matrices  $K_k$  given recursively by the Riccati equation

$$K_N = Q_N,$$

$$P_k = A_k' K_{k+1} B_k (R_k + B_k' K_{k+1} B_k)^{-1} B_k' K_{k+1} A_k,$$

$$K_k = A_k' K_{k+1} A_k - P_k - Q_k.$$

The key step in this derivation is that at stage  $k$  of the DP algorithm, the minimization over  $u_k$  that defines  $J_k(I_k)$  involves the additional terms

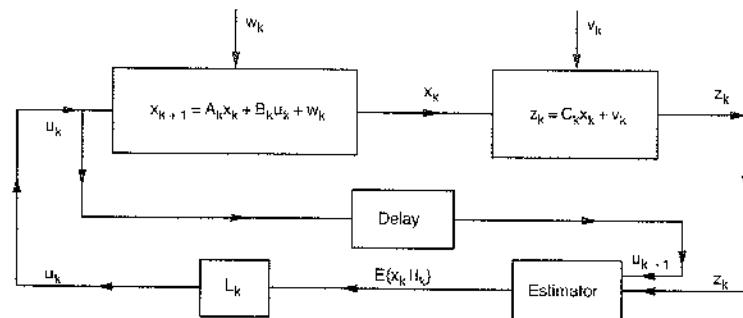
$$E\{(x_s - E\{x_s \mid I_s\})' P_s (x_s - E\{x_s \mid I_s\}) \mid I_k, u_k\},$$

where  $s = k+1, \dots, N-1$ . By using the argument of the proof of the earlier lemma, it can be seen that none of these terms depends on  $u_k$  so that the presence of these terms does not affect the minimization in the DP algorithm. As a result, the optimal policy is the same as the one for the perfect information case, except that the state  $x_k$  is replaced by its conditional expectation  $E\{x_k \mid I_k\}$ .

It is interesting to note that the optimal controller can be decomposed into the two parts shown in Fig. 5.2.1:

- (a) An *estimator*, which uses the data to generate the conditional expectation  $E\{x_k \mid I_k\}$ .
- (b) An *actuator*, which multiplies  $E\{x_k \mid I_k\}$  by the gain matrix  $L_k$  and applies the control input  $u_k = L_k E\{x_k \mid I_k\}$ .

Furthermore, the gain matrix  $L_k$  is independent of the statistics of the problem and is the same as the one that would be used if we were faced with the deterministic problem, where  $w_k$  and  $x_0$  would be fixed and equal to their expected values. On the other hand, as shown in Appendix E, the estimate  $\hat{x}$  of a random vector  $x$  given some information (random vector)  $I$ , which minimizes the mean squared error  $E_x\{\|x - \hat{x}\|^2 \mid I\}$  is precisely the conditional expectation  $E\{x \mid I\}$  (expand the quadratic form and set to zero the derivative with respect to  $\hat{x}$ ). Thus the *estimator portion of the optimal controller is an optimal solution of the problem of estimating the state  $x_k$  assuming no control takes place, while the actuator portion is an optimal solution of the control problem assuming perfect state information prevails*. This property, which shows that the two portions of the optimal controller can be designed independently as optimal solutions of an estimation and a control problem, has been called the *separation theorem for*



**Figure 5.2.1** Structure of the optimal controller for the linear-quadratic problem. It consists of an estimator, which generates the conditional expectation  $E\{x_k | I_k\}$ , and an actuator, which multiplies  $E\{x_k | I_k\}$  by the gain matrix  $L_k$ .

*linear systems and quadratic cost* and occupies a central position in modern automatic control theory.

Another interesting observation is that the optimal controller applies at each time  $k$  the control that would be applied when faced with the deterministic problem of minimizing the cost-to-go

$$x'_N Q_N x_N + \sum_{i=k}^{N-1} (x'_i Q_i x_i + u'_i R_i u_i),$$

and the input disturbances  $w_k, w_{k+1}, \dots, w_{N-1}$  and current state  $x_k$  were known and fixed at their conditional expected values, which are zero and  $E\{x_k | I_k\}$ , respectively. This is another manifestation of the *certainty equivalence principle*, which was referred to in Section 4.1. A similar result holds in the case of correlated disturbances; see Exercise 5.1.

### Implementation Aspects – Steady-State Controller

As explained in the perfect information case, the linear form of the actuator portion of the optimal policy is particularly attractive for implementation. In the imperfect information case, however, we need to implement an estimator that produces the conditional expectation

$$\hat{x}_k = E\{x_k | I_k\},$$

and this is not easy in general. Fortunately, in the important special case, where the *disturbances  $w_k, v_k$ , and the initial state  $x_0$  are Gaussian random vectors*, a convenient implementation of the estimator is possible by means of the Kalman filtering algorithm, which is developed in Appendix E. This algorithm is organized recursively so that to produce  $\hat{x}_{k+1}$  at time  $k+1$ ,

only the most recent measurement  $z_{k+1}$  is needed, together with  $\hat{x}_k$  and  $u_k$ . In particular, we have for all  $k = 0, \dots, N-1$ ,

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + \Sigma_{k+1|k-1} C'_{k+1} N_{k+1}^{-1} (z_{k+1} - C_{k+1}(A_k \hat{x}_k + B_k u_k)),$$

and

$$\hat{x}_0 = E\{x_0\} + \Sigma_{0|0} C'_0 N_0^{-1} (z_0 - C_0 E\{x_0\}),$$

where the matrices  $\Sigma_{k|k}$  are precomputable and are given recursively by

$$\Sigma_{k+1|k+1} = \Sigma_{k+1|k} - \Sigma_{k+1|k} C'_{k+1} (C_{k+1} \Sigma_{k+1|k} C'_{k+1} + N_{k+1})^{-1} C_{k+1} \Sigma_{k+1|k},$$

$$\Sigma_{k+1|k} = A_k \Sigma_{k|k} A'_k + M_k, \quad k = 0, 1, \dots, N-1,$$

with

$$\Sigma_{0|0} = S - SC'_0(C_0 SC'_0 + N_0)^{-1} C_0 S.$$

In these equations,  $M_k$ ,  $N_k$ , and  $S$  are the covariance matrices of  $w_k$ ,  $v_k$ , and  $x_0$ , respectively, and we assume that  $w_k$  and  $v_k$  have zero mean; that is

$$E\{w_k\} = E\{v_k\} = 0,$$

$$M_k = E\{w_k w'_k\}, \quad N_k = E\{v_k v'_k\},$$

$$S = E\{(x_0 - E\{x_0\})(x_0 - E\{x_0\})'\}.$$

In addition, the matrices  $N_k$  are assumed to be positive definite.

Consider now the case where the system and measurement equations, and the disturbance statistics are stationary. We can then drop subscripts from the system matrices. Assume that *the pair  $(A, B)$  is controllable and that the matrix  $Q$  can be written as  $Q = F'F$ , where  $F$  is a matrix such that the pair  $(A, F)$  is observable*. By the theory of Section 4.1, if the horizon tends to infinity, the optimal controller tends to the steady-state policy

$$\mu^*(I_k) = L \hat{x}_k, \quad (5.10)$$

where

$$L = -(R + B' K B)^{-1} B' K A, \quad (5.11)$$

and  $K$  is the unique positive semidefinite symmetric solution of the algebraic Riccati equation

$$K = A'(K - KB(R + B'KB)^{-1}B'K)A + Q.$$

By a similar argument, it can be shown (see Appendix E) that  $\hat{x}_k$  can be generated in the limit as  $k \rightarrow \infty$  by a steady-state Kalman filtering algorithm:

$$\hat{x}_{k+1} = (A + BL)\hat{x}_k + \Sigma C' N^{-1} (z_{k+1} - C(A + BL)\hat{x}_k), \quad (5.12)$$

where  $\bar{\Sigma}$  is given by

$$\bar{\Sigma} = \Sigma - \Sigma C' (C \Sigma C' + N)^{-1} C \Sigma,$$

and  $\Sigma$  is the unique positive semidefinite symmetric solution of the Riccati equation

$$\Sigma = A(\Sigma - \Sigma C' (C \Sigma C' + N)^{-1} C \Sigma) A' + M.$$

The assumptions required for this are that the pair  $(A, C)$  is observable and that the matrix  $M$  can be written as  $M = DD'$ , where  $D$  is a matrix such that the pair  $(A, D)$  is controllable. The steady-state controller of Eqs. (5.10)-(5.12) is particularly attractive for practical implementation. Furthermore, as shown in Appendix E, it results in a stable closed-loop system, under the preceding controllability and observability assumptions.

### 5.3 MINIMUM VARIANCE CONTROL OF LINEAR SYSTEMS

We have considered so far the control of linear systems in state variable form as in the previous section. However, linear systems are often modeled by means of an input-output equation, which is more economical in the number of parameters needed to describe the system dynamics. In this section we consider single-input, single-output, linear, time-invariant systems, and a special type of quadratic cost function. The resulting optimal policy is particularly simple and has found wide application. We first introduce some of the basic facts regarding linear systems in input-output form. Detailed discussions may be found in the books by Åström and Wittenmark [AsW84], [AsW90], Goodwin and Sin [GoS84], and Whittle [Whi63].

We consider a single-input single-output discrete-time linear system, which is specified by an equation of the form

$$y_k + a_1 y_{k-1} + \cdots + a_m y_{k-m} = b_0 u_k + b_1 u_{k-1} + \cdots + b_m u_{k-m}, \quad (5.13)$$

where  $a_i, b_i$  are given scalars. The scalar sequences  $\{u_k | k = 0, \pm 1, \pm 2, \dots\}$ ,  $\{y_k | k = 0, \pm 1, \pm 2, \dots\}$  are viewed as the input and output of the system, respectively. Note that we allow time to extend to  $-\infty$  as well as  $+\infty$ ; this will be useful for describing generic properties of the system relating to stability. We will later revert to our usual convention of starting at time 0 and proceeding forward.

It is convenient to describe this type of system by means of the *backward shift operator*, denoted  $s$ , which when operating on a sequence  $\{x_k | k = 0, \pm 1, \pm 2, \dots\}$  shifts its index by one unit; that is,

$$s(x_k) = x_{k-1}, \quad k = 0, \pm 1, \pm 2, \dots$$

We denote by  $s^r$  the operator resulting from  $r$  successive applications of  $s$ :

$$s^r(x_k) = x_{k-r}, \quad k = 0, \pm 1, \pm 2, \dots \quad (5.14)$$

We also write for simplicity  $s^r x_k = x_{k-r}$ . The *forward shift operator*, denoted  $s^{-1}$ , is the inverse of  $s$  and is defined by

$$s^{-1}(x_k) = x_{k+1}, \quad k = 0, \pm 1, \pm 2, \dots$$

Thus the notation (5.14) holds for all integers  $r$ . We can form linear combinations of operators of the form  $s^r$ . Thus, for example, the operator  $(s + 2s^2)$  is defined by

$$(s + 2s^2)(x_k) = x_{k-1} + 2x_{k-2}, \quad k = 0, \pm 1, \pm 2, \dots$$

With this notation, Eq. (5.13) can be written as

$$A(s)y_k = B(s)u_k,$$

where  $A(s), B(s)$  are the operators

$$A(s) = 1 + a_1 s + \cdots + a_m s^m,$$

$$B(s) = b_0 + b_1 s + \cdots + b_m s^m.$$

Sometimes it is convenient to write the equation  $A(s)y_k = B(s)u_k$  as

$$y_k = \frac{B(s)}{A(s)} u_k$$

or

$$\frac{A(s)}{B(s)} y_k = u_k.$$

The meaning of both equations is that the sequences  $\{y_k\}$  and  $\{u_k\}$  are related via  $A(s)y_k = B(s)u_k$ . There is a certain ambiguity here in that, for a fixed  $\{u_k\}$ , the equation  $A(s)y_k = B(s)u_k$  has an infinite number of solutions in  $\{y_k\}$ . For example, the equation

$$y_k + a y_{k-1} = u_k$$

for  $u_k \equiv 0$  has as solutions all sequences of the form  $y_k = \beta(-a)^k$ , where  $\beta$  is any scalar; the solution becomes unique only after some boundary condition for the sequence  $\{y_k\}$  is specified. As will be discussed shortly, however, for stable systems and for a *bounded* sequence  $\{u_k\}$  there is a unique solution  $\{y_k\}$  that is *bounded*. It is this solution that will be denoted by  $(B(s)/A(s))u_k$  in what follows. The reader who is familiar with linear

dynamic system theory will note that  $B(s)/A(s)$  can be viewed as a *transfer function* involving  $z$ -transforms.

We now introduce some terminology. When the sequences  $\{y_k\}$  and  $\{u_k\}$  satisfy  $A(s)y_k = B(s)u_k$ , we say that  $y_k$  is *obtained by passing  $u_k$  through the filter  $B(s)/A(s)$* . This comes from engineering terminology, where linear time-invariant systems are commonly referred to as filters. We also refer to the equation  $A(s)y_k = B(s)u_k$  as the filter  $B(s)/A(s)$ .

A filter  $B(s)/A(s)$  is said to be *stable* if the polynomial  $A(s)$  has all its (complex) roots strictly outside the unit circle of the complex plane; that is,  $|p| > 1$  for all complex  $p$  satisfying  $A(p) = 0$ . A stable filter  $B(s)/A(s)$  has the following two properties:

- (a) Every solution  $\{y_k\}$  of

$$A(s)y_k = 0$$

satisfies  $\lim_{k \rightarrow \infty} y_k = 0$ ; that is, the output  $y_k$  tends to zero if the input sequence  $\{u_k\}$  is identically zero.

- (b) For every bounded sequence  $\{\bar{u}_k\}$ , the equation

$$A(s)y_k = B(s)\bar{u}_k$$

has a *unique* solution  $\{\bar{y}_k\}$  within the class of bounded sequences. Furthermore, every solution  $\{y_k\}$  (possibly unbounded) of the equation satisfies

$$\lim_{k \rightarrow \infty} (y_k - \bar{y}_k) = 0.$$

For example, consider the system

$$y_k - 0.5y_{k-1} = u_k.$$

Given the bounded input sequence  $\bar{u}_k = \{\dots, 1, 1, 1, \dots\}$ , the set of all solutions is given by

$$y_k = 2 + \frac{\beta}{2^k},$$

where  $\beta$  is a scalar, but of these the only bounded solution is  $\bar{y}_k = \{\dots, 2, 2, 2, \dots\}$ . The solution  $\{\bar{y}_k\}$  can thus be naturally associated with the input sequence  $\{u_k\}$ ; it is also known as the *forced response* of the system due to the input  $\{u_k\}$ .

### ARMAX Models – Reduction to State Space Form

We now consider a linear system with output  $y_k$ , which is driven by two inputs: a random noise input  $\epsilon_k$ , and a control input  $u_k$ . It has the form

$$y_k + a_1y_{k-1} + \dots + a_my_{k-m} = b_1u_{k-1} + \dots + b_mu_{k-m} + \epsilon_k + c_1\epsilon_{k-1} + \dots + c_m\epsilon_{k-m}, \quad (5.15)$$

and it is known as an ARMAX model (AutoRegressive, Moving Average, with eXogenous input). We assume throughout that the random variables  $\epsilon_k$  are mutually independent. We can write the model in the shorthand form

$$A(s)y_k = B(s)u_k + C(s)\epsilon_k,$$

where the polynomials  $A(s)$ ,  $B(s)$ , and  $C(s)$  are given by

$$A(s) = 1 + a_1s + \dots + a_ms^m,$$

$$B(s) = b_1s + \dots + b_ms^m,$$

$$C(s) = 1 + c_1s + \dots + c_ms^m.$$

The ARMAX model is very common and its derivation is outlined in Appendix F, where it is shown that without loss of generality we can assume that  $C(s)$  has no roots strictly inside the unit circle. For much of the analysis in subsequent sections, it will be necessary to exclude the critical case where  $C(s)$  has roots on the unit circle and assume that  $C(s)$  has all its roots strictly outside the unit circle. This assumption is usually satisfied in practice.

In several situations, analysis and algorithms relating to the ARMAX model are greatly simplified if  $C(s) = 1$  so that the noise terms  $C(s)\epsilon_k = \epsilon_k$  are independent. However, this is typically an unrealistic assumption. To emphasize this point and see how easily the noise can be correlated, suppose that we have a first-order system

$$x_{k+1} = ax_k + w_k,$$

where we observe

$$y_k = x_k + v_k.$$

Then

$$\begin{aligned} y_{k+1} &= x_{k+1} + v_{k+1} \\ &= ax_k + w_k + v_{k+1} \\ &= a(y_k - v_k) + w_k + v_{k+1}, \end{aligned}$$

so finally

$$y_{k+1} = ay_k + v_{k+1} - av_k + w_k.$$

However, the noise sequence  $\{v_{k+1} - av_k + w_k\}$  is correlated even if  $\{v_k\}$  and  $\{w_k\}$  are individually and mutually independent.

The ARMAX model (5.15) can be put into state space form. The process is based on state augmentation and can perhaps be best understood in terms of an example. Consider the system

$$y_k + a_1y_{k-1} + a_2y_{k-2} = b_1u_{k-1} + b_2u_{k-2} + \epsilon_k + c_1\epsilon_{k-1}. \quad (5.16)$$

We have

$$\begin{pmatrix} y_{k+1} \\ y_k \\ u_k \\ \epsilon_{k+1} \end{pmatrix} = \begin{pmatrix} -a_1 & -a_2 & b_2 & c_1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_k \\ y_{k-1} \\ u_{k-1} \\ \epsilon_k \end{pmatrix} + \begin{pmatrix} b_1 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \epsilon_{k+1} \\ 0 \\ 0 \\ \epsilon_{k+1} \end{pmatrix}. \quad (5.17)$$

By setting

$$x_k = \begin{pmatrix} y_k \\ y_{k-1} \\ u_{k-1} \\ \epsilon_k \end{pmatrix}, \quad w_k = \begin{pmatrix} \epsilon_{k-1} \\ 0 \\ 0 \\ \epsilon_{k+1} \end{pmatrix},$$

$$A = \begin{pmatrix} -a_1 & -a_2 & b_2 & c_1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

we can write Eq. (5.17) as

$$x_{k+1} = Ax_k + Bu_k + w_k,$$

where  $\{w_k\}$  is a stationary independent process. We have arrived at this state space model through state augmentation. Notice that the state  $x_k$  includes  $\epsilon_k$ . Thus if the controller is assumed to know at time  $k$  only the present and past outputs  $y_k, y_{k-1}, \dots$ , and past controls  $u_{k-1}, u_{k-2}, \dots$  (but not  $\epsilon_k, \epsilon_{k-1}, \dots$ ), we are faced with a model of imperfect state information. If  $c_1 = 0$  in Eq. (5.16) then the state space model can be simplified so that

$$x_k = \begin{pmatrix} y_k \\ y_{k-1} \\ u_{k-1} \end{pmatrix},$$

in which case we have perfect state information. More generally, we have perfect state information in the ARMAX model (5.15) if  $b_1 \neq 0$  and  $c_1 = c_2 = \dots = c_m = 0$ .

### Minimum Variance Control: Perfect State Information Case

Consider the perfect state information case of the ARMAX model (5.15):

$$y_k + a_1 y_{k-1} + \dots + a_m y_{k-m} = b_1 u_{k-1} + \dots + b_m u_{k-m} + \epsilon_k,$$

where  $b_1 \neq 0$ . A problem of interest, known as the *minimum variance control problem*, is to select  $u_k$  as a function of the present and past outputs  $y_k, y_{k-1}, \dots$ , as well as the past controls  $u_{k-1}, u_{k-2}, \dots$ , so as to minimize the cost

$$E \left\{ \sum_{k=1}^N (y_k)^2 \right\}.$$

There are no constraints on  $u_k$ . By transforming the system to state space form, we see that this problem can be reduced to a perfect state information linear-quadratic problem where the state  $x_k$  is

$$(y_k, y_{k-1}, \dots, y_{k-m+1}, u_{k-1}, \dots, u_{k-m+1})'.$$

The problem is of the same nature as the linear-quadratic problem of Section 4.1 except that the corresponding matrices  $R_k$  in the quadratic cost function are zero here. Nonetheless, in Section 4.1 we used the invertibility of  $R_k$  only to ensure that various matrices in the optimal policy and the Riccati equation are invertible. If invertibility of these matrices can be guaranteed by other means, the same analysis applies even if  $R_k$  is positive semidefinite. This is indeed the case here. An analysis analogous to the one of Section 4.1 shows that the optimal control  $u_k^*$  at time  $k$  (given  $y_k, y_{k-1}, \dots, y_{k-m+1}$  and  $u_{k-1}, \dots, u_{k-m+1}$ ) is the same as the one that would be applied if all future disturbances  $\epsilon_{k+1}, \dots, \epsilon_N$  were set equal to zero, their expected value (certainty equivalence). It follows that

$$\mu_k^*(y_k, \dots, y_{k-m+1}, u_{k-1}, \dots, u_{k-m+1}) = \frac{1}{b_1} (a_1 y_k + \dots + a_m y_{k-m+1} - b_2 u_{k-1} - \dots - b_m u_{k-m+1}),$$

and  $\{u_k^*\}$  is generated via the equation

$$b_1 u_k^* + b_2 u_{k-1}^* + \dots + b_m u_{k-m+1}^* = a_1 y_k + a_2 y_{k-1} + \dots + a_m y_{k-m+1}.$$

In other words,  $\{u_k^*\}$  is generated by passing  $\{y_k\}$  through the linear filter  $\bar{A}(s)/\bar{B}(s)$ , where

$$\bar{A}(s) = a_1 + a_2 s + \dots + a_m s^{m-1} = s^{-1}(A(s) - 1),$$

$$\bar{B}(s) = b_1 + b_2 s + \dots + b_m s^{m-1} = s^{-1}B(s),$$

as shown in Fig. 5.3.1. The resulting closed-loop system is

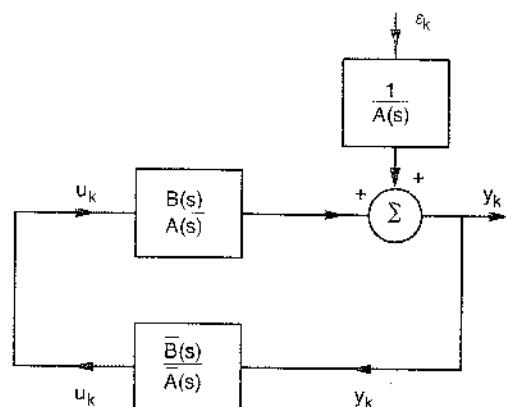
$$y_k = \epsilon_k \quad (5.18)$$

and the associated cost is

$$NE\{(\epsilon_k)^2\}.$$

Notice that the optimal policy, called *minimum variance control law*, is time invariant and does not depend on the horizon  $N$ .

Whereas the optimal closed-loop system as given by Eq. (5.18) is clearly stable, we can anticipate serious difficulties if the filter  $\bar{A}(s)/\bar{B}(s)$  in the feedback loop is unstable. For if  $\bar{B}(s)$  has some roots inside the unit circle, then the sequence  $\{u_k\}$  will tend to be unbounded. This is illustrated by the following example.



**Figure 5.3.1** Minimum variance control with perfect state information. Structure of the optimal closed-loop system, where  $A(s) = 1 + a_1s + \dots + a_ms^m$ ,  $B(s) = b_1s + \dots + b_ms^m$ ,  $\bar{A}(s) = s^{-1}(A(s) - 1)$ , and  $\bar{B}(s) = s^{-1}B(s)$ .

### Example 5.3.1 (An Optimal but Unstable Controller)

Consider the system

$$y_k + y_{k-1} = u_{k-1} - 2u_{k-2} + \epsilon_k.$$

The optimal policy is

$$u_k = 2u_{k-1} + y_k$$

and the optimal closed-loop system is

$$y_k = \epsilon_k,$$

which is a stable system. On the other hand, the last two equations yield

$$u_k = 2u_{k-1} + \epsilon_k.$$

Thus,  $u_k$  is generated by an *unstable* system, and in fact it is given by

$$u_k = \sum_{n=0}^k 2^n \epsilon_{k-n}.$$

Therefore, even though the output  $y_k$  stays bounded, the control  $u_k$  typically becomes unbounded.

For another view of the same difficulty, suppose that the coefficients  $b_1, \dots, b_m$  of  $\bar{B}(s)$  are slightly different from the ones of the true system.

We will show that if the feedback filter  $\bar{A}(s)/\bar{B}(s)$  is unstable, then the closed-loop system is also unstable in the sense that both  $u_k$  and  $y_k$  become unbounded with probability one.

Assume that the system is governed by

$$A^0(s)y_k = B^0(s)u_k + \epsilon_k, \quad (5.19)$$

while the policy is calculated under the assumption that the system model is

$$A(s)y_k = B(s)u_k + \epsilon_k,$$

where the coefficients of  $A(s)$  and  $B(s)$  differ slightly from those of  $A^0(s)$ ,  $B^0(s)$ . Define  $\bar{A}^0(s)$ ,  $\bar{B}^0(s)$  by

$$1 + s\bar{A}^0(s) = A^0(s),$$

$$s\bar{B}^0(s) = B^0(s).$$

Note that  $\bar{A}^0(s) = \bar{A}(s)$  and  $\bar{B}^0(s) = \bar{B}(s)$  if  $A^0(s) = A(s)$ ,  $B^0(s) = B(s)$ . By multiplying Eq. (5.19) with  $\bar{B}(s)$  and by using the relation defining the optimal policy

$$\bar{B}(s)u_k = \bar{A}(s)y_k,$$

we obtain

$$\bar{B}(s)A^0(s)y_k = B^0(s)\bar{A}(s)y_k + \bar{B}(s)\epsilon_k.$$

If the coefficients of  $\bar{A}^0(s)$  and  $\bar{B}^0(s)$  are close to those of  $\bar{A}(s)$ ,  $\bar{B}(s)$ , then the roots of the polynomial

$$\bar{B}(s) + s(\bar{B}(s)\bar{A}^0(s) - \bar{B}^0(s)\bar{A}(s))$$

are close to the roots of  $\bar{B}(s)$ . Thus the *closed-loop system is stable only if the roots of  $\bar{B}(s)$  are outside the unit circle*, or equivalently, if and only if the filter  $\bar{A}(s)/\bar{B}(s)$  is stable. If our model is exact, the closed-loop system will be stable due to what is commonly referred to as a *pole-zero cancellation*. However, the slightest modeling discrepancy will induce instability.

The conclusion from the preceding analysis is that the minimum variance control law is advisable only if it can be realized through a stable filter [ $\bar{B}(s)$  has roots outside the unit circle]. Even if  $\bar{B}(s)$  has its roots outside the unit circle, but some of these roots are near the unit circle, the performance of the minimum variance policy can be very sensitive to variations in the parameters of the polynomials  $A(s)$  and  $B(s)$ . One way to overcome this sensitivity is to change the cost to

$$E \left\{ \sum_{k=1}^N ((y_k)^2 + R(u_{k-1})^2) \right\},$$

where  $R$  is some positive parameter. This requires solution via the Riccati equation as in Section 4.1. For a detailed derivation, see Åström [Ast83].

In some problems, the system equation includes an additional external input sequence  $\{v_k\}$ , the values of which can be measured by the controller as they occur. In particular, consider the scalar system

$$\begin{aligned} y_k + a_1 y_{k-1} + \cdots + a_m y_{k-m} &= b_1 u_{k-1} + \cdots + b_m u_{k-m} \\ &\quad + d_1 v_{k-1} + \cdots + d_m v_{k-m} + \epsilon_k, \end{aligned}$$

where each value  $v_k$  becomes known to the controller without error at time  $k$ . The minimum variance controller then takes the form

$$\begin{aligned} \mu_k^*(y_k, \dots, y_{k-m+1}, u_{k-1}, \dots, u_{k-m-1}, v_k, \dots, v_{k-m+1}) \\ = \frac{1}{b_1} (a_1 y_k + \cdots + a_m y_{k-m+1} - d_1 v_k - \cdots - d_m v_{k-m+1} \\ - b_2 u_{k-1} - \cdots - b_m u_{k-m+1}), \end{aligned}$$

and the optimal controls  $\{u_k^*\}$  are generated by

$$\bar{B}(s)u_k^* = \bar{A}(s)y_k - \bar{D}(s)v_k,$$

where

$$\begin{aligned} \bar{A}(s) &= a_1 + a_2 s + \cdots + a_m s^{m-1}, \\ \bar{B}(s) &= b_1 + b_2 s + \cdots + b_m s^{m-1}, \\ \bar{D}(s) &= d_1 + d_2 s + \cdots + d_m s^{m-1}. \end{aligned}$$

The closed-loop system is again  $y_k = \epsilon_k$ , but for practical purposes it is stable only if  $\bar{B}(s)$  has its roots outside the unit circle. The process whereby external inputs are measured and used for control is commonly referred to as *feedforward control*.

### Imperfect State Information Case

Consider now the general ARMAX model

$$\begin{aligned} y_k + a_1 y_{k-1} + \cdots + a_m y_{k-m} &= b_M u_{k-M} + \cdots + b_m u_{k-m} \\ &\quad + \epsilon_k + c_1 \epsilon_{k-1} + \cdots + c_m \epsilon_{k-m} \end{aligned}$$

or, equivalently,

$$A(s)y_k = B(s)u_k + C(s)\epsilon_k,$$

where

$$\begin{aligned} A(s) &= 1 + a_1 s + \cdots + a_m s^m, \\ B(s) &= b_M s^M + \cdots + b_m s^m, \\ C(s) &= 1 + c_1 s + \cdots + c_m s^m. \end{aligned}$$

We assume the following:

- (1)  $b_M \neq 0$  and  $1 \leq M \leq m$ .
- (2)  $\{\epsilon_k\}$  is a zero mean, independent, stationary process.
- (3) The polynomial  $C(s)$  has all its roots outside the unit circle. (As explained in Appendix F, this assumption is not overly restrictive.)

The controller knows at each time  $k$  the past inputs and outputs. Thus the information vector at time  $k$  is

$$I_k = (y_k, y_{k-1}, \dots, y_{-m+1}, u_{k-1}, u_{k-2}, \dots, u_{-m+M}).$$

(We include in the information vector the control inputs  $u_{-1}, \dots, u_{-m+M}$ . If control starts at time 0, these inputs will be zero.) There are no constraints on  $u_k$ . The problem is to find a policy  $\{\mu_0(I_0), \dots, \mu_{N-1}(I_{N-1})\}$  that minimizes

$$E \left\{ \sum_{k=1}^N (y_k)^2 \right\}.$$

By using state augmentation, we can cast this problem into the framework of the linear-quadratic problem of Section 5.2. The corresponding linear system in state space format involves a state  $x_k$  given by

$$x_k = (y_{k+M-1}, \dots, y_{k+M-m}, u_{k-1}, \dots, u_{k+M-m}, \epsilon_{k+M-1}, \dots, \epsilon_{k+M-m}).$$

Because  $y_{k+M-1}, \dots, y_{k+M-m}$  and  $\epsilon_{k+M-1}, \dots, \epsilon_{k+M-m}$  are unknown to the controller, we are faced with a problem of imperfect state information.

An analysis analogous to the one of Section 5.2 shows that certainty equivalence holds; that is, the *optimal control  $u_k^*$  at time  $k$  given  $I_k$  is the same as the one that would be applied in the deterministic problem where the current state*

$$x_k = (y_{k+M-1}, \dots, y_{k+M-m}, u_{k-1}, \dots, u_{k+M-m}, \epsilon_{k+M-1}, \dots, \epsilon_{k+M-m})$$

*is set equal to its conditional expected value given  $I_k$ , and the future disturbances  $\epsilon_{k+M}, \dots, \epsilon_N$  are set equal to zero (their expected value).*

Thus the optimal control  $u_k^* = \mu_k^*(I_k)$  is obtained by solving for  $u_k$  the equation

$$\begin{aligned} E\{y_{k+M} | u_k, I_k\} &= E\{y_{k+M} | y_k, y_{k-1}, \dots, y_{-m+1}, u_k, u_{k-1}, \dots, u_{-m+M}\} \\ &= 0. \end{aligned}$$

This leads to the problem of calculating  $E\{y_{k+M} | I_k, u_k\}$ , known as the *forecasting* or *prediction* problem, which is important in its own right. We first treat the easier case where there is no delay ( $M = 1$ ) and then discuss the more general case where the delay can be positive.

### Forecasting for ARMAX Models – No Delay ( $M = 1$ )

Assume that  $M = 1$ . We would like to generate an equation for the forecast  $E\{y_{k+1} | I_k, u_k\}$ , and then determine the optimal control  $u_k^* = \mu_k^*(I_k)$  by setting this forecast to zero. Let us introduce an auxiliary sequence  $\{z_k\}$  via the equation

$$z_k = y_k - \epsilon_k.$$

A key fact is that, since  $\{\epsilon_k\}$  is an independent, zero-mean sequence, we have

$$E\{z_{k+1} | I_k, u_k\} = E\{y_{k+1} | I_k, u_k\}.$$

We can thus obtain the desired forecast of  $y_{k+1}$  by forecasting  $z_{k+1}$  instead. We can then obtain the optimal control  $u_k^*$  by setting  $E\{z_{k+1} | I_k, u_k^*\} = 0$ .

By using the definition  $z_k = y_k - \epsilon_k$  to express  $y_k$  in terms of  $z_k$  in the ARMAX model equation for  $M = 1$ , we obtain

$$z_{k+1} + c_1 z_k + \cdots + c_m z_{k-m+1} = b_1 u_k + \cdots + b_m u_{k-m+1} + w_k, \quad (5.20)$$

where

$$w_k = (c_1 - a_1)y_k + \cdots + (c_m - a_m)y_{k-m+1}.$$

We note that  $w_k$  is perfectly observable by the controller; however, the scalars  $z_0, \dots, z_{k-m+1}$  are not known to the controller because the initial conditions  $z_0, \dots, z_{1-m}$  of the system (5.20) are unknown. Nonetheless, the system (5.20) is stable, since the roots of the polynomial  $C(s)$  have been assumed to be outside the unit circle. As a result, the initial conditions do not matter asymptotically. In other words, if we generate a sequence  $\{\hat{y}_k\}$  using the system (5.20) and zero initial conditions, i.e.,

$$\hat{y}_{k+1} + c_1 \hat{y}_k + \cdots + c_m \hat{y}_{k-m+1} = b_1 u_k + \cdots + b_m u_{k-m+1} + w_k,$$

with

$$\hat{y}_0 = 0, \quad \hat{y}_{-1} = 0, \quad \dots \quad \hat{y}_{1-m} = 0,$$

then we will have

$$\lim_{k \rightarrow \infty} (\hat{y}_k - z_k) = 0.$$

Thus,  $\hat{y}_{k+1}$  is an asymptotically accurate approximation to the optimal forecast  $E\{y_{k+1} | I_k, u_k\}$ .

### Minimum Variance Control: Imperfect State Information and No Delay

Based on the earlier discussion, an asymptotically accurate approximation to the minimum variance policy is obtained by setting  $u_k$  to the value that makes  $\hat{y}_{k+1} = 0$ ; that is, by solving for  $u_k$  the equation

$$\hat{y}_{k+1} + c_1 \hat{y}_k + \cdots + c_m \hat{y}_{k-m+1} = b_1 u_k + \cdots + b_m u_{k-m+1} + w_k.$$

If this policy is followed, however, the earlier forecasts  $\hat{y}_k, \dots, \hat{y}_{k-m+1}$  will be equal to zero. Thus the (approximate) minimum variance policy is given by

$$\begin{aligned} u_k &= \frac{1}{b_1} (w_k - b_2 u_{k-1} - \cdots - b_m u_{k-m+1}) \\ &= \frac{1}{b_1} ((a_1 - c_1)y_k + \cdots + (a_m - c_m)y_{k-m+1} \\ &\quad - b_2 u_{k-1} - \cdots - b_m u_{k-m+1}). \end{aligned}$$

By substituting this policy in the ARMAX model

$$\begin{aligned} y_{k+1} + a_1 y_k + \cdots + a_m y_{k-m+1} &= b_1 u_k + \cdots + b_m u_{k-m+1} \\ &\quad + \epsilon_{k+1} + c_1 \epsilon_k + \cdots + c_m \epsilon_{k-m+1}, \end{aligned}$$

we see that the closed-loop system becomes

$$y_{k+1} - \epsilon_{k+1} + c_1(y_k - \epsilon_k) + \cdots + c_m(y_{k-m+1} - \epsilon_{k-m+1}) = 0,$$

or equivalently  $C(s)(y_k - \epsilon_k) = 0$ . Since  $C(s)$  has its roots outside the unit circle, this is a stable system, and we have

$$y_k = \epsilon_k + \gamma(k),$$

where  $\gamma(k) \rightarrow 0$  as  $k \rightarrow \infty$ .

### Forecasting: The General Case

Consider now the general case where the delay  $M$  can be greater than 1. The forecasting problem can still be nicely solved by using a certain trick to transform the ARMAX equation into a more convenient form. To this end, we first obtain polynomials  $F(s)$  and  $G(s)$  of the form

$$F(s) = 1 + f_1 s + \cdots + f_{M-1} s^{M-1},$$

$$G(s) = g_0 + g_1 s + \cdots + g_{m-1} s^{m-1},$$

which satisfy

$$C(s) = A(s)F(s) + s^M G(s). \quad (5.21)$$

The coefficients of  $F(s)$  and  $G(s)$  are uniquely determined from those of  $C(s)$  and  $A(s)$  by equating coefficients of both sides of the relation

$$\begin{aligned} 1 + c_1 s + \cdots + c_m s^m &= (1 + a_1 s + \cdots + a_m s^m)(1 + f_1 s + \cdots + f_{M-1} s^{M-1}) \\ &\quad + s^M(g_0 + g_1 s + \cdots + g_{m-1} s^{m-1}). \end{aligned}$$

**Example 5.3.2**

Let  $m = 3$  and  $M = 2$ . Then the preceding equation takes the form

$$1 + c_1s + c_2s^2 + c_3s^3 = (1 + a_1s + a_2s^2 + a_3s^3)(1 + f_1s) + s^2(g_0 + g_1s + g_2s^2),$$

and by equating coefficients we have

$$c_1 = a_1 + f_1, \quad c_2 = a_2 + a_1f_1 + g_0, \quad c_3 = a_3 + a_2f_1 + g_1, \quad a_3f_1 + g_2 = 0,$$

from which  $f_1$ ,  $g_0$ ,  $g_1$ , and  $g_2$  are uniquely determined.

The ARMAX model can be written as

$$A(s)y_{k+M} = \bar{B}(s)u_k + C(s)\epsilon_{k+M}, \quad (5.22)$$

where

$$\bar{B}(s) = s^{-M}B(s) = b_M + b_{M+1}s + \cdots + b_ms^{m-M}.$$

Multiplying both sides of Eq. (5.22) with  $F(s)$ , we have

$$F(s)A(s)y_{k+M} = F(s)\bar{B}(s)u_k + F(s)C(s)\epsilon_{k+M},$$

and using Eq. (5.21) to express  $F(s)A(s)$  as  $C(s) - s^M G(s)$ , we obtain

$$(C(s) - s^M G(s))y_{k+M} = F(s)\bar{B}(s)u_k + F(s)C(s)\epsilon_{k+M},$$

or equivalently

$$C(s)(y_{k+M} - F(s)\epsilon_{k+M}) = F(s)\bar{B}(s)u_k + G(s)y_k. \quad (5.23)$$

Let us now introduce the auxiliary sequence  $\{z_k\}$  via the equation

$$z_{k+M} = y_{k+M} - F(s)\epsilon_{k+M} = y_{k+M} - \epsilon_{k+M} - f_1\epsilon_{k+M-1} - \cdots - f_{M-1}\epsilon_{k+1}.$$

Note that when  $M = 1$ , we have  $F(s) = 1$  and  $z_k = y_k - \epsilon_k$ , so  $\{z_k\}$  is the same sequence as the one introduced earlier for the case of no delay. Again, since  $\{\epsilon_k\}$  is an independent, zero-mean sequence, by taking expectations in the definition  $z_{k+M} = y_{k+M} - F(s)\epsilon_{k+M}$ , we obtain

$$E\{z_{k+M} | I_k, u_k\} = E\{y_{k+M} | I_k, u_k\},$$

and we can obtain the desired forecast of  $y_{k+M}$  by forecasting  $z_{k+M}$  in its place. Furthermore, by Eq. (5.23),  $z_{k+M}$  is written as

$$C(s)z_{k+M} = w_k$$

or

$$z_{k+M} + c_1z_{k+M-1} + \cdots + c_mz_{k+M-m} = w_k, \quad (5.24)$$

where

$$w_k = F(s)\bar{B}(s)u_k + G(s)y_k. \quad (5.25)$$

Since the scalar  $w_k$  of Eq. (5.25) is available at time  $k$  (i.e., it is determined from  $I_k$  and  $u_k$ ), the system (5.24) can serve as a basis for forecasting  $z_{k+M}$ . We would be able to predict exactly  $z_{k+M}$  and use it as a forecast of  $y_{k+M}$  if we knew appropriate initial conditions with which to start the equation (5.24) that generates it. We don't know such initial conditions, but because this equation represents a stable system, the choice of initial conditions does not matter asymptotically, as we proceed to explain more formally.

We consider the sequence  $\hat{y}_{k+M}$  generated by

$$\hat{y}_{k+M} + c_1\hat{y}_{k+M-1} + \cdots + c_m\hat{y}_{k+M-m} = w_k$$

with initial condition

$$\hat{y}_{M-1} = \hat{y}_{M-2} = \cdots = \hat{y}_{M-m} = 0, \quad (5.26)$$

and we claim that the forecast  $E\{z_{k+M} | I_k\}$  can be approximated by  $\hat{y}_{k+M}$ . To see this, note that from Eqs. (5.24) to (5.26) we have

$$z_{k+M} = \hat{y}_{k+M} + (\gamma_1(k)z_{M-1} + \cdots + \gamma_m(k)z_{M-m})$$

and

$$E\{z_{k+M} | I_k, u_k\} = \hat{y}_{k+M} + \sum_{i=1}^m \gamma_i(k)E\{z_{M-i} | I_k, u_k\},$$

where  $\gamma_1(k), \dots, \gamma_m(k)$  are appropriate scalars depending on  $k$ . Since  $C(s)$  has all its roots outside the unit circle, we have (compare with the discussion on stability earlier in this section)

$$\lim_{k \rightarrow \infty} \gamma_1(k) = \lim_{k \rightarrow \infty} \gamma_2(k) = \cdots = \lim_{k \rightarrow \infty} \gamma_m(k) = 0.$$

It follows that, for large values of  $k$ ,

$$\hat{y}_{k+M} \approx E\{z_{k+M} | I_k, u_k\} = E\{y_{k+M} | I_k, u_k\}.$$

(More precisely, we have  $|\hat{y}_{k+M} - E\{y_{k+M} | I_k, u_k\}| \rightarrow 0$  as  $k \rightarrow \infty$ , where the convergence is in the mean-square sense.)

In conclusion, an asymptotically accurate approximation to the optimal forecast  $E\{y_{k+M} | I_k, u_k\}$  is given by  $\hat{y}_{k+M}$  and is generated by the equation

$$\hat{y}_{k+M} + c_1\hat{y}_{k+M-1} + \cdots + c_m\hat{y}_{k+M-m} = F(s)\bar{B}(s)u_k + G(s)y_k \quad (5.27)$$

with the initial condition

$$\hat{y}_{M-1} = \hat{y}_{M-2} = \cdots = \hat{y}_{M-m} = 0. \quad (5.28)$$

### Minimum Variance Control: The General Case

Based on the earlier discussion, the minimum variance policy is obtained by solving for  $u_k$  the equation  $E\{y_{k+M} | I_k, u_k\} = 0$ . Thus an asymptotically accurate approximation is obtained by setting  $u_k$  to the value that makes  $\hat{y}_{k+M} = 0$ , that is, by solving for  $u_k$  the equation [cf. Eqs. (5.27) and (5.28)]

$$F(s)\bar{B}(s)u_k + G(s)y_k = c_1\hat{y}_{k+M-1} + \cdots + c_m\hat{y}_{k+M-m}.$$

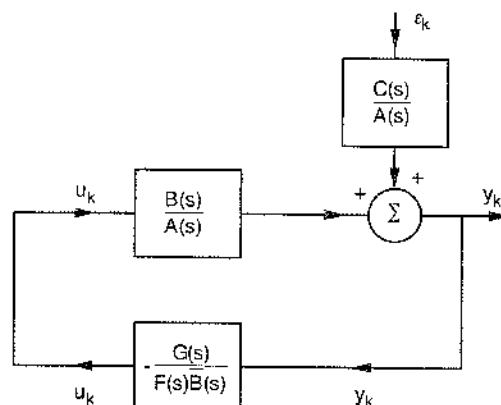
If this policy is followed, however, the earlier forecasts  $\hat{y}_{k+M-1}, \dots, \hat{y}_{k+M-m}$  will be equal to zero. Thus the (approximate) minimum variance policy is given by

$$F(s)\bar{B}(s)u_k - G(s)y_k = 0; \quad (5.29)$$

that is,  $u_k^*$  is generated by passing  $y_k$  through the linear filter

$$-G(s)/F(s)\bar{B}(s),$$

as shown in Fig. 5.3.2.



**Figure 5.3.2** Minimum variance control with imperfect state information. Structure of the optimal closed-loop system.

From Eqs. (5.23) and (5.29), we obtain the equation for the closed-loop system

$$C(s)(y_{k+M} - F(s)\epsilon_{k+M}) = 0.$$

Since  $C(s)$  has its roots outside the unit circle, we obtain

$$y_{k+M} = F(s)\epsilon_{k+M} + \gamma(k),$$

where  $\gamma(k) \rightarrow 0$  as  $k \rightarrow \infty$ . So asymptotically, the closed-loop system takes the form

$$y_k = c_k + f_1\epsilon_{k-1} + \cdots + f_{M-1}\epsilon_{k-M+1}.$$

Let us consider now the stability properties of the closed-loop system when the true system parameters differ slightly from those of the assumed model. Let the true system be described by

$$A^0(s)y_k = s^M\bar{B}^0(s)u_k + C^0(s)c_k, \quad (5.30)$$

while  $u_k$  is given by the minimum variance policy

$$F(s)\bar{B}(s)u_k + G(s)y_k = 0, \quad (5.31)$$

where

$$C(s) = A(s)F(s) + s^M G(s).$$

Operating on Eq. (5.30) with  $F(s)\bar{B}(s)$  and using Eq. (5.31), we obtain

$$F(s)\bar{B}(s)A^0(s)y_k = -s^M\bar{B}^0(s)G(s)y_k + F(s)\bar{B}(s)C^0(s)\epsilon_k.$$

Combining the last two equations and collecting terms, we have

$$\{F(s)\bar{B}(s)A^0(s) + (C(s) - A(s)F(s))\bar{B}^0(s)\}y_k = F(s)\bar{B}(s)C^0(s)\epsilon_k$$

or

$$\{\bar{B}^0(s)C(s) + F(s)(\bar{B}(s)A^0(s) - A(s)\bar{B}^0(s))\}y_k = F(s)\bar{B}(s)C^0(s)\epsilon_k.$$

If the coefficients of  $A^0(s)$ ,  $\bar{B}^0(s)$ , and  $C^0(s)$  are near those of  $A(s)$ ,  $\bar{B}(s)$ , and  $C(s)$ , then the poles of the closed-loop system will be near the roots of  $\bar{B}(s)C(s)$ . Thus the closed-loop system will be in effect stable only if the roots of  $\bar{B}(s)$  are strictly outside the unit circle, similar to the perfect state information case examined earlier.

### 5.4 SUFFICIENT STATISTICS

The main difficulty with the DP algorithm for imperfect state information problems is that it is carried out over a state space of expanding dimension. As a new measurement is added at each stage  $k$ , the dimension of the state (the information vector  $I_k$ ) increases accordingly. This motivates an effort to reduce the data that are truly necessary for control purposes. In other words, it is of interest to look for quantities known as *sufficient statistics*,

which ideally would be of smaller dimension than  $I_k$  and yet summarize all the essential content of  $I_k$  as far as control is concerned.

Consider the DP algorithm (5.4) and (5.5), restated here for convenience:

$$\begin{aligned} J_{N-1}(I_{N-1}) &= \min_{u_{N-1} \in U_{N-1}} \left[ E_{x_{N-1}, w_{N-1}} \{ g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) \right. \\ &\quad \left. + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \mid I_{N-1}, u_{N-1} \} \right], \end{aligned} \quad (5.32)$$

$$J_k(I_k) = \min_{u_k \in U_k} \left[ E_{x_k, w_k, z_{k+1}} \{ g_k(x_k, u_k, w_k) + J_{k+1}(I_k, z_{k+1}, u_k) \mid I_k, u_k \} \right]. \quad (5.33)$$

Suppose that we can find a function  $S_k(I_k)$  of the information vector, such that a minimizing control in Eqs. (5.32) and (5.33) depends on  $I_k$  via  $S_k(I_k)$ . By this we mean that the minimization in the right-hand side of the DP algorithm (5.32) and (5.33) can be written in terms of some function  $H_k$  as

$$\min_{u_k \in U_k} H_k(S_k(I_k), u_k).$$

Such a function  $S_k$  is called a *sufficient statistic*. Its salient feature is that an optimal policy obtained by the preceding minimization can be written as

$$\mu_k^*(I_k) = \bar{\mu}_k(S_k(I_k)),$$

where  $\bar{\mu}_k$  is an appropriate function. Thus, if the sufficient statistic is characterized by a set of fewer numbers than the information vector  $I_k$ , it may be easier to implement the policy in the form  $u_k = \bar{\mu}_k(S_k(I_k))$  and take advantage of the resulting data reduction.

#### 5.4.1 The Conditional State Distribution

There are many different functions that can serve as sufficient statistics. The identity function  $S_k(I_k) = I_k$  is certainly one of them. In this section, we will derive another important sufficient statistic: the conditional probability distribution  $P_{x_k|I_k}$  of the state  $x_k$ , given the information vector  $I_k$ . An extra assumption is necessary for this, namely that the *probability distribution of the observation disturbance  $v_{k+1}$  depends explicitly only on the immediately preceding state, control, and system disturbance  $x_k, u_k, w_k$ , and not on  $x_{k-1}, \dots, x_0, u_{k-1}, \dots, u_0, w_{k-1}, \dots, w_0, v_{k-1}, \dots, v_0$* . Under this assumption, we will show that for all  $k$  and  $I_k$ , we have

$$J_k(I_k) = \min_{u_k \in U_k} H_k(P_{x_k|I_k}, u_k) = \bar{J}_k(P_{x_k|I_k}), \quad (5.34)$$

where  $H_k$  and  $\bar{J}_k$  are appropriate functions.

To this end, we will use an important fact that relates to state estimation of discrete-time stochastic systems: the conditional distribution  $P_{x_k|I_k}$  can be generated recursively by an equation of the form

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1}), \quad (5.35)$$

where  $\Phi_k$  is some function that can be determined from the data of the problem. Let us postpone a justification of this for the moment, and accept it for the purpose of the following discussion.

We note that to perform the minimization in Eq. (5.32), it is sufficient to know the distribution  $P_{x_{N-1}|I_{N-1}}$  together with the distribution  $P_{w_{N-1}|x_{N-1}, u_{N-1}}$ , which is part of the problem data. Thus, the minimization in the right-hand side of Eq. (5.32) is of the form

$$J_{N-1}(I_{N-1}) = \min_{u_{N-1} \in U_{N-1}} H_{N-1}(P_{x_{N-1}|I_{N-1}}, u_{N-1}) = \bar{J}_{N-1}(P_{x_{N-1}|I_{N-1}})$$

for appropriate functions  $H_{N-1}$  and  $\bar{J}_{N-1}$ .

We now use induction, i.e., we assume that

$$J_{k+1}(I_{k+1}) = \min_{u_{k+1} \in U_{k+1}} H_{k+1}(P_{x_{k+1}|I_{k+1}}, u_{k+1}) = \bar{J}_{k+1}(P_{x_{k+1}|I_{k+1}}), \quad (5.36)$$

for appropriate functions  $H_{k+1}$  and  $\bar{J}_{k+1}$ , and we show that

$$J_k(I_k) = \min_{u_k \in U_k} H_k(P_{x_k|I_k}, u_k) = \bar{J}_k(P_{x_k|I_k}), \quad (5.37)$$

for appropriate functions  $H_k$  and  $\bar{J}_k$ .

Indeed, using Eqs. (5.35) and (5.36), the DP equation (5.33) is written as

$$J_k(I_k) = \min_{u_k \in U_k} E \left\{ g_k(x_k, u_k, w_k) + \bar{J}_{k+1}(\Phi_k(P_{x_k|I_k}, u_k, z_{k+1})) \mid I_k, u_k \right\}. \quad (5.38)$$

In order to calculate the expression being minimized over  $u_k$  above, we need, in addition to  $P_{x_k|I_k}$ , the joint distribution

$$P(x_k, w_k, z_{k+1} \mid I_k, u_k)$$

or, equivalently,

$$P(x_k, w_k, h_{k+1}(f_k(x_k, u_k, w_k), u_k, v_{k+1}) \mid I_k, u_k).$$

This distribution can be expressed in terms of  $P_{x_k|I_k}$ , the given distributions

$$P(w_k \mid x_k, u_k), \quad P(v_{k+1} \mid f_k(x_k, u_k, w_k), u_k, w_k),$$

and the system equation  $x_{k+1} = f_k(x_k, u_k, w_k)$ . Therefore the expression minimized over  $u_k$  in Eq. (5.38) can be written as a function of  $P_{x_k|I_k}$  and  $u_k$ , and the DP equation (5.33) can be written as

$$J_k(I_k) = \min_{u_k \in U_k} H_k(P_{x_k|I_k}, u_k)$$

for a suitable function  $H_k$ . Thus the induction is complete and it follows that the distribution  $P_{x_k|I_k}$  is a sufficient statistic.

Note that if the conditional distribution  $P_{x_k|I_k}$  is uniquely determined by another expression  $S_k(I_k)$ , i.e.,

$$P_{x_k|I_k} = G_k(S_k(I_k))$$

for an appropriate function  $G_k$ , then  $S_k(I_k)$  is also a sufficient statistic. Thus, for example, if we can show that  $P_{x_k|I_k}$  is a Gaussian distribution, then the mean and the covariance matrix corresponding to  $P_{x_k|I_k}$  form a sufficient statistic.

Regardless of its computational value, the representation of the optimal policy as a sequence of functions of the conditional probability distribution  $P_{x_k|I_k}$ ,

$$\mu_k(I_k) = \bar{\mu}_k(P_{x_k|I_k}), \quad k = 0, 1, \dots, N-1,$$

is conceptually very useful. It provides a decomposition of the optimal controller in two parts:

- (a) An *estimator*, which uses at time  $k$  the measurement  $z_k$  and the control  $u_{k-1}$  to generate the probability distribution  $P_{x_k|I_k}$ .
- (b) An *actuator*, which generates a control input to the system as a function of the probability distribution  $P_{x_k|I_k}$  (Fig. 5.4.1).

This interpretation has formed the basis for various suboptimal control schemes that separate the controller a priori into an estimator and an actuator and attempt to design each part in a manner that seems "reasonable." Schemes of this type will be discussed in Chapter 6.

### The Conditional State Distribution Recursion

There remains to justify the recursion

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1}). \quad (5.39)$$

Let us first give an example.

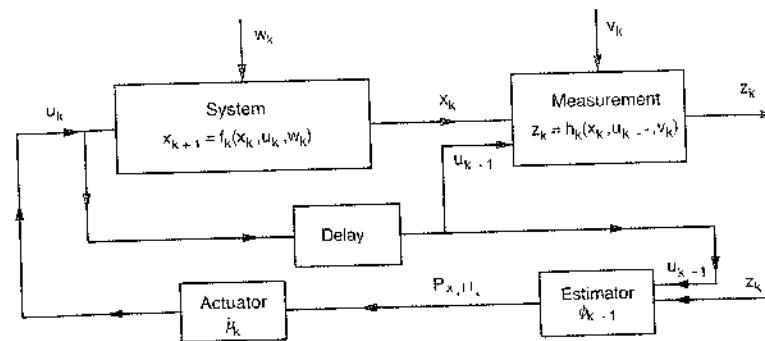


Figure 5.4.1 Conceptual separation of the optimal controller into an estimator and an actuator.

### Example 5.4.1 (Search Problem)

In a classical problem of search, one has to decide at each period whether to search a site that may contain a treasure. If a treasure is present, the search reveals it with probability  $\beta$ , in which case the treasure is removed from the site. Here the state has two values: either a treasure is present in the site or it is not. The control  $u_k$  takes two values: search and not search. If the site is searched, the observation  $z_{k+1}$  takes two values, treasure found or not found, while if the site is not searched, the value of  $z_{k+1}$  is irrelevant.

Denote

$p_k$ : probability a treasure is present at the beginning of period  $k$ .

This probability evolves according to the equation

$$p_{k+1} = \begin{cases} p_k & \text{if the site is not searched at time } k, \\ 0 & \text{if the site is searched and a treasure is found,} \\ \frac{p_k(1-\beta)}{p_k(1-\beta)+1-p_k} & \text{if the site is searched but no treasure is found.} \end{cases}$$

The second relation holds because the treasure is removed after a successful search. The third relation follows by application of Bayes' rule ( $p_{k+1}$  is equal to the  $k$ th period probability of a treasure being present and the search being unsuccessful, divided by the probability of an unsuccessful search). The preceding equation defines the desired recursion for the conditional distribution of the state and is a special case of Eq. (5.39).

The general form of the recursion

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1})$$

is developed in Exercise 5.7 for the case where the state, control, observation, and disturbance spaces are finite sets. In the case where these spaces

are the real line and all random variables involved possess probability density functions, the conditional density  $p(x_{k+1} | I_{k+1})$  is generated from  $p(x_k | I_k)$ ,  $u_k$ , and  $z_{k+1}$  by means of the equation

$$\begin{aligned} p(x_{k+1} | I_{k+1}) &= p(x_{k+1} | I_k, u_k, z_{k+1}) \\ &= \frac{p(x_{k+1}, z_{k+1} | I_k, u_k)}{p(z_{k+1} | I_k, u_k)} \\ &= \frac{p(x_{k+1} | I_k, u_k)p(z_{k+1} | I_k, u_k, x_{k+1})}{\int_{-\infty}^{\infty} p(x_{k+1} | I_k, u_k)p(z_{k+1} | I_k, u_k, x_{k+1})dx_{k+1}}. \end{aligned}$$

In this equation all the probability densities appearing in the right-hand side may be expressed in terms of  $p(x_k | I_k)$ ,  $u_k$ , and  $z_{k+1}$ . In particular, the density  $p(x_{k+1} | I_k, u_k)$  may be expressed through  $p(x_k | I_k)$ ,  $u_k$ , and the system equation  $x_{k+1} = f_k(x_k, u_k, w_k)$  using the given density  $p(w_k | x_k, u_k)$  and the relation

$$p(w_k | I_k, u_k) = \int_{-\infty}^{\infty} p(x_k | I_k)p(w_k | x_k, u_k)dx_k.$$

Similarly, the density  $p(z_{k+1} | I_k, u_k, x_{k+1})$  is expressed through the measurement equation  $z_{k+1} = h_{k+1}(x_{k+1}, u_k, v_{k+1})$  using the densities

$$p(x_k | I_k), \quad p(w_k | x_k, u_k), \quad p(v_{k+1} | x_k, u_k, w_k).$$

By substituting these expressions in the equation for  $p(x_{k+1} | I_{k+1})$ , we obtain a dynamic system equation for the conditional state distribution of the desired form. Other similar examples will be given in subsequent sections. A mathematically rigorous substantiation of the recursion  $P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1})$  can be found in Bertsekas and Shreve [BeS78].

### Alternative Perfect State Information Reduction

Finally, let us formally rewrite the DP algorithm in terms of the sufficient statistic  $P_{x_k|I_k}$ . Using Eqs. (5.35), (5.37), and (5.38), we have for  $k < N - 1$

$$\begin{aligned} \bar{J}_k(P_{x_k|I_k}) &= \min_{u_k \in U_k} \left[ \mathbb{E}_{x_k, w_k, z_{k+1}} \left\{ g_k(x_k, u_k, w_k) \right. \right. \\ &\quad \left. \left. + \bar{J}_{k+1}(\Phi_k(P_{x_k|I_k}, u_k, z_{k+1})) \mid I_k, u_k \right\} \right]. \end{aligned} \quad (5.40)$$

In the case where  $k = N - 1$ , we have

$$\begin{aligned} \bar{J}_{N-1}(P_{x_{N-1}|I_{N-1}}) &= \min_{u_{N-1} \in U_{N-1}} \left[ \mathbb{E}_{x_{N-1}, w_{N-1}} \left\{ g_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) \right. \right. \\ &\quad \left. \left. + g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) \mid I_{N-1}, u_{N-1} \right\} \right]. \end{aligned} \quad (5.41)$$

This DP algorithm yields the optimal cost as

$$J^* = \mathbb{E}_{z_0} \{ \bar{J}_0(P_{x_0|z_0}) \},$$

where  $\bar{J}_0$  is obtained by the last step, and the probability distribution of  $z_0$  is obtained from the measurement equation  $z_0 = h_0(x_0, v_0)$  and the distributions of  $x_0$  and  $v_0$ .

By observing the form of Eq. (5.40), we note that it has the standard DP structure, except that  $P_{x_k|I_k}$  plays the role of the "state." Indeed the role of the "system" is played by the recursive estimator of  $P_{x_k|I_k}$ ,

$$P_{x_{k+1}|I_{k+1}} = \Phi_k(P_{x_k|I_k}, u_k, z_{k+1}),$$

and this system fits the framework of the basic problem (the role of control is played by  $u_k$  and the role of the disturbance is played by  $z_{k+1}$ ). Furthermore, the controller can calculate (at least in principle) the state  $P_{x_k|I_k}$  of this system at time  $k$ , so perfect state information prevails. Thus the alternate DP algorithm (5.40)-(5.41) may be viewed as the DP algorithm of the perfect state information problem that involves the above system, whose state is  $P_{x_k|I_k}$ , and an appropriately reformulated cost function. In the absence of perfect knowledge of the state, the controller can be viewed as controlling the "probabilistic state"  $P_{x_k|I_k}$  so as to minimize the expected cost-to-go conditioned on the information  $I_k$  available.

### Example 5.4.1 (Continued)

Let us write the DP algorithm (5.40) for the search problem of Example 5.4.1, assuming that the treasure's worth is  $V$ , that each search costs  $C$ , and that once we decide not to search at a particular time, then we cannot search at future times. The algorithm takes the form

$$\bar{J}_k(p_k) = \max \left[ 0, -C + p_k \beta V + (1 - p_k \beta) \bar{J}_{k+1} \left( \frac{p_k(1 - \beta)}{p_k(1 - \beta) + 1 - p_k} \right) \right],$$

with  $\bar{J}_N(p_N) = 0$ . From this algorithm, it is straightforward to show by induction that the functions  $\bar{J}_k$  satisfy  $\bar{J}_k(p_k) = 0$  for all  $p_k \leq C/(\beta V)$ , and that it is optimal to search at period  $k$  if and only if

$$C \leq p_k \beta V.$$

Thus, it is optimal to search if and only if the expected reward from the next search is greater or equal to the cost of the search.

### 5.4.2 Finite-State Systems

We will now consider systems that are stationary finite-state Markov chains, in which case the conditional probability distribution  $P_{x_k|u_k}$  is characterized by a finite set of numbers. The states are denoted  $1, 2, \dots, n$ . When a control  $u$  is applied, the system moves from state  $i$  to state  $j$  with probability  $p_{ij}(u)$ . The control  $u$  is chosen from a finite set  $U$ . Following a state transition, an observation is made by the controller. There is a finite number of possible observation outcomes, and the probability of each depends on the current state and the preceding control. The information available to the controller at stage  $k$  is the information vector

$$I_k = (z_1, \dots, z_k, u_0, \dots, u_{k-1}),$$

where for all  $i$ ,  $z_i$  and  $u_i$  are the observation and control at stage  $i$ , respectively. Following the observation  $z_k$ , a control  $u_k$  is chosen by the controller, and a cost  $g(x_k, u_k)$  is incurred, where  $x_k$  is the current (hidden) state. The terminal cost for being at state  $x$  at the end of the  $N$  stages is denoted  $G(x)$ . We wish to minimize the expected value of the sum of costs incurred over the  $N$  stages.

As discussed in Section 5.4.1, one can reformulate the problem into a problem of perfect state information: the objective is to control the column vector of conditional probabilities

$$p_k = (p_k^1, \dots, p_k^n)',$$

where

$$p_k^i = P(x_k = i | I_k), \quad i = 1, \dots, n.$$

We refer to  $p_k$  as the *belief state*. It evolves according to an equation of the form

$$p_{k+1} = \Phi(p_k, u_k, z_{k+1}).$$

where the function  $\Phi$  represents an estimator, as discussed in Section 5.4.1. The initial belief state  $p_0$  is given.

The corresponding DP algorithm [see Eqs. (5.40) and (5.41)] has the form

$$\bar{J}_k(p_k) = \min_{u_k \in U} \left[ p_k' g(u_k) + E_{z_{k+1}} \left\{ \bar{J}_{k+1}(\Phi(p_k, u_k, z_{k+1})) \mid p_k, u_k \right\} \right], \quad (5.42)$$

where  $g(u_k)$  is the column vector with components  $g(1, u_k), \dots, g(n, u_k)$ , and  $p_k' g(u_k)$ , the expected stage cost, is the inner product of the vectors  $p_k$  and  $g(u_k)$ . The algorithm starts at stage  $N$ , with

$$\bar{J}_N(p_N) = p_N' G,$$

where  $G$  is the column vector with components the terminal costs  $G(i)$ ,  $i = 1, \dots, n$ , and proceeds backwards. Note that in this DP algorithm, the conditional distribution of  $z_{k+1}$  given  $p_k$  and  $u_k$  can be computed using the transition probabilities  $p_{ij}(u)$ , and the known conditional distribution of  $z_{k+1}$  given  $x_{k+1}$  and  $u_k$ . In particular, we have for any possible observation value  $z$ ,

$$P(z_{k+1} = z \mid p_k, u_k) = \sum_{i=1}^n p_k^i \sum_{j=1}^n p_{ij}(u_k) P(z_{k+1} = z \mid x_{k+1} = j, u_k).$$

It turns out that the cost-to-go functions  $\bar{J}_k$  in the DP algorithm are *piecewise linear* and *concave*. The demonstration of this fact is straightforward, but tedious, and is outlined in Exercise 5.7. A consequence of the piecewise linearity property is that  $\bar{J}_k$  can be characterized by a finite set of scalars. Still, however, for fixed  $k$ , the number of these scalars can increase fast with  $N$ , and there may be no computationally efficient way to solve the problem (see Papadimitriou and Tsitsiklis [PaT87]). We will not discuss here any special procedures for computing  $\bar{J}_k$  (see Lovejoy [Lov91a], [Lov91b], and Smallwood and Sondik [SmS73], [Son71]). Instead we will demonstrate the DP algorithm by means of examples.

#### Example 5.4.2 (Machine Repair)

In the two-state machine repair example of Section 5.1, let us denote

$$p_1 = P(x_1 = \bar{P} \mid I_1), \quad p_0 = P(x_0 = \bar{P} \mid I_0).$$

The equation relating  $p_1, p_0, u_0, z_1$  is written as

$$p_1 = \Phi_0(p_0, u_0, z_1).$$

One may verify by straightforward calculation that  $\Phi_0$  is given by

$$p_1 = \Phi_0(p_0, u_0, z_1) = \begin{cases} \frac{1}{7} & \text{if } u_0 = S, \quad z_1 = G, \\ \frac{3}{7} & \text{if } u_0 = S, \quad z_1 = B, \\ \frac{1+2p_0}{7-4p_0} & \text{if } u_0 = C, \quad z_1 = G, \\ \frac{3+6p_0}{7+4p_0} & \text{if } u_0 = C, \quad z_1 = B. \end{cases}$$

The DP algorithm (5.42) may be written in terms of  $p_0, p_1$ , and  $\Phi_0$  above as

$$\bar{J}_1(p_1) = \min[2p_1, 1],$$

$$\begin{aligned} \bar{J}_0(p_0) = \min & \left[ 2p_0 + P(z_1 = G \mid p_0, C) \bar{J}_1(\Phi_0(p_0, C, G)) \right. \\ & + P(z_1 = B \mid p_0, C) \bar{J}_1(\Phi_0(p_0, C, B)), \\ & 1 + P(z_1 = G \mid p_0, S) \bar{J}_1(\Phi_0(p_0, S, G)) \\ & \left. + P(z_1 = B \mid p_0, S) \bar{J}_1(\Phi_0(p_0, S, B)) \right]. \end{aligned}$$

The probabilities entering in the second equation may be expressed in terms of  $p_0$  by straightforward calculation as

$$\begin{aligned} P(z_1 = G \mid p_0, C) &= \frac{7 - 4p_0}{12}, & P(z_1 = B \mid p_0, C) &= \frac{5 + 4p_0}{12}, \\ P(z_1 = G \mid p_0, S) &= \frac{7}{12}, & P(z_1 = B \mid p_0, S) &= \frac{5}{12}. \end{aligned}$$

Using these values we have

$$\begin{aligned} \bar{J}_0(p_0) &= \min \left[ 2p_0 + \frac{7 - 4p_0}{12} \bar{J}_1 \left( \frac{1 + 2p_0}{7 - 4p_0} \right) + \frac{5 + 4p_0}{12} \bar{J}_1 \left( \frac{3 + 6p_0}{5 + 4p_0} \right), \right. \\ &\quad \left. 1 + \frac{7}{12} \bar{J}_1 \left( \frac{1}{7} \right) + \frac{5}{12} \bar{J}_1 \left( \frac{3}{5} \right) \right]. \end{aligned}$$

By minimization in the equation defining  $\bar{J}_1(p_1)$ , we obtain an optimal policy for the last stage

$$\bar{\mu}_1^*(p_1) = \begin{cases} C & \text{if } p_1 \leq \frac{1}{2}, \\ S & \text{if } p_1 > \frac{1}{2}. \end{cases}$$

Also by substitution of  $\bar{J}_1(p_1)$  and by carrying out the straightforward calculation, we obtain

$$\bar{J}_0(p_0) = \begin{cases} \frac{19}{12} & \text{if } \frac{3}{8} \leq p_0 \leq 1, \\ \frac{7+32p_0}{12} & \text{if } 0 \leq p_0 \leq \frac{3}{8}, \end{cases}$$

and an optimal policy for the first stage:

$$\bar{\mu}_0^*(p_0) = \begin{cases} C & \text{if } p_0 \leq \frac{3}{8}, \\ S & \text{if } p_0 > \frac{3}{8}. \end{cases}$$

Note that

$$\begin{aligned} P(x_0 = \bar{P} \mid z_0 = G) &= \frac{1}{7}, & P(x_0 = \bar{P} \mid z_0 = B) &= \frac{3}{5}, \\ P(z_0 = G) &= \frac{7}{12}, & P(z_0 = B) &= \frac{5}{12}, \end{aligned}$$

so that the formula

$$J^* = E\{\bar{J}_0(P_{z_0 \mid z_0})\} = \frac{7}{12} \bar{J}_0\left(\frac{1}{7}\right) + \frac{5}{12} \bar{J}_0\left(\frac{3}{5}\right) = \frac{176}{144}$$

yields the same optimal cost as the one obtained in Section 5.1 by means of the DP algorithm (5.4) and (5.5).

### Example 5.4.3 (A Problem of Instruction)

Consider a problem of instruction where the objective is to teach a student a certain simple item. At the beginning of each period, the student may be in one of two possible states:

$L$ : Item learned.

$\bar{L}$ : Item not learned.

At the beginning of each period, the instructor must make one of two decisions:

$T$ : Terminate the instruction.

$\bar{T}$ : Continue the instruction for one period and then conduct a test that indicates whether the student has learned the item.

The test has two possible outcomes:

$R$ : Student gives a correct answer.

$\bar{R}$ : Student gives an incorrect answer.

The transition probabilities from one state to the next if instruction takes place are given by

$$\begin{aligned} P(x_{k+1} = L \mid x_k = L) &= 1, & P(x_{k+1} = \bar{L} \mid x_k = L) &= 0, \\ P(x_{k+1} = L \mid x_k = \bar{L}) &= t, & P(x_{k+1} = \bar{L} \mid x_k = \bar{L}) &= 1 - t, \end{aligned}$$

where  $t$  is a given scalar with  $0 < t < 1$ .

The outcome of the test depends probabilistically on the state of knowledge of the student as follows:

$$\begin{aligned} P(z_k = R \mid x_k = L) &= 1, & P(z_k = \bar{R} \mid x_k = L) &= 0, \\ P(z_k = R \mid x_k = \bar{L}) &= r, & P(z_k = \bar{R} \mid x_k = \bar{L}) &= 1 - r, \end{aligned}$$

where  $r$  is a given scalar with  $0 < r < 1$ . The probabilistic structure of the problem is illustrated in Fig. 5.4.2.

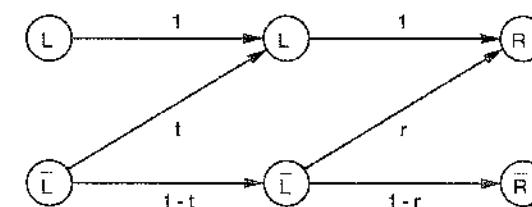


Figure 5.4.2 Probabilistic structure of the instruction problem.

The cost of instruction and testing is  $I$  per period. The cost of terminating instruction is 0 or  $C > 0$  if the student has learned or has not learned the item, respectively. The objective is to find an optimal instruction-termination policy for each period  $k$  as a function of the test information accrued up to that period, assuming that there is a maximum of  $N$  periods of instruction.

It is straightforward to reformulate this problem into the framework of the basic problem with imperfect state information and to conclude that the decision whether to terminate or continue instruction at period  $k$  should depend on the conditional probability that the student has learned the item given the test results so far. This probability is denoted

$$p_k = P(x_k | I_k) = P(x_k = L | z_0, z_1, \dots, z_k).$$

In addition, we can use the DP algorithm (5.40) and (5.41) defined over the space of the sufficient statistic  $p_k$  to obtain an optimal policy. However, rather than proceeding with this elaborate reformulation, we prefer to argue and obtain this DP algorithm directly.

Concerning the evolution of the conditional probability  $p_k$  (assuming instruction occurs), we have by Bayes' rule

$$p_{k+1} = P(x_{k+1} = L | z_0, \dots, z_{k+1}) = \frac{P(x_{k+1} = L, z_{k+1} | z_0, \dots, z_k)}{P(z_{k+1} | z_0, \dots, z_k)},$$

where

$$\begin{aligned} P(z_{k+1} | z_0, \dots, z_k) \\ = P(x_{k+1} = L | z_0, \dots, z_k)P(z_{k+1} | z_0, \dots, z_k, x_{k+1} = L) \\ + P(x_{k+1} = \bar{L} | z_0, \dots, z_k)P(z_{k+1} | z_0, \dots, z_k, x_{k+1} = \bar{L}). \end{aligned}$$

From the probabilistic descriptions given, we have

$$P(z_{k+1} | z_0, \dots, z_k, x_{k+1} = L) = P(z_{k+1} | x_{k+1} = L) = \begin{cases} 1 & \text{if } z_{k+1} = R, \\ 0 & \text{if } z_{k+1} = \bar{R}. \end{cases}$$

$$\begin{aligned} P(z_{k+1} | z_0, \dots, z_k, x_{k+1} = \bar{L}) &= P(z_{k+1} | x_{k+1} = \bar{L}) \\ &= \begin{cases} r & \text{if } z_{k+1} = R, \\ 1 - r & \text{if } z_{k+1} = \bar{R}. \end{cases} \end{aligned}$$

$$P(x_{k+1} = L | z_0, \dots, z_k) = p_k + (1 - p_k)t,$$

$$P(x_{k+1} = \bar{L} | z_0, \dots, z_k) = (1 - p_k)(1 - t).$$

Combining these equations, we obtain

$$p_{k+1} = \Phi(p_k, z_{k+1}) = \begin{cases} \frac{p_k + (1 - p_k)t}{p_k + (1 - p_k)t + (1 - p_k)(1 - t)r} & \text{if } z_{k+1} = R, \\ 0 & \text{if } z_{k+1} = \bar{R}, \end{cases}$$

or equivalently

$$p_{k+1} = \Phi(p_k, z_{k+1}) = \begin{cases} \frac{1 - (1 - t)(1 - p_k)}{1 - (1 - t)(1 - r)(1 - p_k)} & \text{if } z_{k+1} = R, \\ 0 & \text{if } z_{k+1} = \bar{R}. \end{cases} \quad (5.43)$$

This equation is a special case of the general recursive update equation (5.39) for the conditional probability of the state. A cursory examination of Eq. (5.43) shows that, as expected, the conditional probability  $p_{k+1}$  that the student has learned the item increases with every correct answer and drops to zero with every incorrect answer.

We now derive the DP algorithm for the problem. At the end of the  $N$ th period, assuming instruction has continued to that period, the expected cost is

$$\bar{J}_N(p_N) = (1 - p_N)C.$$

At the end of period  $N - 1$ , the instructor has calculated the conditional probability  $p_{N-1}$  that the student has learned the item and wishes to decide whether to terminate instruction and incur an expected cost  $(1 - p_{N-1})C$  or continue the instruction and incur an expected cost  $I + E\{\bar{J}_N(p_N)\}$ . This leads to the following equation for the optimal expected cost-to-go:

$$\bar{J}_{N-1}(p_{N-1}) = \min[(1 - p_{N-1})C, I + (1 - t)(1 - p_{N-1})C].$$

The term  $(1 - p_{N-1})C$  is the cost of terminating instruction, while the term  $(1 - t)(1 - p_{N-1})$  is the probability that the student still has not learned the item following an additional period of instruction.

Similarly, the algorithm is written for every stage  $k$  by replacing  $N$  by  $k + 1$ :

$$\bar{J}_k(p_k) = \min \left[ (1 - p_k)C, I + \underset{z_{k+1}}{E} \left\{ \bar{J}_{k+1}(\Phi(p_k, z_{k+1})) \right\} \right].$$

Now using expression (5.43) for the function  $\Phi$  and the probabilities

$$\begin{aligned} P(z_{k+1} = \bar{R} | p_k) &= (1 - t)(1 - r)(1 - p_k), \\ P(z_{k+1} = R | p_k) &= 1 - (1 - t)(1 - r)(1 - p_k), \end{aligned}$$

we have

$$\bar{J}_k(p_k) = \min[(1 - p_k)C, I + A_k(p_k)], \quad (5.44)$$

where

$$\begin{aligned} A_k(p_k) &= P(z_{k+1} = R | I_k) \bar{J}_{k+1}(\Phi(p_k, R)) \\ &\quad + P(z_{k+1} = \bar{R} | I_k) \bar{J}_{k+1}(\Phi(p_k, \bar{R})) \end{aligned}$$

or, equivalently, using Eq. (5.43),

$$\begin{aligned} A_k(p_k) &= (1 - (1 - t)(1 - r)(1 - p_k)) \bar{J}_{k+1} \left( \frac{1 - (1 - t)(1 - p_k)}{1 - (1 - t)(1 - r)(1 - p_k)} \right) \\ &\quad + (1 - t)(1 - r)(1 - p_k) \bar{J}_{k+1}(0). \end{aligned}$$

As shown in Fig. 5.4.3, if  $I + (1 - t)C \leq C$  or, equivalently, if

$$I < tC,$$

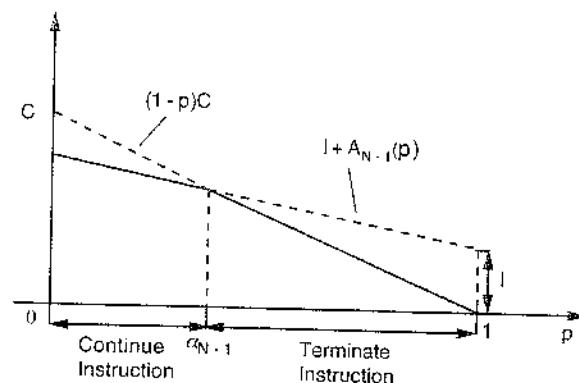


Figure 5.4.3 Determining the optimal instruction policy in the last period.

then there exists a scalar  $\alpha_{N-1}$  with  $0 < \alpha_{N-1} < 1$  that determines an optimal policy for the last period:

$$\begin{aligned} \text{continue instruction} &\quad \text{if } p_{N-1} \leq \alpha_{N-1}, \\ \text{terminate instruction} &\quad \text{if } p_{N-1} > \alpha_{N-1}. \end{aligned}$$

In the opposite case, where  $I \geq tC$ , the cost of instruction is so high relative to the cost of not learning that instructing the student is never optimal.

It may be shown by induction (Exercise 5.8) that if  $I < tC$ , the functions  $A_k(p)$  are concave and piecewise linear for each  $k$  and satisfy, for all  $k$ ,

$$A_k(1) = 0.$$

Furthermore, they satisfy for all  $k$ ,

$$A_k(p) \geq A_k(p'), \quad \text{for } 0 \leq p < p' \leq 1,$$

$$A_{k-1}(p) \leq A_k(p) \leq A_{k+1}(p), \quad \text{for all } p \in [0, 1].$$

Thus the functions  $(1 - p_k)C$  and  $I + A_k(p_k)$  intersect at a single point, and from the DP algorithm (5.44), we obtain that the optimal policy for each period is determined by the unique scalars  $\alpha_k$ , which are such that

$$(1 - \alpha_k)C = I + A_k(\alpha_k), \quad k = 0, 1, \dots, N-1.$$

An optimal policy for period  $k$  is given by

$$\begin{aligned} \text{continue instruction} &\quad \text{if } p_k \leq \alpha_k, \\ \text{terminate instruction} &\quad \text{if } p_k > \alpha_k. \end{aligned}$$

Since the functions  $A_k(p)$  are monotonically nondecreasing with respect to  $k$ , it follows from Fig. 5.4.4 that

$$\alpha_{N-1} \leq \alpha_{N-2} \leq \dots \leq \alpha_k \leq \dots \leq 1 - \frac{1}{C},$$

and therefore the sequence  $\{\alpha_k\}$  converges to some scalar  $\bar{\alpha}$  as  $k \rightarrow \infty$ . Thus, as the horizon gets longer, the optimal policy (at least for the initial stages) can be approximated by the stationary policy

$$\begin{aligned} \text{continue instruction} &\quad \text{if } p_k \leq \bar{\alpha}, \\ \text{terminate instruction} &\quad \text{if } p_k > \bar{\alpha}. \end{aligned} \tag{5.45}$$

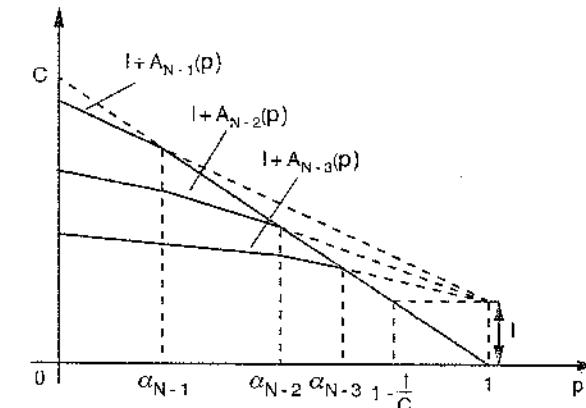


Figure 5.4.4 Demonstrating that the instruction thresholds are decreasing with time.

It turns out that this stationary policy has a convenient implementation that does not require the calculation of the conditional probability at each stage. From Eq. (5.43), we see that  $p_{k+1}$  increases over  $p_k$  if a correct answer  $R$  is given, and drops to zero if an incorrect answer  $\bar{R}$  is given. For  $m = 1, 2, \dots$ , define recursively the probability  $\pi_m$  of getting  $m$  successive correct answers following an incorrect answer:

$$\pi_1 = \Phi(0, R), \quad \pi_2 = \Phi(\pi_1, R), \quad \dots, \quad \pi_{k+1} = \Phi(\pi_k, R), \dots$$

Let  $n$  be the smallest integer for which  $\pi_n > \bar{\alpha}$ . Then the stationary policy (5.45) can be implemented by terminating instruction if and only if  $n$  successive correct answers have been received.

**Example 5.4.4 (Sequential Hypothesis Testing)**

Let us consider a hypothesis testing problem that is typical of statistical sequential analysis. A decision maker can make observations, at a cost  $C$  each, relating to two hypotheses. Given a new observation, he can either accept one of the hypotheses or delay the decision for one more period, pay the cost  $C$ , and obtain a new observation. At issue is trading off the cost of observation with the higher probability of accepting the wrong hypothesis.

Let  $z_0, z_1, \dots, z_{N-1}$  be the sequence of observations. We assume that they are independent, identically distributed random variables taking values on a finite set  $Z$ . Suppose we know that the probability distribution of the  $z_k$ 's is either  $f_0$  or  $f_1$  and that we are trying to decide on one of these. Here, for any element  $z \in Z$ ,  $f_0(z)$  and  $f_1(z)$  denote the probabilities of  $z$  when  $f_0$  and  $f_1$  are the true distributions, respectively. At time  $k$  after observing  $z_0, \dots, z_k$ , we may either stop observing and accept either  $f_0$  or  $f_1$ , or we may take an additional observation at a cost  $C > 0$ . If we stop observing and make a choice, then we incur zero cost if our choice is correct, and costs  $L_0$  and  $L_1$  if we choose incorrectly  $f_0$  and  $f_1$ , respectively. We are given the a priori probability  $p$  that the true distribution is  $f_0$ , and we assume that at most  $N$  observations are possible.

It can be seen that we are faced with an imperfect state information problem involving the two states:

$x^0$ : true distribution is  $f_0$ ,

$x^1$ : true distribution is  $f_1$ .

The alternate DP algorithm (5.40) and (5.41) is defined over the interval  $[0, 1]$  of possible values of the conditional probability

$$p_k = P(x_k = x^0 \mid z_0, \dots, z_k).$$

Similar to the previous section, we will obtain this algorithm directly.

The conditional probability  $p_k$  is generated recursively according to the following equation [assuming  $f_0(z) > 0, f_1(z) > 0$  for all  $z \in Z$ ]:

$$p_0 = \frac{p f_0(z_0)}{p f_0(z_0) + (1-p) f_1(z_0)}, \quad (5.46)$$

$$p_{k+1} = \frac{p_k f_0(z_{k+1})}{p_k f_0(z_{k+1}) + (1-p_k) f_1(z_{k+1})}, \quad k = 0, 1, \dots, N-2, \quad (5.47)$$

where  $p$  is the a priori probability that the true distribution is  $f_0$ . The optimal expected cost for the last period is

$$\bar{J}_{N-1}(p_{N-1}) = \min[(1-p_{N-1})L_0, p_{N-1}L_1], \quad (5.48)$$

where  $(1-p_{N-1})L_0$  is the expected cost for accepting  $f_0$  and  $p_{N-1}L_1$  is the expected cost for accepting  $f_1$ . Taking into account Eqs. (5.46) and (5.47),

we obtain the optimal expected cost-to-go for the  $k$ th period as

$$\begin{aligned} \bar{J}_k(p_k) = \min & \left[ (1-p_k)L_0, p_k L_1, \right. \\ & \left. C + E_{z_{k+1}} \left\{ \bar{J}_{k+1} \left( \frac{p_k f_0(z_{k+1})}{p_k f_0(z_{k+1}) + (1-p_k) f_1(z_{k+1})} \right) \right\} \right], \end{aligned}$$

where the expectation over  $z_{k+1}$  is taken with respect to the probability distribution

$$p(z_{k+1}) = p_k f_0(z_{k+1}) + (1-p_k) f_1(z_{k+1}), \quad z_{k+1} \in Z.$$

Equivalently, for  $k = 0, 1, \dots, N-2$ ,

$$\bar{J}_k(p_k) = \min[(1-p_k)L_0, p_k L_1, C + A_k(p_k)], \quad (5.49)$$

where

$$A_k(p_k) = E_{z_{k+1}} \left\{ \bar{J}_{k+1} \left( \frac{p_k f_0(z_{k+1})}{p_k f_0(z_{k+1}) + (1-p_k) f_1(z_{k+1})} \right) \right\}. \quad (5.50)$$

An optimal policy for the last period (see Fig. 5.4.5) is obtained from the minimization indicated in Eq. (5.48):

$$\text{accept } f_0 \quad \text{if } p_{N-1} \geq \gamma,$$

$$\text{accept } f_1 \quad \text{if } p_{N-1} < \gamma,$$

where  $\gamma$  is determined from the relation  $(1-\gamma)L_0 = \gamma L_1$  or equivalently

$$\gamma = \frac{L_0}{L_0 + L_1}.$$

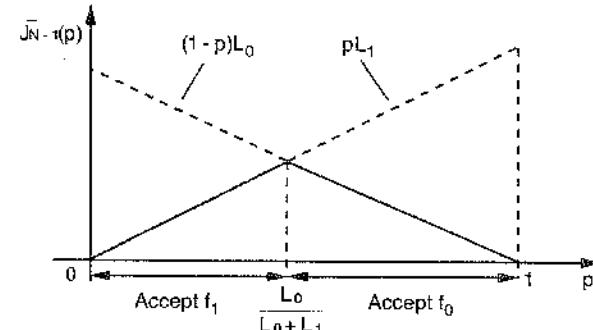


Figure 5.4.5 Determining the optimal policy for the last period.

We now prove that the functions  $A_k : [0, 1] \rightarrow R$  of Eq. (5.50) are concave, and satisfy for all  $k$  and  $p \in [0, 1]$

$$A_k(0) = A_k(1) = 0, \quad (5.51)$$

$$A_{k+1}(p) \leq A_k(p). \quad (5.52)$$

Indeed, we have for all  $p \in [0, 1]$

$$\bar{J}_{N-2}(p) \leq \min[(1-p)L_0, pL_0] = \bar{J}_{N-1}(p).$$

By making use of the stationarity of the system and the monotonicity property of DP (Exercise 1.23 in Chapter 1), we obtain

$$\bar{J}_k(p) \leq \bar{J}_{k+1}(p)$$

for all  $k$  and  $p \in [0, 1]$ . Using Eq. (5.50), we obtain  $A_{k+1}(p) \leq A_k(p)$  for all  $k$  and  $p \in [0, 1]$ .

To prove concavity of  $A_k$  in view of Eqs. (5.48) and (5.49), it is sufficient to show that concavity of  $\bar{J}_{k+1}$  implies concavity of  $A_k$  through relation (5.50). Indeed, assume that  $\bar{J}_{k+1}$  is concave over  $[0, 1]$ . Let  $z^1, z^2, \dots, z^n$  denote the elements of the observation space  $Z$ . We have from Eq. (5.50) that

$$A_k(p) = \sum_{i=1}^n (pf_0(z^i) + (1-p)f_1(z^i))\bar{J}_{k+1}\left(\frac{pf_0(z^i)}{pf_0(z^i) + (1-p)f_1(z^i)}\right).$$

Hence it is sufficient to show that concavity of  $\bar{J}_{k+1}$  implies concavity of each of the functions

$$H_i(p) = (pf_0(z^i) + (1-p)f_1(z^i))\bar{J}_{k+1}\left(\frac{pf_0(z^i)}{pf_0(z^i) + (1-p)f_1(z^i)}\right).$$

To show concavity of  $H_i$ , we must show that for every  $\lambda \in [0, 1]$ ,  $p_1, p_2 \in [0, 1]$  we have

$$\lambda H_i(p_1) + (1-\lambda)H_i(p_2) \leq H_i(\lambda p_1 + (1-\lambda)p_2).$$

Using the notation

$$\xi_1 = p_1 f_0(z^i) + (1-p_1)f_1(z^i), \quad \xi_2 = p_2 f_0(z^i) + (1-p_2)f_1(z^i),$$

this inequality is equivalent to

$$\begin{aligned} & \frac{\lambda \xi_1}{\lambda \xi_1 + (1-\lambda)\xi_2} \bar{J}_{k+1}\left(\frac{p_1 f_0(z^i)}{\xi_1}\right) + \frac{(1-\lambda)\xi_2}{\lambda \xi_1 + (1-\lambda)\xi_2} \bar{J}_{k+1}\left(\frac{p_2 f_0(z^i)}{\xi_2}\right) \\ & \leq \bar{J}_{k+1}\left(\frac{(\lambda p_1 + (1-\lambda)p_2) f_0(z^i)}{\lambda \xi_1 + (1-\lambda)\xi_2}\right). \end{aligned}$$

This relation, however, is implied by the concavity of  $\bar{J}_{k+1}$ .

Using Eqs. (5.51) and (5.52), we obtain (see Fig. 5.4.6) that if

$$C + A_{N-2}\left(\frac{L_0}{L_0 + L_1}\right) < \frac{L_0 L_1}{L_0 + L_1},$$

then an optimal policy for each period  $k$  is of the form

accept  $f_0$  if  $p_k \geq \alpha_k$ ,

accept  $f_1$  if  $p_k \leq \beta_k$ ,

continue the observations if  $\beta_k < p_k < \alpha_k$ ,

where the scalars  $\alpha_k, \beta_k$  are determined from the relations

$$\beta_k L_1 = C + A_k(\beta_k),$$

$$(1 - \alpha_k)L_0 = C + A_k(\alpha_k).$$

Furthermore, we have

$$\cdots \leq \alpha_{k+1} \leq \alpha_k \leq \alpha_{k-1} \leq \cdots \leq 1 - \frac{C}{L_0},$$

$$\cdots \geq \beta_{k+1} \geq \beta_k \geq \beta_{k-1} \geq \cdots \geq \frac{C}{L_1}.$$

Hence as  $N \rightarrow \infty$  the sequences  $\{\alpha_{N-i}\}, \{\beta_{N-i}\}$  converge to scalars  $\bar{\alpha}, \bar{\beta}$ , respectively, and the optimal policy is approximated by the stationary policy

$$\text{accept } f_0 \text{ if } p_k \geq \bar{\alpha}, \quad (5.53)$$

$$\text{accept } f_1 \text{ if } p_k \leq \bar{\beta}, \quad (5.54)$$

continue the observations if  $\bar{\beta} < p_k < \bar{\alpha}$ . (5.55)

Now the conditional probability  $p_k$  is given by

$$p_k = \frac{p f_0(z_0) \cdots f_0(z_k)}{p f_0(z_0) \cdots f_0(z_k) + (1-p) f_1(z_0) \cdots f_1(z_k)}, \quad (5.56)$$

where  $p$  is the a priori probability that  $f_0$  is the true hypothesis. Using Eq. (5.56), the stationary policy (5.53)-(5.55) can be written in the form

$$\text{accept } f_0 \text{ if } R_k \geq A, \quad (5.57)$$

$$\text{accept } f_1 \text{ if } R_k \leq B, \quad (5.58)$$

continue the observations if  $B < R_k < A$ , (5.59)

where

$$A = \frac{(1-p)\bar{\alpha}}{p(1-\bar{\alpha})},$$

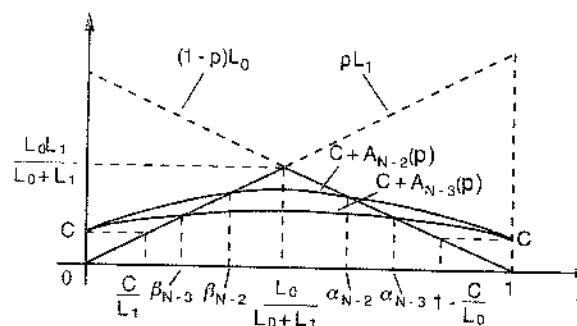


Figure 5.4.6 Determining the optimal hypothesis testing policy.

$$B = \frac{(1-p)\bar{\beta}}{p(1-\beta)},$$

and

$$R_k = \frac{f_0(z_0) \cdots f_0(z_k)}{f_1(z_0) \cdots f_1(z_k)}.$$

Note that  $R_k$  can be easily generated by means of the recursive equation

$$R_{k+1} = \frac{f_0(z_{k+1})}{f_1(z_{k+1})} R_k.$$

The policy (5.57)-(5.59) is known as the *sequential probability ratio test*, and was among the first formal methods studied in statistical sequential analysis by Wald [Wa47]. The optimality of this policy for the infinite horizon version of the problem will be shown in Vol. II, Chapter 3.

## 5.5 NOTES, SOURCES, AND EXERCISES

For literature on linear-quadratic problems with imperfect state information, see the references quoted for Section 4.1 and the survey paper by Wittenhause [Wit71]. The Kalman filtering algorithm is discussed in many textbooks, such as Anderson and Moore [AnM79], Ljung and Soderstrom [LjS83]. For linear-quadratic problems with Gaussian uncertainties and observation cost in the spirit of Exercise 5.6, see Aoki and Li [AoL69]. Exercise 5.1, which indicates the form of the certainty equivalence principle when the random disturbances are correlated, is based on an unpublished report by the author [Ber70]. The minimum variance approach is described in Aström and Wittenmark [AsW84], and Whittle [Whi63].

## Sec. 5.5 Notes, Sources, and Exercises

Problems with exponential cost functions are discussed in James, Baras, and Elliott [JBE94], and Fernandez-Gaucherand and Markus [FeM94].

The idea of data reduction via a sufficient statistic gained wide attention following the paper by Striebel [Str65]; see also Shiryaev [Shi64], [Shi66]. For the analog of the sufficient statistic idea in sequential minimax problems with set membership description of the uncertainty, see Bertsekas and Rhodes [BeR73].

Sufficient statistics have been used for analysis of finite-state problems with imperfect state information by Eckles [Eck68], and by Smallwood and Sondik [SmS73], [Son71]. The proof of piecewise linearity of the cost-to-go functions, and an algorithm for their computation are given by Smallwood and Sondik [SmS73], [Son71]. For further material on finite-state problems with imperfect state information, see Arapostathis et. al. [ABF93], Lovejoy [Lov91a], [Lov91b], and White and Scherer [WhS89].

The instruction model described in Example 5.4.3 has been considered (with some variations) by a number of authors such as Atkinson, Bower, and Crothers [ABC65], Groen and Atkinson [GrA66], Karush and Dear [KaD66], and Smallwood [Sma71].

For a discussion of the sequential probability ratio test (cf. Example 5.4.4) and related subjects, see Chernoff [Che72], DeGroot [DeG70], [Whi82], and the references quoted therein. The treatment given here stems from Arrow, Blackwell, and Girshick [ABG49].

## EXERCISES

### 5.1 (Linear Quadratic Problems – Correlated Disturbances) [www](#)

Consider the linear system and measurement equation of Section 5.2 and consider the problem of finding a policy  $\{\mu_0^*(I_0), \dots, \mu_{N-1}^*(I_{N-1})\}$  that minimizes the quadratic cost

$$E \left\{ x_N' Q x_N + \sum_{k=0}^{N-1} u_k' R_k u_k \right\}.$$

Assume, however, that the random vectors  $x_0, w_0, \dots, w_{N-1}, v_0, \dots, v_{N-1}$  are correlated and have a given joint probability distribution, and finite first and second moments. Show that the optimal policy is given by

$$\mu_k^*(I_k) = L_k E\{y_k | I_k\},$$

where the gain matrices  $L_k$  are obtained from the algorithm

$$L_k = -(B'_k K_{k+1} B_k + R_k)^{-1} B'_k K_{k+1} A_k,$$

$$K_N = Q.$$

$$K_k = A'_k (K_{k+1} - K_{k+1} B_k (B'_k K_{k+1} B_k + R_k)^{-1} B'_k K_{k+1}) A_k,$$

and the vectors  $y_k$  are given by

$$y_k = x_k + A_k^{-1} w_k + A_k^{-1} A_{k+1}^{-1} w_{k+1} + \cdots + A_k^{-1} \cdots A_{N-1}^{-1} w_{N-1}$$

(assuming the matrices  $A_0, A_1, \dots, A_{N-1}$  are invertible). Hint: Show that the cost can be written as

$$E \left\{ y_0' K_0 y_0 + \sum_{k=0}^{N-1} (u_k - L_k y_k)' P_k (u_k - L_k y_k) \right\},$$

where

$$P_k = B'_k K_{k+1} B_k + R_k.$$

## 5.2

Consider the scalar system

$$x_{k+1} = x_k + u_k + w_k,$$

$$z_k = x_k + v_k,$$

where we assume that the initial condition  $x_0$ , and the disturbances  $w_k$  and  $v_k$  are all independent. Let the cost be

$$E \left\{ x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + u_k^2) \right\},$$

and let the given probability distributions be

$$p(x_0 = 2) = \frac{1}{2}, \quad p(w_k = 1) = \frac{1}{2}, \quad p(v_k = \frac{1}{4}) = \frac{1}{2},$$

$$p(x_0 = -2) = \frac{1}{2}, \quad p(w_k = -1) = \frac{1}{2}, \quad p(u_k = -\frac{1}{4}) = \frac{1}{2}.$$

- (a) Determine the optimal policy. Hint: For this problem,  $E\{x_k | I_k\}$  can be determined from  $E\{x_{k-1} | I_{k-1}\}$ ,  $u_{k-1}$ , and  $z_k$ .
- (b) Determine the policy that is identical to the optimal except that it uses a linear least squares estimator of  $x_k$  given  $I_k$  in place of  $E\{x_k | I_k\}$  (see Appendix E – this policy can be shown to be optimal within the class of all policies that are linear functions of the measurements).
- (c) Determine the asymptotic form of the policies in parts (a) and (b) as  $N \rightarrow \infty$ .

## 5.3

A linear system with Gaussian disturbances and Gaussian initial state

$$x_{k+1} = Ax_k + Bu_k + w_k,$$

is to be controlled so as to minimize a quadratic cost similar to that in Section 5.2. The difference is that the controller has the option of choosing at each time  $k$  one of two types of measurements for the next stage ( $k+1$ ):

$$\text{first type: } z_{k+1} = C^1 x_{k+1} + v_{k+1}^1$$

$$\text{second type: } z_{k+1} = C^2 x_{k+1} + v_{k+1}^2.$$

Here  $C^1$  and  $C^2$  are given matrices of appropriate dimension, and  $\{v_k^1\}$  and  $\{v_k^2\}$  are zero-mean, independent, random sequences with given finite covariances that do not depend on  $x_0$  and  $\{w_k\}$ . There is a cost  $g_1$  (or  $g_2$ ) each time a measurement of type 1 (or type 2) is taken. The problem is to find the optimal control and measurement selection policy that minimizes the expected value of the sum of the quadratic cost

$$x_N' Q x_N + \sum_{k=0}^{N-1} (x_k' Q x_k + u_k' R u_k)$$

and the total measurement cost. Assume for convenience that  $N = 2$  and that the first measurement  $z_0$  is of type 1. Show that the optimal measurement selection at  $k = 0$  and  $k = 1$  does not depend on the value of the information vectors  $I_0$  and  $I_1$ , and can be determined a priori. Describe the nature of the optimal policy.

## 5.4

Consider a scalar single-input, single-output system given by

$$y_k + a_1 y_{k-1} + \cdots + a_m y_{k-m} = b_M u_{k-M} + \cdots + b_1 u_{k-1} + \\ + c_k + c_1 \epsilon_{k-1} + \cdots + c_m \epsilon_{k-m} + v_{k-n},$$

where  $1 \leq M \leq m$ ,  $0 \leq n \leq m$ , and  $v_k$  is generated by an equation of the form

$$v_k + d_1 v_{k-1} + \cdots + d_m v_{k-m} = \xi_k + \epsilon_1 \xi_{k-1} + \cdots + \epsilon_m \xi_{k-m},$$

and the polynomials  $(1 + c_1 s + \cdots + c_m s^m)$ ,  $(1 + d_1 s + \cdots + d_m s^m)$ , and  $(1 + \epsilon_1 s + \cdots + \epsilon_m s^m)$  have roots strictly outside the unit circle. The value of the scalar  $v_k$  is observed by the controller at time  $k$  together with  $y_k$ . The sequences  $\{\epsilon_k\}$  and  $\{\xi_k\}$  are zero mean independent identically distributed with finite variances. Find an easily implementable approximation to the minimum variance controller minimizing  $E\{\sum_{k=0}^N (y_k)^2\}$ . Discuss the stability properties of the closed-loop system.

## 5.5

- (a) Within the framework of the basic problem with imperfect state information, consider the case where the system and the observations are linear:

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + w_k, \\z_k &= C_k x_k + v_k.\end{aligned}$$

The initial state  $x_0$  and the disturbances  $w_k$  and  $v_k$  are assumed Gaussian and mutually independent. Their covariances are given, and  $w_k$  and  $v_k$  have zero mean. Show that  $E\{x_0 \mid I_0\}, \dots, E\{x_{N-1} \mid I_{N-1}\}$  constitute a sufficient statistic for this problem.

- (b) Use the result of part (a) to obtain an optimal policy for the special case of the single-stage problem involving the scalar system and observation

$$\begin{aligned}x_1 &= x_0 + u_0, \\z_0 &= x_0 + v_0,\end{aligned}$$

and the cost function  $E\{|x_k|\}$ .

- (c) Generalize part (b) for the case of the scalar system

$$\begin{aligned}x_{k+1} &= a x_k + u_k, \\z_k &= c x_k + v_k,\end{aligned}$$

and the cost function  $E\{\sum_{k=1}^N |x_k|\}$ . The scalars  $a$  and  $c$  are given. Note: You may find useful the following "differentiation of an integral" formula:

$$\begin{aligned}\frac{d}{dy} \int_{\alpha(y)}^{\beta(y)} f(y, \xi) d\xi &= \int_{\alpha(y)}^{\beta(y)} \frac{df(y, \xi)}{dy} d\xi \\&+ f(y, \beta(y)) \frac{d\beta(y)}{dy} - f(y, \alpha(y)) \frac{d\alpha(y)}{dy}.\end{aligned}$$

## 5.6

Consider a machine that can be in one of two states, good or bad. Suppose that the machine produces an item at the end of each period. The item produced is either good or bad depending on whether the machine is in a good or bad state at the beginning of the corresponding period, respectively. We suppose that once the machine is in a bad state it remains in that state until it is replaced. If the machine is in a good state at the beginning of a certain period, then with probability  $t$  it will be in the bad state at the end of the period. Once an item is produced, we may inspect the item at a cost  $I$  or not inspect. If an inspected item is found to be bad, the machine is replaced with a machine in good state at a cost  $R$ . The cost for producing a bad item is  $C > 0$ . Write a DP algorithm for obtaining an optimal inspection policy assuming a machine initially in good state and a horizon of  $N$  periods. Solve the problem for  $t = 0.2$ ,  $I = 1$ ,  $R = 3$ ,  $C = 2$ , and  $N = 8$ . (The optimal policy is to inspect at the end of the third period and not inspect in any other period.)

5.7 (Finite-State Systems – Imperfect State Information) [www](#)

Consider a system that at any time can be in any one of a finite number of states  $1, 2, \dots, n$ . When a control  $u$  is applied, the system moves from state  $i$  to state  $j$  with probability  $p_{ij}(u)$ . The control  $u$  is chosen from a finite collection  $u^1, u^2, \dots, u^m$ . Following each state transition, an observation is made by the controller. There is a finite number of possible observation outcomes  $z^1, z^2, \dots, z^q$ . The probability of occurrence of  $z^\theta$ , given that the current state is  $j$  and the preceding control was  $u$ , is denoted by  $r_j(u, \theta)$ ,  $\theta = 1, \dots, q$ .

- (a) Consider the column vector of conditional probabilities

$$P_k = [p_k^1, \dots, p_k^m]',$$

where

$$p_k^j = P(x_k = j \mid z_0, \dots, z_k, u_0, \dots, u_{k-1}), \quad j = 1, \dots, n,$$

and show that it can be updated according to

$$p_{k+1}^j = \frac{\sum_{i=1}^n p_k^i p_{ij}(u_k) r_j(u_k, z_{k+1})}{\sum_{s=1}^n \sum_{i=1}^n p_k^i p_{is}(u_k) r_s(u_k, z_{k+1})}, \quad j = 1, \dots, n.$$

Write this equation in the compact form

$$P_{k+1} = \frac{[r(u_k, z_{k+1})] * [P(u_k)' P_k]}{r(u_k, z_{k+1})' P(u_k)' P_k},$$

where prime denotes transposition and

$P(u_k)$  is the  $n \times n$  transition probability matrix with  $ij$ th element  $p_{ij}(u_k)$ ,

$r(u_k, z_{k+1})$  is the column vector with  $j$ th coordinate  $r_j(u_k, z_{k+1})$ ,

$[P(u_k)' P_k]$  is the  $j$ th coordinate of the vector  $P(u_k)' P_k$ ,

$[r(u_k, z_{k+1})] * [P(u_k)' P_k]$  is the vector whose  $j$ th coordinate is the scalar  $r_j(u_k, z_{k+1}) [P(u_k)' P_k]$ .

- (b) Assume that there is a cost for each stage  $k$  denoted  $g_k(i, u, j)$  and associated with the control  $u$  and a transition from  $i$  to  $j$ . There is no terminal cost. Consider the problem of finding a policy that minimizes the sum of expected costs per stage over  $N$  periods. Show that the corresponding DP algorithm is given by

$$\bar{J}_{N-1}(P_{N-1}) = \min_{u \in \{u^1, \dots, u^m\}} P_{N-1}' G_{N-1}(u),$$

$$\begin{aligned}\bar{J}_k(P_k) &= \min_{u \in \{u^1, \dots, u^m\}} \left[ P_k' G_k(u) \right. \\&\quad \left. + \sum_{\theta=1}^q r(u, \theta)' P(u)' P_k \bar{J}_{k+1} \left( \frac{[r(u, \theta)] * [P(u)' P_k]}{r(u, \theta)' P(u)' P_k} \right) \right],\end{aligned}$$

where  $G_k(u)$  is the vector of expected  $k$ th stage costs given by

$$G_k(u) = \begin{pmatrix} \sum_{j=1}^n p_{1j}(u)g_k(1, u, j) \\ \vdots \\ \sum_{j=1}^n p_{nj}(u)g_k(n, u, j) \end{pmatrix}.$$

- (c) Show that, for all  $k$ ,  $\bar{J}_k$  when viewed as a function defined on the set of vectors with nonnegative coordinates, is *positively homogeneous*; that is,

$$\bar{J}_k(\lambda P_k) = \lambda \bar{J}_k(P_k)$$

for all  $\lambda > 0$ . Use this fact to write the DP algorithm in the simpler form

$$\bar{J}_k(P_k) = \min_{u \in \{u^1, \dots, u^m\}} \left[ P'_k G_k(u) + \sum_{\theta=1}^q \bar{J}_{k-1}([r(u, \theta)] * [P(u)' P_k]) \right].$$

- (d) Show by induction that, for all  $k$ ,  $\bar{J}_k$  has the form

$$\bar{J}_k(P_k) = \min [P'_k \alpha_k^1, \dots, P'_k \alpha_k^{m_k}],$$

where  $\alpha_k^1, \dots, \alpha_k^{m_k}$  are some vectors in  $\mathbb{R}^n$ .

### 5.8

Consider the functions  $\bar{J}_k(p_k)$  in the instruction problem of Example 5.4.3. Show inductively that each of these functions is piecewise linear, concave, and of the form

$$\bar{J}_k(p_k) = \min [\alpha_k^1 + \beta_k^1 p_k, \alpha_k^2 + \beta_k^2 p_k, \dots, \alpha_k^{m_k} + \beta_k^{m_k} p_k],$$

where  $\alpha_k^1, \dots, \alpha_k^{m_k}, \beta_k^1, \dots, \beta_k^{m_k}$  are suitable scalars.

### 5.9

A person is offered  $N$  free plays to be distributed as he pleases between two slot machines A and B. Machine A pays  $\alpha$  dollars with known probability  $s$  and nothing with probability  $(1 - s)$ . Machine B pays  $\beta$  dollars with probability  $p$  and nothing with probability  $(1 - p)$ . The person does not know  $p$  but instead has an a priori probability distribution  $F(p)$  for  $p$ . The problem is to find a playing policy that maximizes expected profit. Let  $(m + n)$  denote the number of plays in machine B after  $k$  free plays ( $m + n \leq k$ ), and let  $m$  denote the number of successes and  $n$  the number of failures. Show that a DP algorithm for this problem is given for  $m + n \leq k$  by

$$\bar{J}_{N-1}(m, n) = \max [s\alpha, p(m, n)\beta],$$

$$\bar{J}_k(m, n) = \max \left[ s(s + \bar{J}_{k+1}(m, n)) + (1 - s)\bar{J}_{k+1}(m, n), p(m, n)(\beta + \bar{J}_{k-1}(m + 1, n)) + (1 - p(m, n))\bar{J}_{k-1}(m, n + 1) \right]$$

where

$$p(m, n) = \frac{\int_0^1 p^{m+1} (1-p)^n dF(p)}{\int_0^1 p^m (1-p)^n dF(p)}.$$

Solve the problem for  $N = 6$ ,  $\alpha = \beta = 1$ ,  $s = 0.6$ ,  $dF(p)/dp = 1$  for  $0 \leq p \leq 1$ . [The answer is to play machine B for the following pairs  $(m, n) : (0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (2, 1), (3, 1), (4, 1)$ . Otherwise, machine A should be played.]

### 5.10

A person is offered 2 to 1 odds in a coin-tossing game where he wins whenever a tail occurs. However, he suspects that the coin is biased and has an a priori probability distribution  $F(p)$  for the probability  $p$  that a head occurs at each toss. The problem is to find an optimal policy of deciding whether to continue or stop participating in the game given the outcomes of the game so far. A maximum of  $N$  tossings is allowed. Indicate how such a policy can be found by means of DP.

### 5.11

Consider the ARMAX model of Section 5.3 where instead of  $E \left\{ \sum_{k=1}^N (y_k)^2 \right\}$ , the cost is

$$E \left\{ \sum_{k=1}^N (y_k - \bar{y})^2 \right\},$$

where  $\bar{y}$  is a given scalar. Generalize the minimum variance policy for this case.

### 5.12

Consider the ARMAX model

$$y_k + ay_{k-1} = u_{k-M} + \epsilon_k,$$

where  $M \geq 1$ . Show that the minimum variance controller is

$$\mu_k(I_k) = au_{k-1} - a^2 u_{k-2} + \dots - (-1)^{M-1} a^{M-1} u_{k-M+1} - (-1)^M a^M y_k,$$

that the resulting closed-loop system is

$$y_k = \epsilon_k - a\epsilon_{k-1} + a^2 \epsilon_{k-2} - \dots + (-1)^{M-1} a^{M-1} \epsilon_{k-M+1},$$

and that the long-term output variance is

$$E \{ (y_k)^2 \} = \frac{1 - a^{2M}}{1 - a^2} E \{ (\epsilon_k)^2 \}.$$

Discuss the qualitative difference between the cases  $|a| < 1$  and  $|a| > 1$ , and relate it to the stability properties of the uncontrolled system  $y_k + ay_{k-1} = \epsilon_k$  and the size of the delay  $M$ .

### 5.13 (Linear-Quadratic Problems with Disturbance Estimation)

Consider the linear-quadratic problem discussed in Section 4.1 ( $A_k, B_k$ : known). The state  $x_k$  is perfectly observed at each stage, and the demands  $w_k$  are independent, identically distributed random vectors. However, the (common) distribution of the  $w_k$  is unknown. Instead it is known that this distribution is one out of two given distributions  $F_1$  and  $F_2$ , and that the a priori probability that  $F_1$  is the correct distribution is a given scalar  $q$ , with  $0 < q < 1$ . For convenience, assume that  $w_k$  can take a finite number of values under each of  $F_1$  and  $F_2$ .

- (a) Formulate this as an imperfect state information problem, and identify the state, control, system disturbance, observation, and observation disturbance.
- (b) Show that  $(x_k, q_k)$ , where

$$q_k = P(\text{distribution is } F_1 \mid w_0, \dots, w_{k-1}),$$

is a suitable sufficient statistic, and write a corresponding DP algorithm.

- (c) Show that the optimal control law is of the form

$$\mu_k(x_k, q_k) = -(B'_k K_{k+1} B_k + R_k)^{-1} B'_k K_{k+1} A_k x_k + c_k(q_k),$$

where the matrices  $K_k$  are given by the Riccati equation, and  $c_k(q_k)$  are appropriate functions of  $q_k$ . Hint: Show that the cost-to-go function has the form

$$J_k(x_k, q_k) = x_k' K_k x_k + a_k(q_k)' x_k + b_k(q_k),$$

where  $a_k(q_k)$  and  $b_k(q_k)$  are appropriate functions of  $q_k$ .

### 5.14 (Asset Selling Problem with Offer Estimation)

Consider the asset selling problem of Section 4.4. The offers  $w_k$  are independent and identically distributed. However, the (common) distribution of the  $w_k$  is unknown. Instead it is known that this distribution is one out of two given distributions  $F_1$  and  $F_2$ , and that the a priori probability that  $F_1$  is the correct distribution is a given scalar  $q$ , with  $0 < q < 1$ .

- (a) Formulate this as an imperfect state information problem, and identify the state, control, system disturbance, observation, and observation disturbance.
- (b) Show that  $(x_k, q_k)$ , where

$$q_k = P(\text{distribution is } F_1 \mid w_0, \dots, w_{k-1}),$$

is a suitable sufficient statistic, write a corresponding DP algorithm, and derive the form of the optimal selling policy.

### 5.15

Consider an inventory control problem where stock evolves according to

$$x_{k+1} = x_k + u_k - w_k,$$

and the cost of stage  $k$  is

$$cu_k + h \max(0, w_k - x_k - u_k) + p \max(0, x_k + u_k - w_k),$$

where  $c, h$ , and  $p$  are positive scalars with  $p > c$ . There is no terminal cost. The stock  $x_k$  is perfectly observed at each stage. The demands  $w_k$  are independent, identically distributed, nonnegative random variables. However, the (common) distribution of the  $w_k$  is unknown. Instead it is known that this distribution is one out of two given distributions  $F_1$  and  $F_2$ , and that the a priori probability that  $F_1$  is the correct distribution is a given scalar  $q$ , with  $0 < q < 1$ .

- (a) Formulate this as an imperfect state information problem, and identify the state, control, system disturbance, observation, and observation disturbance.
- (b) Write a DP algorithm in terms of a suitable sufficient statistic.
- (c) Characterize as best as you can the optimal policy.

### 5.16

Consider the search problem of Example 5.4.1 for different values of the search horizon  $N$ .

- (a) Show that for any value of the a priori probability  $p_0$  that is strictly less than 1, there is a threshold value of  $N$ , call it  $\bar{N}$ , such that the optimal reward function  $J_0(p_0)$  is independent of  $N$  as long as  $N > \bar{N}$ .
- (b) For  $N$  greater than the threshold  $\bar{N}$  of part (a) and for a given value of  $p_0$ , give a method to calculate the value of  $J_0(p_0)$  that does not use the DP algorithm.
- (c) Suppose that there are two sites that can be searched, instead of one. The sites may contain a treasure of corresponding values  $V^1$  and  $V^2$  (independently of each other), and the probabilities of a successful search are  $\beta_1$  and  $\beta_2$ , respectively. After finding a treasure in one site, one may continue searching for the treasure in the other site (but of course each search costs  $C$ ). Write a DP algorithm involving the probabilities  $p_k^1$  and  $p_k^2$  that a treasure is present at sites 1 and 2, respectively.
- (d) Under the assumptions of part (c), show that for any values of the a priori probabilities  $p_0^1, p_0^2$ , there is a threshold value of  $N$ , call it  $\bar{N}$ , such that the optimal cost-to-go function  $J_0(p_0^1, p_0^2)$  is independent of  $N$  as long as  $N > \bar{N}$ . Find the optimal search policy if  $N > \bar{N}$ .

# *Approximate Dynamic Programming*

## Contents

6.1.	Certainty Equivalent and Adaptive Control . . . . .	p. 283
6.1.1.	Caution, Probing, and Dual Control . . . . .	p. 289
6.1.2.	Two-Phase Control and Identifiability . . . . .	p. 291
6.1.3.	Certainty Equivalent Control and Identifiability . . . . .	p. 293
6.1.4.	Self-Tuning Regulators . . . . .	p. 298
6.2.	Open-Loop Feedback Control . . . . .	p. 300
6.3.	Limited Lookahead Policies . . . . .	p. 304
6.3.1.	Performance Bounds for Limited Lookahead Policies . . . . .	p. 305
6.3.2.	Computational Issues in Limited Lookahead . . . . .	p. 310
6.3.3.	Problem Approximation - Enforced Decomposition . . . . .	p. 312
6.3.4.	Aggregation . . . . .	p. 319
6.3.5.	Parametric Cost-to-Go Approximation . . . . .	p. 325
6.4.	Rollout Algorithms . . . . .	p. 335
6.4.1.	Discrete Deterministic Problems . . . . .	p. 342
6.4.2.	$Q$ -Factors Evaluated by Simulation . . . . .	p. 361
6.4.3.	$Q$ -Factor Approximation . . . . .	p. 363
6.5.	Model Predictive Control and Related Methods . . . . .	p. 366
6.5.1.	Rolling Horizon Approximations . . . . .	p. 367
6.5.2.	Stability Issues in Model Predictive Control . . . . .	p. 369
6.5.3.	Restricted Structure Policies . . . . .	p. 376
6.6.	Additional Topics in Approximate DP . . . . .	p. 382
6.6.1.	Discretization . . . . .	p. 382
6.6.2.	Other Approximation Approaches . . . . .	p. 384
6.7.	Notes, Sources, and Exercises . . . . .	p. 386

We have seen that it is sometimes possible to use the DP algorithm to obtain an optimal policy in closed form. However, this tends to be the exception. In most cases a numerical solution is necessary. The associated computational requirements are often overwhelming, and for many problems a complete solution of the problem by DP is impossible. To a great extent, the reason lies in what Bellman has called the "curse of dimensionality." This refers to an exponential increase of the required computation as the problem's size increases.

Consider for example a problem where the state, control, and disturbance spaces are the Euclidean spaces  $\mathbb{R}^n$ ,  $\mathbb{R}^m$ , and  $\mathbb{R}^r$ , respectively. In a straightforward numerical approach, these spaces are discretized. Taking  $d$  discretization points per state axis results in a state space grid with  $d^n$  points. For each of these points the minimization in the right-hand side of the DP equation must be carried out numerically, which involves comparison of as many as  $d^m$  numbers. To calculate each of these numbers, one must calculate an expected value over the disturbance, which is the weighted sum of as many as  $d^r$  numbers. Finally, the calculations must be done for each of the  $N$  stages. Thus as a first approximation, the number of computational operations is at least of the order of  $Nd^n$  and can be of the order of  $Nd^{n+m+r}$ . It follows that for perfect state information problems with Euclidean state and control spaces, DP can be applied numerically only if the dimensions of the spaces are relatively small. Based on the analysis of the preceding chapter, we can also conclude that for problems of imperfect state information the situation is hopeless, except for very simple or very special cases.

In the real world, there is an additional aspect of optimal control problems that can have a profound impact on the feasibility of DP as a practical solution method. In particular, there are many circumstances where the structure of the given problem is known well in advance, but some of the problem data, such as various system parameters, may be unknown until shortly before control is needed, thus seriously constraining the amount of time available for the DP computation. Usually this occurs as a result of one or both of the following situations:

- (a) *A family of problems is addressed, rather than a single problem,* and we do not get to know the exact problem to be solved until shortly before the control process begins. As an example, consider a problem of planning the daily route of a utility vehicle within a street network so that it passes through a number of points where it must perform some service. The street network and the vehicle characteristics may be known well in advance, but the service points may vary from day to day, and may not become known until shortly before the vehicle begins its route. This example is typical of situations, where the same problem must periodically be solved with small variations in its data. Yet, if DP is to be used, the solution of one instance of the problem

may not help appreciably in solving a different instance.

- (b) *The problem data changes as the system is being controlled.* As an example, consider the route planning example in case (a) above, and assume that new service points to be visited arise as the vehicle is on its way. It is possible in principle to model these data changes in terms of stochastic disturbances, but then we may end up with a problem that is too complicated for analysis or solution by DP. A frequently employed alternative is to use *on-line replanning*, whereby the problem is resolved on-line with the new data, as soon as these data become available, and control continues with a policy that corresponds to the new data.

A common feature of the above situations, which can seriously impact the solution, is that there may be stringent time constraints for the computation of the controls. This may substantially exacerbate the "curse of dimensionality" problem mentioned above.

As indicated by the above discussion, in practice one often has to settle for a suboptimal control scheme that strikes a reasonable balance between convenient implementation and adequate performance. In this chapter we discuss some general approaches for suboptimal control, which are based on approximations to the DP algorithm. We begin with two general schemes to simplify the DP computation, certainty equivalent control (Section 6.1), which replaces the stochastic quantities of the problem by deterministic nominal values, and open-loop-feedback control (Section 6.2), which ignores in part the availability of information in the future. These two schemes set the stage for limited lookahead control, which together with its many variations (Sections 6.3-6.5), is one of the principal approaches for suboptimal control. We also discuss adaptive control in the context of certainty equivalent control. This discussion is not used in subsequent developments, so the reader may skip Sections 6.1.1-6.1.4 if desired.

## 6.1 CERTAINTY EQUIVALENT AND ADAPTIVE CONTROL

The *certainty equivalent controller* (CEC) is a suboptimal control scheme that is inspired by linear-quadratic control theory. It applies at each stage the control that would be optimal if the uncertain quantities were fixed at some "typical" values; that is, it acts as if a form of the certainty equivalence principle were holding.

The advantage of the CEC is that it replaces the DP algorithm with what is often a much less demanding computation: the solution of a *deterministic* optimal control problem at each stage. This problem yields an optimal control sequence, the first component of which is used at the current stage, while the remaining components are discarded. The main attractive characteristic of the CEC is its ability to deal with stochastic

and even imperfect information problems by using the mature and effective methodology of deterministic optimal control.

We describe the CEC for the general problem with imperfect state information of Section 5.1. As can be expected, the implementation is considerably simpler if the controller has perfect state information. Suppose that we have an “estimator” that uses the information vector  $I_k$  to produce a “typical” value  $\bar{x}_k(I_k)$  of the state. Assume also that for every state-control pair  $(x_k, u_k)$  we have selected a “typical” value of the disturbance, which we denote by  $\bar{w}_k(x_k, u_k)$ . For example, if the state spaces and disturbance spaces are convex subsets of Euclidean spaces, the expected values

$$\bar{x}_k(I_k) = E\{x_k | I_k\}, \quad \bar{w}_k(x_k, u_k) = E\{w_k | x_k, u_k\},$$

can serve as typical values.

The control input  $\bar{\mu}_k(I_k)$  applied by the CEC at each time  $k$  is determined by the following rule:

- (1) Given the information vector  $I_k$ , compute the state estimate  $\bar{x}_k(I_k)$ .
- (2) Find a control sequence  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{N-1}\}$  that solves the deterministic problem obtained by fixing the uncertain quantities  $x_k$  and  $w_k, \dots, w_{N-1}$  at their typical values:

$$\text{minimize } g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, u_i, \bar{w}_i(x_i, u_i))$$

subject to the initial condition  $x_k = \bar{x}_k(I_k)$  and the constraints

$$u_i \in U_i, \quad x_{i+1} = f_i(x_i, u_i, \bar{w}_i(x_i, u_i)), \quad i = k, k+1, \dots, N-1.$$

- (3) Use as control the first element in the control sequence found:

$$\bar{\mu}_k(I_k) = \bar{u}_k.$$

Note that step (1) is unnecessary if we have perfect state information; in this case we simply use the known value of the  $x_k$ . The deterministic optimization problem in step (2) must be solved at each time  $k$ , once the initial state  $\bar{x}_k(I_k)$  becomes known by means of an estimation (or perfect observation) procedure. A total of  $N$  such problems must be solved by the CEC at every system run. In many cases of interest, these deterministic problems can be solved by powerful numerical methods such as conjugate gradient, Newton’s method, augmented Lagrangian, and sequential quadratic programming methods; see e.g. Luenberger [Lue84] or Bertsekas [Ber99]. Furthermore, the implementation of the CEC requires no storage of the type required for the optimal feedback controller.

An alternative to solving  $N$  optimal control problems in an “on-line” fashion is to solve these problems *a priori*. This is accomplished by computing an optimal feedback controller for the deterministic optimal control problem obtained from the original problem by replacing all uncertain quantities by their typical values. It is easy to verify, based on the equivalence of open-loop and feedback implementation of optimal controllers for deterministic problems, that the implementation of the CEC given earlier is equivalent to the following.

Let  $\{\mu_0^d(x_0), \dots, \mu_{N-1}^d(x_{N-1})\}$  be an optimal controller obtained from the DP algorithm for the deterministic problem

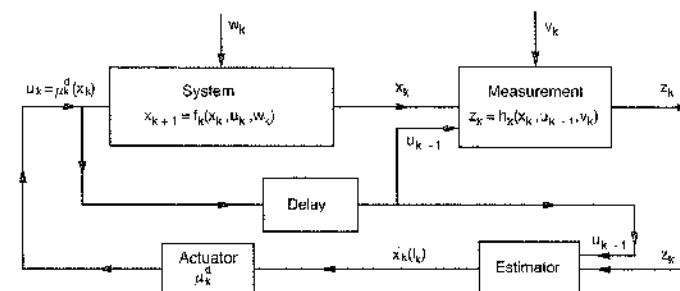
$$\text{minimize } g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), \bar{w}_k(x_k, u_k))$$

$$\text{subject to } x_{k+1} = f_k(x_k, \mu_k(x_k), \bar{w}_k(x_k, u_k)), \quad \mu_k(x_k) \in U_k, \quad k \geq 0.$$

Then the control input  $\bar{\mu}_k(I_k)$  applied by the CEC at time  $k$  is given by

$$\bar{\mu}_k(I_k) = \mu_k^d(\bar{x}_k(I_k))$$

as shown in Fig. 6.1.1.



**Figure 6.1.1** Structure of the certainty equivalent controller when implemented in feedback form.

In other words, an equivalent alternative implementation of the CEC consists of finding a feedback controller  $\{\mu_0^d, \mu_1^d, \dots, \mu_{N-1}^d\}$  that is optimal for a corresponding deterministic problem, and subsequently using this controller for control of the uncertain system [modulo substitution of the state  $x_k$  by its estimate  $\bar{x}_k(I_k)$ ]. Either one of the definitions given for the CEC can serve as a basis for its implementation. Depending on the nature of the problem, one method may be preferable to the other.

The CEC approach often performs well in practice and yields near-optimal policies. In fact, for the linear-quadratic problems of Sections 4.1

and 5.2, the CEC is identical to the optimal controller (certainty equivalence principle). It is possible, however, that a CEC performs strictly worse than the optimal open-loop controller (see Exercise 6.2).

In what follows in this section, we will discuss a few variants of the CEC, and we will then focus on one particular type of methodology, adaptive control of systems with unknown parameters.

### Certainty Equivalent Control with Heuristics

Even though the CEC approach simplifies a great deal the computations, it still requires the solution of a deterministic optimal control problem at each stage. This problem may be difficult, and a more convenient approach may be to solve it suboptimally using a heuristic algorithm. To simplify notation, let us assume perfect state information [the ideas to be discussed can also be applied to imperfect state information problems, by substituting  $x_k$  with its estimate  $\bar{x}_k(I_k)$ ]. Then, in this approach, given  $x_k$ , we use some (easily implementable) heuristic to find a suboptimal control sequence  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{N-1}\}$  for the problem

$$\text{minimize } g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, u_i, \bar{w}_i(x_i, u_i))$$

subject to

$$u_i \in U_i(x_i), \quad x_{i+1} = f_i(x_i, u_i, \bar{w}_i(x_i, u_i)), \quad i = k, k+1, \dots, N-1.$$

We then use  $\bar{u}_k$  as the control for stage  $k$ .

An important enhancement of this idea is to use minimization over the first control  $u_k$  and to use the heuristic only for the remaining stages  $k+1, \dots, N-1$ . To implement this variant of the CEC, we must apply at time  $k$  a control  $\bar{u}_k$  that minimizes over  $u_k \in U_k(x_k)$  the expression

$$g_k(x_k, u_k, \bar{w}_k(x_k, u_k)) + H_{k+1}(f_k(x_k, u_k, \bar{w}_k(x_k, u_k))), \quad (6.1)$$

where  $H_{k+1}$  is the cost-to-go function corresponding to the heuristic, i.e.,  $H_{k+1}(x_{k+1})$  is the cost incurred over the remaining stages  $k+1, \dots, N-1$  starting from a state  $x_{k+1}$ , using the heuristic, and assuming that the future disturbances will be equal to their typical values  $\bar{w}_i(x_i, u_i)$ . Note that for any next-stage state  $x_{k+1}$ , it is not necessary to have a closed-form expression for the heuristic cost-to-go  $H_{k+1}(x_{k+1})$ . Instead we can generate this cost by running the system forward from  $x_{k+1}$  and accumulating the corresponding single-stage costs. Since the heuristic must be run for each possible value of the control  $u_k$  to calculate the costs  $H_{k+1}(f_k(x_k, u_k, \bar{w}_k(x_k, u_k)))$  needed in the minimization, it is necessary to discretize the control constraint set if it is not already finite.

Note that the general structure of the preceding variant of the CEC is similar to the one of standard DP. It involves minimization of the expression (6.1), which is the sum of a current stage cost and a cost-to-go starting from the next state. The difference with DP is that the optimal cost-to-go  $J_{k+1}^*(x_{k+1})$  is replaced by the heuristic cost  $H_{k+1}(x_{k+1})$ , and the disturbance  $w_k$  is replaced by its typical value  $\bar{w}_k(x_k, u_k)$  (so that there is no need to take expectation over  $w_k$ ). We thus encounter for the first time an important suboptimal control idea, based on an approximation to the DP algorithm: *minimizing at each stage  $k$  the sum of approximations to the current stage cost and the optimal cost-to-go*. This idea is central in other types of suboptimal control such as the limited lookahead, rollout, and model predictive control approaches, which will be discussed in Sections 6.3-6.5.

### Partially Stochastic Certainty Equivalent Control

In the preceding descriptions of the CEC all future disturbances are fixed at their typical values. A useful variation for some imperfect state information problems is to take into account the stochastic nature of these disturbances, and to treat the problem as one of perfect state information, using an estimate  $\bar{x}_k(I_k)$  of  $x_k$  as if it were exact. Thus, if  $\{\mu_k^p(x_0), \dots, \mu_{N-1}^p(x_{N-1})\}$  is an optimal policy obtained from the DP algorithm for the stochastic *perfect state information* problem

$$\text{minimize } E \left\{ g_N(x_N) - \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

subject to  $x_{k+1} = f_k(x_k, \mu_k(x_k), w_k)$ ,  $\mu_k(x_k) \in U_k$ ,  $k = 0, \dots, N-1$ , then the control input  $\bar{\mu}_k(I_k)$  applied by this variant of CEC at time  $k$  is given by

$$\bar{\mu}_k(I_k) = \mu_k^p(\bar{x}_k(I_k)).$$

Generally, there are several variants of the CEC, where the stochastic uncertainty about some of the unknown quantities is explicitly dealt with, while all other unknown quantities are replaced by estimates obtained in a variety of ways. Let us provide some examples.

#### Example 6.1.1 (Multiaccess Communication)

Consider the slotted Aloha system described in Example 5.1.1. It is very difficult to obtain an optimal policy for this problem, primarily because there is no simple characterization of the conditional distribution of the state (the system backlog), given the channel transmission history. We therefore resort to a suboptimal policy. As discussed in Section 5.1, the perfect state information version of the problem admits a simple optimal policy:

$$\mu_k(x_k) = \frac{1}{x_k}, \quad \text{for all } x_k \geq 1.$$

As a result, there is a natural partially stochastic CEC,

$$\bar{\mu}_k(I_k) = \min \left[ 1, \frac{1}{\bar{x}_k(I_k)} \right],$$

where  $\bar{x}_k(I_k)$  is an estimate of the current packet backlog based on the entire past channel history of successes, idles, and collisions (which is  $I_k$ ). Recursive estimators for generating  $\bar{x}_k(I_k)$  are discussed by Mikhailov [Mik79], Hajek and van Loon [HaL82], Tsitsiklis [Tsi87], and Bertsekas and Gallager [BeG92].

### Example 6.1.2 (Finite-State Systems with Imperfect State Information)

Consider the case where the system is a finite-state Markov chain under imperfect state information. The partially stochastic CEC approach is to solve the corresponding problem of perfect state information, and then use the controller thus obtained for control of the imperfectly observed system, modulo substitution of the exact state by an estimate obtained via the Viterbi algorithm described in Section 2.2.2. In particular, suppose that  $\{\mu_0^p, \dots, \mu_{N-1}^p\}$  is an optimal policy for the corresponding problem where the state is perfectly observed. Then the partially stochastic CEC, given the information vector  $I_k$ , uses the Viterbi algorithm to obtain (in real time) an estimate  $\bar{x}(I_k)$  of the current state  $x_k$ , and applies the control

$$\bar{\mu}_k(I_k) = \mu_k^p(\bar{x}_k(I_k)).$$

### Example 6.1.3 (Systems with Unknown Parameters)

We have been dealing so far with systems having a known system equation. In practice, however, there are many cases where the system parameters are not known exactly or change over time. One possible approach is to estimate the unknown parameters from input-output records of the system by using system identification techniques. This is a broad and important methodology, for which we refer to textbooks such as Kumar and Varaiya [KuV86], Ljung and Soderstrom [LjS83], and Ljung [Lju86]. However, system identification can be time consuming, and thus difficult to apply in an on-line control context. Furthermore, the estimation must be repeated if the parameters change.

The alternative is to formulate the stochastic control problem so that unknown parameters are dealt with directly. It can be shown that problems involving unknown system parameters can be embedded within the framework of our basic problem with imperfect state information by using state augmentation. Indeed, let the system equation be of the form

$$x_{k+1} = f_k(x_k, \theta, u_k, w_k),$$

where  $\theta$  is a vector of unknown parameters with a given a priori probability distribution. We introduce an additional state variable  $y_k = \theta$  and obtain a system equation of the form

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, y_k, u_k, w_k) \\ y_k \end{pmatrix}.$$

This equation can be written compactly as

$$\tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, u_k, w_k),$$

where  $\tilde{x}_k = (x_k, y_k)$  is the new state, and  $\tilde{f}_k$  is an appropriate function. The initial state is

$$\tilde{x}_0 = (x_0, \theta).$$

With a suitable reformulation of the cost function, the resulting problem becomes one that fits our usual framework.

Unfortunately, however, since  $y_k$  (i.e.,  $\theta$ ) is unobservable, we are faced with a problem of imperfect state information even if the controller knows the state  $x_k$  exactly. Thus, typically an optimal solution cannot be found. Nonetheless, the partially stochastic CEC approach is often convenient. In particular, suppose that for a fixed parameter vector  $\theta$ , we can compute the corresponding optimal policy

$$\{\mu_0^*(J_0, \theta), \dots, \mu_{N-1}^*(I_{N-1}, \theta)\};$$

this is true for example if for a fixed  $\theta$ , the problem is linear-quadratic of the type considered in Sections 4.1 and 5.2. Then a partially stochastic CEC takes the form

$$\bar{\mu}_k(I_k) = \mu_k^*(I_k, \hat{\theta}_k),$$

where  $\hat{\theta}_k$  is some estimate of  $\theta$  based on the information vector  $I_k$ . Thus, in this approach, the system is identified while it is being controlled. However, the estimates of the unknown parameters are used as if they were exact.

The approach of the preceding example is one of the principal methods of *adaptive control*, that is, control that adapts itself to changing values of system parameters. In the remainder of this section, we discuss some of the associated issues. Because adaptive control is somewhat disjoint from other material in the chapter, the reader may skip directly to Section 6.2.

#### 6.1.1 Caution, Probing, and Dual Control

Suboptimal control is often guided by the qualitative nature of optimal control. It is therefore important to try to understand some of the characteristic features of the latter in the case where some of the system parameters are unknown. One of these is the need for balance between “caution” (the need for conservatism in applying control, since the system is not fully known), and “probing” (the need for aggressiveness in applying control, in order to excite the system enough to be able to identify it). These notions cannot be easily quantified, but often manifest themselves in specific control schemes. The following example provides some orientation; see also Bar-Shalom [Bar81].

**Example 6.1.4 [Kurn83]**

Consider the linear scalar system

$$x_{k+1} = x_k + bu_k + w_k, \quad k = 0, 1, \dots, N-1,$$

and the quadratic terminal cost  $E\{(x_N)^2\}$ . Here everything is as in Section 4.1 (perfect state information) except that the control coefficient  $b$  is unknown. Instead, it is known that the a priori probability distribution of  $b$  is Gaussian with mean and variance

$$\bar{b} = E\{b\} > 0, \quad \sigma_b^2 = E\{(b - \bar{b})^2\}.$$

Furthermore,  $w_k$  is zero mean Gaussian with variance  $\sigma_w^2$  for each  $k$ .

Consider first the case where  $N = 1$ , so the cost is calculated to be

$$E\{(x_1)^2\} = E\{(x_0 + bu_0 + w_0)^2\} = x_0^2 + 2\bar{b}x_0 u_0 + (\bar{b}^2 + \sigma_b^2)u_0^2 + \sigma_w^2.$$

The minimum over  $u_0$  is attained at

$$u_0 = -\frac{\bar{b}}{\bar{b}^2 + \sigma_b^2}x_0,$$

and the optimal cost is verified by straightforward calculation to be

$$\frac{\sigma_b^2}{\bar{b}^2 + \sigma_b^2}x_0^2 + \sigma_w^2.$$

Therefore, the optimal control here is *cautious* in that the optimum  $|u_0|$  decreases as the uncertainty in  $b$  (i.e.,  $\sigma_b^2$ ) increases.

Consider next the case where  $N = 2$ . The optimal cost-to-go at stage 1 is obtained by the preceding calculation:

$$J_1(I_1) = \frac{\sigma_b^2(1)}{(\bar{b}(1))^2 + \sigma_b^2(1)}x_1^2 + \sigma_w^2, \quad (6.2)$$

where  $I_1 = (x_0, u_0, x_1)$  is the information vector and

$$\bar{b}(1) = E\{b | I_1\}, \quad \sigma_b^2(1) = E\{(b - \bar{b}(1))^2 | I_1\}.$$

Let us focus on the term  $\sigma_b^2(1)$  in the expression (6.2) for  $J_1(I_1)$ . We can obtain  $\sigma_b^2(1)$  from the equation  $x_1 = x_0 + bu_0 + w_0$  (which we view as a noise-corrupted measurement of  $b$ ) and least-squares estimation theory (see Appendix E). The formula for  $\sigma_b^2(1)$  will be of no further use to us, so we just state it without going into the calculation:

$$\sigma_b^2(1) = \frac{\sigma_b^2 \sigma_w^2}{u_0^2 \sigma_b^2 + \sigma_w^2}.$$

The salient feature of this equation is that  $\sigma_b^2(1)$  is affected by the control  $u_0$ . Basically, if  $|u_0|$  is small, the measurement  $x_1 = x_0 + bu_0 + w_0$  is dominated by  $w_0$  and the "signal-to-noise ratio" is small. Thus to achieve small error variance  $\sigma_b^2(1)$  [which is desirable in view of Eq. (6.2)], we must apply a control  $u_0$  that is large in absolute value. A choice of large control to enhance parameter identification is often referred to as *probing*. On the other hand, if  $|u_0|$  is large,  $|x_1|$  will also be large, and this is not desirable in view of Eq. (6.2). Therefore, in choosing  $u_0$  we must strike a balance between caution (choosing a small value to keep  $x_1$  reasonably small) and probing (choosing a large value to improve the signal-to-noise ratio and enhance estimation of  $b$ ).

The tradeoff between the control objective and the parameter estimation objective is commonly referred to as *dual control*.

**6.1.2 Two-Phase Control and Identifiability**

An apparently reasonable form of suboptimal control in the presence of unknown parameters (cf. Example 6.1.3) is to separate the control process into two phases, a *parameter identification phase* and a *control phase*. In the first phase the unknown parameters are identified, while the control takes no account of the interim results of identification. The final parameter estimates from the first phase are then used to implement an optimal control law in the second phase. This alternation of identification and control phases may be repeated several times during any system run in order to take into account subsequent changes of the parameters.

One drawback of this approach is that information gathered during the identification phase is not used to adjust the control law until the beginning of the second phase. Furthermore, it is not always easy to determine when to terminate one phase and start the other.

A second difficulty, of a more fundamental nature, is due to the fact that the control process may make some of the unknown parameters invisible to the identification process. This is the problem of parameter *identifiability*, discussed by Ljung [Lju86], which is best explained by means of an example.

**Example 6.1.5**

Consider the scalar system

$$x_{k+1} = ax_k + bu_k + w_k, \quad k = 0, 1, \dots, N-1,$$

with the quadratic cost

$$E \left\{ \sum_{k=1}^N (x_k)^2 \right\}.$$

We assume perfect state information, so if the parameters  $a$  and  $b$  are known, this is a minimum variance control problem (cf. Section 5.3), and the optimal

control law is

$$\mu_k^*(x_k) = -\frac{a}{b}x_k.$$

Assume now that the parameters  $a$  and  $b$  are unknown, and consider the two-phase method. During the first phase the control law

$$\tilde{\mu}_k(x_k) = \gamma x_k \quad (6.3)$$

is used ( $\gamma$  is some scalar; for example,  $\gamma = -\bar{a}/\bar{b}$ , where  $\bar{a}$  and  $\bar{b}$  are a priori estimates of  $a$  and  $b$ , respectively). At the end of the first phase, the control law is changed to

$$\tilde{\mu}_k(x_k) = -\frac{\hat{a}}{\hat{b}}x_k,$$

where  $\hat{a}$  and  $\hat{b}$  are the estimates obtained from the identification process. However, with the control law (6.3), the closed-loop system is

$$x_{k+1} = (a + b\gamma)x_k + w_k,$$

so the identification process can at best identify the value of  $(a + b\gamma)$  but not the values of both  $a$  and  $b$ . In other words, the identification process cannot discriminate between pairs of values  $(a_1, b_1)$  and  $(a_2, b_2)$  such that  $a_1 + b_1\gamma = a_2 + b_2\gamma$ . Therefore,  $a$  and  $b$  are not identifiable when feedback control of the form (6.3) is applied.

One way to correct the difficulty is to add an additional known input  $\delta_k$  to the control law (6.3); that is, use

$$\tilde{\mu}_k(x_k) = \gamma x_k + \delta_k.$$

Then the closed-loop system becomes

$$x_{k+1} = (a + b\gamma)x_k + b\delta_k + w_k,$$

and the knowledge of  $\{x_k\}$  and  $\{\delta_k\}$  makes it possible to identify  $(a + b\gamma)$  and  $b$ . Given  $\gamma$ , one can then obtain estimates of  $a$  and  $b$ . Actually, to guarantee this in a more general context where the system is of higher dimension, the sequence  $\{\delta_k\}$  must satisfy certain conditions: it must be “persistently exciting” (see for example Ljung and Soderstrom [LjS83] for further explanation of this concept).

A second possibility to bypass the identifiability problem is to change the structure of the system by artificially introducing a one-unit delay in the control feedback. Thus, instead of considering control laws of the form  $\tilde{\mu}_k(x_k) = \gamma x_k$ , as in Eq. (6.3), we consider controls of the form

$$u_k = \tilde{\mu}_k(x_{k-1}) = \gamma x_{k-1}.$$

The closed-loop system then becomes

$$x_{k+1} = ax_k + b\gamma x_{k-1} + w_k,$$

and given  $\gamma$ , it is possible to identify both parameters  $a$  and  $b$ . This technique can be generalized for systems of arbitrary order, but artificially introducing a control delay makes the system less responsive to control.

### 6.1.3 Certainty Equivalent Control and Identifiability

At the opposite extreme of the two-phase method we have the certainty equivalent control approach, where the parameter estimates are incorporated into the control law as they are generated, and they are treated as if they were true values. In terms of the system

$$x_{k+1} = f_k(x_k, \theta, u_k, w_k)$$

considered in Example 6.1.3, suppose that, for each possible value of  $\theta$ , the control law  $\pi^*(\theta) = \{\mu_0^*(\cdot, \theta), \dots, \mu_{N-1}^*(\cdot, \theta)\}$  is optimal with respect to a certain cost  $J_\pi(x_0, \theta)$ . Then the (suboptimal) control used at time  $k$  is

$$\hat{\mu}_k(I_k) = \mu_k^*(x_k, \hat{\theta}_k),$$

where  $\hat{\theta}_k$  is an estimate of  $\theta$  based on the information

$$I_k = \{x_0, x_1, \dots, x_k, u_0, u_1, \dots, u_{k-1}\}$$

available at time  $k$ ; for example,

$$\hat{\theta}_k = E\{\theta | I_k\}$$

or, more likely in practice, an estimate obtained via an on-line system identification method (see [KuV86], [LjS83], [Lju86]).

One would hope that when the horizon is very long, the parameter estimates  $\hat{\theta}_k$  will converge to the true value  $\theta$ , so the certainty equivalent controller will become asymptotically optimal. Unfortunately, we will see that difficulties related to identifiability arise here as well.

Suppose for simplicity that the system is stationary with a priori known transition probabilities  $P\{x_{k+1} | x_k, u_k, \theta\}$  and that the control law used is also stationary:

$$\hat{\mu}_k(I_k) = \mu^*(x_k, \hat{\theta}_k), \quad k = 0, 1, \dots$$

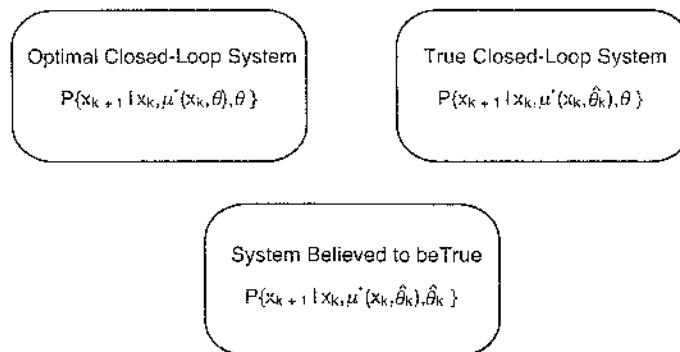
There are three systems of interest here (cf. Fig. 6.1.2):

- (a) The system (perhaps falsely) believed by the controller to be true, which evolves probabilistically according to

$$P\{x_{k+1} | x_k, \mu^*(x_k, \hat{\theta}_k), \hat{\theta}_k\}.$$

- (b) The true closed-loop system, which evolves probabilistically according to

$$P\{x_{k+1} | x_k, \mu^*(x_k, \hat{\theta}_k), \theta\}.$$



**Figure 6.1.2** The three systems involved in certainty equivalent control, where  $\theta$  is the true parameter and  $\hat{\theta}_k$  is the parameter estimate at time  $k$ . Loss of optimality occurs when the true system differs asymptotically from the optimal closed-loop system. If the parameter estimates converge to some value  $\hat{\theta}$ , the true system typically becomes asymptotically equal to the system believed to be true. However, the parameter estimates need not converge, and even if they do, both systems may be different asymptotically from the optimal.

- (c) The optimal closed-loop system that corresponds to the true value of the parameter, which evolves probabilistically according to

$$P\{x_{k+1} | x_k, \mu^*(x_k, \theta), \theta\}.$$

For asymptotic optimality, we would like the last two systems to be equal asymptotically. This will certainly be true if  $\hat{\theta}_k \rightarrow \theta$ . However, it is quite possible that either

- (1)  $\hat{\theta}_k$  does not converge to anything, or that
- (2)  $\hat{\theta}_k$  converges to a parameter  $\hat{\theta} \neq \theta$ .

There is not much we can say about the first case, so we concentrate on the second. To see how the parameter estimates can converge to a wrong value, assume that for some  $\hat{\theta} \neq \theta$  and all  $x_{k+1}, x_k$ , we have

$$P\{x_{k+1} | x_k, \mu^*(x_k, \hat{\theta}), \hat{\theta}\} = P\{x_{k+1} | x_k, \mu^*(x_k, \hat{\theta}), \theta\}. \quad (6.4)$$

In words, *there is a false value of parameter for which the system under closed-loop control looks exactly as if the false value were true*. Then, if the controller estimates at some time the parameter to be  $\hat{\theta}$ , subsequent data will tend to reinforce this erroneous estimate. As a result, a situation may develop where the identification procedure locks onto a wrong parameter value, regardless of how long information is collected. This is a difficulty with identifiability of the type discussed earlier in connection with two-phase control.

On the other hand, if the parameter estimates converge to some (possibly wrong) value, we can argue intuitively that the first two systems (believed and true) typically become equal in the limit as  $k \rightarrow \infty$ ; since, generally, parameter estimate convergence in identification methods implies that the data obtained are asymptotically consistent with the view of the system one has based on the current estimates. However, the believed and true systems may or may not become asymptotically equal to the optimal closed-loop system. We first present two examples that illustrate how, even when the parameter estimates converge, the true closed-loop system can differ asymptotically from the optimal, thereby resulting in a certainty equivalent controller that is strictly suboptimal. We then discuss the special case of the self-tuning regulator for ARMAX models with unknown parameters, where, remarkably, it turns out that all three of the above systems are typically equal in the limit, even though the parameter estimates typically converge to false values.

#### Example 6.1.6 [BoV79]

Consider a two-state system with two controls  $u^1$  and  $u^2$ . The transition probabilities depend on the control applied as well as a parameter  $\theta$ , which is known to take one of two values  $\theta^*$  and  $\hat{\theta}$ . They are as shown in Fig. 6.1.3. There is zero cost for a transition from state 1 to itself and a unit cost for all other transitions. Therefore, the optimal control at state 1 is the one that maximizes the probability of the state remaining at 1. Assume that the true parameter is  $\theta^*$  and that

$$p_{11}(u^1, \hat{\theta}) > p_{11}(u^2, \hat{\theta}), \quad p_{11}(u^1, \theta^*) < p_{11}(u^2, \theta^*).$$

Then the optimal control is  $u^2$ , but if the controller *thinks* that the true parameter is  $\hat{\theta}$ , it will apply  $u^1$ . Suppose also that

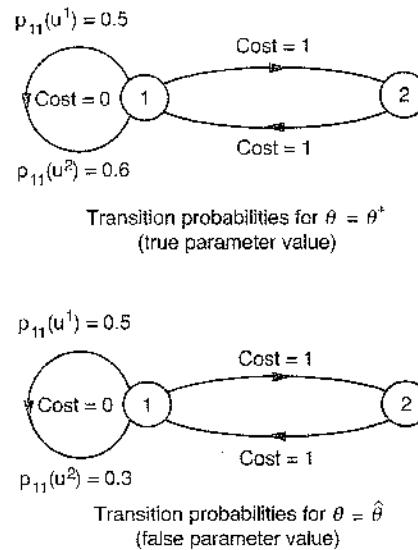
$$p_{11}(u^1, \hat{\theta}) = p_{11}(u^1, \theta^*).$$

Then, under  $u^1$  the system looks identical for both values of the parameter, so if the controller estimates the parameter to be  $\hat{\theta}$  and applies  $u^1$ , subsequent data will tend to reinforce the controller's belief that the true parameter is indeed  $\hat{\theta}$ .

More precisely, suppose that we estimate  $\theta$  by selecting at each time  $k$  the value that maximizes

$$P\{\theta | I_k\} = \frac{P\{I_k | \theta\} P(\theta)}{P(I_k)},$$

where  $P(\theta)$  is the a priori probability that the true parameter is  $\theta$  (this is a popular estimation method). Then if  $P(\hat{\theta}) > P(\theta^*)$ , it can be seen, by using induction, that at each time  $k$ , the controller will estimate falsely  $\theta$  to be  $\hat{\theta}$  and apply the incorrect control  $u^1$ . To avoid the difficulty illustrated in this example, it has been suggested to occasionally deviate from the certainty



**Figure 6.1.3** Transition probabilities for the two-state system of Example 6.1.6. Under the nonoptimal control  $u^1$ , the system looks identical under the true and the false values of the parameter  $\theta$ .

equivalent control, applying other controls that enhance the identification of the unknown parameter (see Doshi and Shreve [DoS80], and Kumar and Lin [KuL82]). For example, by making sure that the control  $u^2$  is used infrequently but infinitely often, we can guarantee that the correct parameter value will be identified by the preceding estimation scheme.

### Example 6.1.7 [Kum83]

Consider the linear scalar system

$$x_{k+1} = ax_k + bu_k + w_k,$$

where we know that the parameters are either  $(a, b) = (1, 1)$  or  $(a, b) = (0, -1)$ . The sequence  $\{w_k\}$  is independent, stationary, zero mean, and Gaussian. The cost is quadratic of the form

$$\sum_{k=0}^{N-1} ((x_k)^2 + 2(u_k)^2),$$

where  $N$  is very large, so the stationary form of the optimal control law is used (see Section 4.1). This control law can be calculated via the Riccati

equation to be

$$\mu^*(x_k) = \begin{cases} -\frac{x_k}{2} & \text{if } (a, b) = (1, 1), \\ 0 & \text{if } (a, b) = (0, -1). \end{cases}$$

To estimate  $(a, b)$ , we use a least-squares identification method. The value of the least-squares criterion at time  $k$  is given by

$$V_k(1, 1) = \sum_{i=0}^{k-1} (x_{i+1} - x_i - u_i)^2, \quad \text{for } (a, b) = (1, 1), \quad (6.5)$$

$$V_k(0, -1) = \sum_{i=0}^{k-1} (x_{i+1} + u_i)^2, \quad \text{for } (a, b) = (0, -1). \quad (6.6)$$

The control applied at time  $k$  is

$$u_k = \tilde{\mu}_k(I_k) = \begin{cases} -\frac{x_k}{2} & \text{if } V_k(1, 1) < V_k(0, -1), \\ 0 & \text{if } V_k(1, 1) > V_k(0, -1). \end{cases}$$

Suppose the true parameters are  $\theta = (0, -1)$ . Then the true system evolves according to

$$x_{k+1} = -u_k + w_k. \quad (6.7)$$

If at time  $k$  the controller estimates incorrectly the parameters to be  $\hat{\theta} = (1, 1)$ , because  $V_k(\hat{\theta}) < V_k(\theta)$ , the control applied will be  $u_k = -x_k/2$  and the true closed-loop system will evolve according to

$$x_{k+1} = \frac{x_k}{2} + w_k. \quad (6.8)$$

On the other hand, the controller *thinks* (given the estimate  $\hat{\theta}$ ) that the closed-loop system will evolve according to

$$x_{k+1} = x_k + u_k + w_k = x_k - \frac{x_k}{2} + w_k - \frac{x_k}{2} + w_k, \quad (6.9)$$

so from Eqs. (6.7) and (6.8) we see that *under the control law  $u_k = -x_k/2$ , the closed-loop system evolves identically for both the true and the false values of the parameters* [cf. Eq. (6.4)].

To see what can go wrong, note that if  $V_k(\hat{\theta}) < V_k(\theta)$  for some  $k$  we will have, from Eqs. (6.5)-(6.9),

$$x_{k+1} + u_k = x_{k+1} - x_k - u_k,$$

so from Eqs. (6.5) and (6.6) we obtain

$$V_{k+1}(\hat{\theta}) < V_{k+1}(\theta).$$

Therefore, if  $V_1(\hat{\theta}) < V_1(\theta)$ , the least-squares identification method will yield the wrong estimate  $\hat{\theta}$  for every  $k$ . To see that this can happen with positive probability, note that, since the true system is  $x_{k+1} = -u_k + w_k$ , we have

$$\begin{aligned} V_1(\hat{\theta}) &= (x_1 - x_0 - u_0)^2 = (w_0 - x_0 - 2u_0)^2, \\ V_1(\theta) &= (x_1 - u_0)^2 = w_0^2. \end{aligned}$$

Therefore, the inequality  $V_1(\hat{\theta}) < V_1(\theta)$  is equivalent to

$$(x_0 + 2u_0)^2 < 2w_0(x_0 + 2u_0),$$

which will occur with positive probability since  $w_0$  is Gaussian.

The preceding examples illustrate that loss of identifiability is a serious problem that frequently arises in the context of certainty equivalent control.

#### 6.1.4 Self-Tuning Regulators

We described earlier the nature of the identifiability issue in certainty equivalent control: under closed-loop control, incorrect parameter estimates can make the system behave as if these estimates were correct [cf. Eq. (6.4)]. As a result, the identification scheme may lock onto false parameter values. This is not necessarily bad, however, since it may happen that the control law implemented on the basis of the false parameter values is near optimal. Indeed, through a fortuitous coincidence, it turns out that *in the practically important minimum variance control formulation (Section 5.3), when the parameter estimates converge, they typically converge to false values, but the resulting control law typically converges to the optimal*. We can get an idea about this phenomenon by means of an example.

##### Example 6.1.8

Consider the simplest ARMAX model:

$$y_{k-1} + ay_k = bu_k + \epsilon_{k+1}.$$

The minimum variance control law when  $a$  and  $b$  are known is

$$u_k = \mu_k(I_k) = \frac{a}{b}y_k.$$

Suppose now that  $a$  and  $b$  are not known but are identified on-line by means of some scheme. The control applied is

$$u_k = \frac{\hat{a}_k}{\hat{b}_k}y_k, \quad (6.10)$$

where  $\hat{a}_k$  and  $\hat{b}_k$  are the estimates obtained at time  $k$ . Then the difficulty with identifiability occurs when

$$\hat{a}_k \rightarrow \hat{a}, \quad \hat{b}_k \rightarrow \hat{b},$$

where  $\hat{a}$  and  $\hat{b}$  are such that the true closed-loop system given by

$$y_{k+1} + a y_k = \frac{b\hat{a}}{\hat{b}} y_k + \epsilon_{k+1}$$

coincides with the closed-loop system that the controller thinks is true on the basis of the estimates  $\hat{a}$  and  $\hat{b}$ . This latter system is

$$y_{k+1} = \epsilon_{k+1}.$$

For these two systems to be identical, we must have

$$\frac{a}{b} = \frac{\hat{a}}{\hat{b}},$$

which means that the control law (6.10) asymptotically becomes optimal despite the fact that the asymptotic estimates  $\hat{a}$  and  $\hat{b}$  may be incorrect.

Example 6.1.8 can be extended to the general ARMAX model of Section 5.3 with no delay:

$$y_k + \sum_{i=1}^m a_i y_{k-i} = \sum_{i=1}^m b_i u_{k-i} + \epsilon_k + \sum_{i=1}^m c_i \epsilon_{k-i}.$$

If the parameter estimates converge (regardless of the identification method used and regardless of whether the limit values are correct), then a minimum variance controller *thinks* that the closed-loop system is asymptotically

$$y_k = \epsilon_k.$$

Furthermore, parameter estimate convergence intuitively means that the true closed-loop system is also asymptotically  $y_k = \epsilon_k$ , and this is clearly the optimal closed-loop system. Results of this type have been proved in the literature in connection with several popular methods for parameter estimation. In fact, surprisingly, in some of these results, the model adopted by the controller is allowed to be incorrect to some extent.

One issue that we have not discussed is whether the parameter estimates indeed converge. A complete analysis of this issue is quite difficult. We refer to the survey paper by Kumar [Kum85], and the textbooks by Goodwin and Sin [GoS84], Kumar and Varaiya [KuV86], and Aström and Wittenmark [AsW90] for a discussion and sources on this subject. However, extensive simulations have shown that with proper implementation, these estimates typically converge for the type of systems likely to arise in many applications.

## 6.2 OPEN-LOOP FEEDBACK CONTROL

Generally, in a problem with imperfect state information, the performance of the optimal policy improves when extra information is available. However, the use of this information may make the DP calculation of the optimal policy intractable. This motivates an approximation, based on a more tractable computation that in part ignores the availability of extra information.

Let us consider the imperfect state information problem under the assumption of Section 5.4.1, which guarantees that the conditional state distribution is a sufficient statistic, i.e., that the probability distribution of the observation disturbance  $v_{k+1}$  depends explicitly only on the immediately preceding state, control, and system disturbance  $x_k, u_k, w_k$ , and not on  $x_{k-1}, \dots, x_0, u_{k-1}, \dots, u_0, w_{k-1}, \dots, w_0, v_{k-1}, \dots, v_0$ .

We introduce a suboptimal policy known as the *open-loop feedback controller* (OLFC), which uses the current information vector  $I_k$  to determine  $P_{x_k|I_k}$ . However, it calculates the control  $u_k$  as if no further measurements will be received, by using an open-loop optimization over the future evolution of the system. In particular,  $u_k$  is determined as follows:

- (1) Given the information vector  $I_k$ , compute the conditional probability distribution  $P_{x_k|I_k}$  (in the case of perfect state information, where  $I_k$  includes  $x_k$ , this step is unnecessary).
- (2) Find a control sequence  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{N-1}\}$  that solves the open-loop problem of minimizing

$$E \left\{ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, u_i, w_i) \mid I_k \right\}$$

subject to the constraints

$$x_{i+1} = f_i(x_i, u_i, w_i), \quad u_i \in U_i, \quad i = k, k+1, \dots, N-1.$$

- (3) Apply the control input

$$\bar{u}_k(I_k) = \bar{u}_k.$$

Thus the OLFC uses at time  $k$  the new measurement  $x_k$  to calculate the conditional probability distribution  $P_{x_k|I_k}$ . However, it selects the control input as if future measurements will be disregarded.

Similar to the CEC, the OLFC requires the solution of  $N$  optimal control problems. Each problem may again be solved by deterministic optimal control techniques. The computations, however, may be more complicated than those for the CEC, since now the cost involves an expectation with

respect to the uncertain quantities. The main difficulty in the implementation of the OLFC is the computation of  $P_{x_k|I_k}$ . In many cases one cannot compute  $P_{x_k|I_k}$  exactly, in which case some “reasonable” approximation scheme must be used. Of course, if we have perfect state information, this difficulty does not arise.

In any suboptimal control scheme, one would like to be assured that measurements are advantageously used. By this we mean that the scheme performs at least as well as any open-loop policy that uses a sequence of controls that is independent of the values of the measurements received. An optimal open-loop policy can be obtained by finding a sequence  $\{u_0^*, u_1^*, \dots, u_{N-1}^*\}$  that minimizes

$$\bar{J}(u_0, u_1, \dots, u_{N-1}) = E \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\}$$

subject to the constraints

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad u_k \in U_k, \quad k = 0, 1, \dots, N-1.$$

A nice property of the OLFC is that it performs at least as well as an optimal open-loop policy, as shown by the following proposition. By contrast, the CEC does not have this property (for a one-stage problem, the optimal open-loop controller and the OLFC are both optimal, but the CEC may be strictly suboptimal; see also Exercise 6.2).

**Proposition 6.2.1:** The cost  $J_{\bar{\pi}}$  corresponding to an OLFC satisfies

$$J_{\bar{\pi}} \leq J_0^*, \quad (6.11)$$

where  $J_0^*$  is the cost corresponding to an optimal open-loop policy.

**Proof:** We assume throughout the proof that all expected values appearing are well defined and finite, and that the minimum in the following Eq. (6.14) is attained for every  $I_k$ . Let  $\bar{\pi} = \{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{N-1}\}$  be the OLFC. Its cost is given by

$$J_{\bar{\pi}} = E_{z_0} \{ \bar{J}_0(I_0) \} = E_{z_0} \{ \bar{J}_0(z_0) \}, \quad (6.12)$$

where  $\bar{J}_0$  is obtained from the recursive algorithm

$$\begin{aligned} \bar{J}_{N-1}(I_{N-1}) &= E_{x_{N-1}, w_{N-1}} \left\{ g_N(f_{N-1}(x_{N-1}, \bar{u}_{N-1}(I_{N-1}), w_{N-1})) \right. \\ &\quad \left. + g_{N-1}(x_{N-1}, \bar{u}_{N-1}(I_{N-1}), w_{N-1}) \mid I_{N-1} \right\}, \end{aligned}$$

$$\begin{aligned}\bar{J}_k(I_k) &= \underset{x_k, w_k, u_{k+1}}{E} \left\{ g_k(x_k, \bar{\mu}_k(I_k), w_k) \right. \\ &\quad \left. + \bar{J}_{k+1}(I_k, h_{k+1}(f_k(x_k, \bar{\mu}_k(I_k), w_k), \bar{\mu}_k(I_k), v_{k+1}), \bar{\mu}_k(I_k)) \mid I_k \right\}, \\ k &= 0, \dots, N-1,\end{aligned}\quad (6.13)$$

where  $h_k$  is the function involved in the measurement equation as in the basic problem with imperfect state information of Section 5.1.

Consider the functions  $J_k^c(I_k)$ ,  $k = 0, 1, \dots, N-1$ , defined by

$$J_k^c(I_k) = \min_{u_i \in U_i} E \left\{ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, u_i, w_i) \mid I_k \right\}. \quad (6.14)$$

The minimization problem in this equation is the one that must be solved at time  $k$  in order to calculate the control  $\bar{\mu}_k(I_k)$  of the OLFC. Clearly,  $J_k^c(I_k)$  can be interpreted as the optimal open-loop cost from time  $k$  to time  $N$  when the current information vector is  $I_k$ . It can be seen that

$$E_{z_0} \{ J_0^c(z_0) \} \leq J_0^*, \quad (6.15)$$

since  $J_0^*$  is the minimum over  $u_0, \dots, u_{N-1}$  of the total expected cost and can be written as

$$\min_{u_0, \dots, u_{N-1}} E_{z_0} \{ E \{ \text{cost} \mid z_0 \} \},$$

while  $E_{z_0} \{ J_0^c(z_0) \}$  can be written as

$$E_{z_0} \left\{ \min_{u_0, \dots, u_{N-1}} E \{ \text{cost} \mid z_0 \} \right\}$$

(we generally have  $E \{ \min[\cdot] \} \leq \min[E \{ \cdot \}]$ ). We will prove that

$$\bar{J}_k(I_k) \leq J_k^c(I_k), \quad \text{for all } I_k \text{ and } k. \quad (6.16)$$

Then from Eqs. (6.12), (6.15), and (6.16), it will follow that

$$J_{\bar{\pi}} \leq J_0^*,$$

which is the relation to be proved. We show Eq. (6.16) by induction.

By the definition of the OLFC and Eq. (6.14), we have

$$\bar{J}_{N-1}(I_{N-1}) = J_{N-1}^c(I_{N-1}), \quad \text{for all } I_{N-1},$$

and hence Eq. (6.16) holds for  $k = N-1$ . Assume that

$$\bar{J}_{k+1}(I_{k+1}) \leq J_{k+1}^c(I_{k+1}), \quad \text{for all } I_{k+1}. \quad (6.17)$$

Then from Eqs. (6.13), (6.14), and (6.17), we have

$$\begin{aligned}\bar{J}_k(I_k) &= \underset{x_k, w_k, v_{k+1}}{E} \left\{ g_k(x_k, \bar{\mu}_k(I_k), w_k) \right. \\ &\quad \left. + \bar{J}_{k+1}(I_k, h_{k+1}(f_k(x_k, \bar{\mu}_k(I_k), w_k), \bar{\mu}_k(I_k), v_{k+1}), \bar{\mu}_k(I_k)) \mid I_k \right\} \\ &\leq \underset{x_k, w_k, v_{k+1}}{E} \left\{ g_k(x_k, \bar{\mu}_k(I_k), w_k) \right. \\ &\quad \left. + J_{k+1}^c(I_k, h_{k+1}(f_k(x_k, \bar{\mu}_k(I_k), w_k), \bar{\mu}_k(I_k), v_{k+1}), \bar{\mu}_k(I_k)) \mid I_k \right\} \\ &= \underset{x_k, w_k, v_{k+1}}{E} \left\{ \min_{\substack{u_i \in U_i \\ i=k+1, \dots, N-1 \\ x_{i+1}=f_i(x_i, u_i, w_i) \\ i=k+1, \dots, N-1 \\ x_{k+1}=\bar{f}_k(x_k, \bar{\mu}_k(I_k), w_k)}} \left\{ g_k(x_k, \bar{\mu}_k(I_k), w_k) \right. \right. \\ &\quad \left. \left. + \sum_{i=k+1}^{N-1} g_i(x_i, u_i, w_i) + g_N(x_N) \mid I_{k+1} \right\} \mid I_k \right\} \\ &\leq \min_{\substack{u_i \in U_i \\ i=k+1, \dots, N-1 \\ x_{i+1}=f_i(x_i, u_i, w_i) \\ i=k+1, \dots, N-1 \\ x_{k+1}=\bar{f}_k(x_k, \bar{\mu}_k(I_k), w_k)}} \left\{ g_N(x_N) + g_k(x_k, \bar{\mu}_k(I_k), w_k) \right\} \\ &\quad + \sum_{i=k+1}^{N-1} g_i(x_i, u_i, w_i) \mid I_k \\ &= J_k^c(I_k).\end{aligned}$$

The second inequality follows by interchanging expectation and minimization (since we generally have  $E \{ \min[\cdot] \} \leq \min[E \{ \cdot \}]$ ) and by "integrating out"  $v_{k+1}$ . The last equality follows from the definition of OLFC. Thus Eq. (6.16) is proved for all  $k$  and the desired result is shown. Q.E.D.

The preceding proposition shows that the OLFC uses the measurements advantageously even though it selects at each period the present control input as if no further measurements will be taken in the future. It is worth noting that by Eq. (6.16),  $J_k^c(I_k)$ , which is the calculated open-loop optimal cost from time  $k$  to time  $N$ , provides a readily obtainable performance bound for the OLFC.

### Partial Open-Loop Feedback Control

A form of suboptimal control that is intermediate between the optimal feedback controller and the OLFC is provided by a generalization of the OLFC called the *partial open-loop feedback controller* (POLFC for short). This controller uses past measurements to compute  $P_{x|I_k}$ , but calculates the control input on the basis that *some* (but not necessarily all) of the

measurements will in fact be taken in the future, and the remaining measurements will not be taken.

This method often allows one to deal with those measurements that are troublesome and complicate the solution. As an example consider an inventory problem such as the one considered in Section 4.2, where forecasts in the form of a probability distribution of each of the future demands become available over time. A reasonable form of POLFC calculates at each stage an optimal  $(s, S)$  policy based on the current forecast of future demands and follows this policy until a new forecast becomes available. When this happens, the current policy is abandoned in favor of a new one that is calculated on the basis of the new probability distribution of future demands, etc. Thus the complications due to the forecasts are bypassed at the expense of suboptimality of the policy obtained.

We note that an analog of Prop. 6.2.1 can be shown for the POLFC (see Bertsekas [Ber76]). In fact the corresponding error bound is potentially much better than the bound (6.11), reflecting the fact that the POLFC takes into account the future availability of some of the measurements.

We will discuss further the idea of ignoring a portion of the information for the purpose of obtaining a tractable suboptimal policy in Section 6.5.3. There we will generalize the OLFC and the POLFC by embedding them within a more general suboptimal scheme.

### 6.3 LIMITED LOOKAHEAD POLICIES

An effective way to reduce the computation required by DP is to truncate the time horizon and use at each stage a decision based on lookahead of a small number of stages. The simplest possibility is to use a *one-step lookahead policy* whereby at stage  $k$  and state  $x_k$  one uses the control  $\bar{u}_k(x_k)$ , which attains the minimum in the expression

$$\min_{u_k \in U_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}, \quad (6.18)$$

where  $\tilde{J}_{k+1}$  is some approximation of the true cost-to-go function  $J_{k+1}$ , with  $\tilde{J}_N = g_N$ . Similarly, a *two-step lookahead policy* applies at time  $k$  and state  $x_k$ , the control  $\bar{u}_k(x_k)$  attaining the minimum in the preceding equation, where now  $\tilde{J}_{k+1}$  is obtained itself on the basis of a one-step lookahead approximation. In other words, for all possible states  $x_{k+1}$  that can be generated via the system equation starting from  $x_k$ ,

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

we have

$$\begin{aligned}\tilde{J}_{k+1}(x_{k+1}) &= \min_{u_{k+1} \in U_{k+1}(x_{k+1})} E\left\{g_{k+1}(x_{k+1}, u_{k+1}, w_{k+1})\right. \\ &\quad \left.+ \tilde{J}_{k+2}(f_{k+1}(x_{k+1}, u_{k+1}, w_{k+1}))\right\},\end{aligned}$$

where  $\tilde{J}_{k+2}$  is some approximation of the cost-to-go function  $J_{k+2}$ . Policies with lookahead of more than two stages are similarly defined.

Note that the limited lookahead approach can be used equally well when the horizon is infinite. One simply uses as the terminal cost-to-go function an approximation to the optimal cost of the infinite horizon problem that starts at the end of the lookahead. Thus the following discussion, with a few straightforward modifications, applies to infinite horizon problems as well.

Given the approximations  $\tilde{J}_k$  to the optimal costs-to-go, the computational savings of the limited lookahead approach are evident. For a one-step lookahead policy, only a single minimization problem has to be solved per stage, while in a two-step policy the corresponding number of minimization problems is one plus the number of all possible next states  $x_{k+1}$  that can be generated from the current state  $x_k$ .

However, even with readily available cost-to-go approximations  $\tilde{J}_k$ , the minimization over  $u_k \in U_k(x_k)$  in the calculation of the one-step lookahead control [cf. Eq. (6.18)] may involve substantial computation. In a variant of the method that aims at reducing this computation, the minimization is done over a subset

$$\overline{U}_k(x_k) \subset U_k(x_k).$$

Thus, the control  $\bar{u}_k(x_k)$  used in this variant is one that attains the minimum in the expression

$$\min_{u_k \in \overline{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}. \quad (6.19)$$

A practical example of this approach is when by using some heuristic or approximate optimization, we identify a subset  $\overline{U}_k(x_k)$  of promising controls, and to save computation, we restrict attention to this subset in the one-step lookahead minimization.

#### 6.3.1 Performance Bounds for Limited Lookahead Policies

Let us denote by  $\bar{J}_k(x_k)$  the expected cost-to-go incurred by a limited lookahead policy  $\{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{N-1}\}$  starting from state  $x_k$  at time  $k$  [ $\bar{J}_k(x_k)$  should be distinguished from  $\tilde{J}_k(x_k)$ , the approximation of the cost-to-go that is used to compute the limited lookahead policy via the minimization in Eq. (6.19)]. It is generally difficult to evaluate analytically the functions  $\bar{J}_k$ , even when the functions  $\tilde{J}_k$  are readily available. We thus aim to obtain some estimates of  $\bar{J}_k(x_k)$ . The following proposition gives a condition under which the one-step lookahead policy achieves a cost  $\bar{J}_k(x_k)$  that is better than the approximation  $\tilde{J}_k(x_k)$ . The proposition also provides a readily computable upper bound to  $\bar{J}_k(x_k)$ .

**Proposition 6.3.1:** Assume that for all  $x_k$  and  $k$ , we have

$$\min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \leq \tilde{J}_k(x_k). \quad (6.20)$$

Then the cost-to-go functions  $\bar{J}_k$  corresponding to a one-step lookahead policy that uses  $\tilde{J}_k$  and  $\bar{U}_k(x_k)$  [cf. Eq. (6.19)] satisfy for all  $x_k$  and  $k$

$$\bar{J}_k(x_k) \leq \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}. \quad (6.21)$$

**Proof:** For  $k = 0, \dots, N-1$ , denote

$$\hat{J}_k(x_k) = \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}, \quad (6.22)$$

and let  $\hat{J}_N = g_N$ . We must show that for all  $x_k$  and  $k$ , we have

$$\bar{J}_k(x_k) \leq \hat{J}_k(x_k).$$

We use backwards induction on  $k$ . In particular, we have  $\bar{J}_N(x_N) = \hat{J}_N(x_N) = \hat{J}_N(x_N) = g_N(x_N)$  for all  $x_N$ . Assuming that  $\bar{J}_{k+1}(x_{k+1}) \leq \hat{J}_{k+1}(x_{k+1})$  for all  $x_{k+1}$ , we have

$$\begin{aligned} \bar{J}_k(x_k) &= E\left\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \bar{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\ &\leq E\left\{g(x_k, \bar{\mu}_k(x_k), w_k) + \hat{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\ &\leq E\left\{g(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\ &= \hat{J}_k(x_k), \end{aligned}$$

for all  $x_k$ . The first equality above follows from the DP algorithm that defines the costs-to-go  $\bar{J}_k$  of the limited lookahead policy, while the first inequality follows from the induction hypothesis, and the second inequality follows from the assumption (6.20). This completes the induction proof. Q.E.D.

Note that by Eq. (6.21), the value  $\hat{J}_k(x_k)$  of Eq. (6.22), which is the calculated one-step lookahead cost from state  $x_k$  at time  $k$ , provides a readily obtainable performance bound for the cost-to-go  $\bar{J}_k(x_k)$  of the one-step lookahead policy. Furthermore, using also the assumption (6.20), we obtain for all  $x_k$  and  $k$ ,

$$\bar{J}_k(x_k) \leq \hat{J}_k(x_k),$$

i.e., the cost-to-go of the one-step lookahead policy is no greater than the lookahead approximation on which it is based. The critical assumption (6.20) in Prop. 6.3.1 can be verified in a few interesting special cases, as indicated by the following examples.

### Example 6.3.1 (Rollout Algorithm)

Suppose that  $\tilde{J}_k(x_k)$  is the cost-to-go of some given (suboptimal) heuristic policy  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$  and that the set  $\bar{U}_k(x_k)$  contains the control  $\mu_k(x_k)$  for all  $x_k$  and  $k$ . The resulting one-step lookahead algorithm is called the *rollout algorithm* and will be discussed extensively in Section 6.4. From the DP algorithm (restricted to the given policy  $\pi$ ), we have

$$\tilde{J}_k(x_k) = E\left\{g_k(x_k, \mu_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \mu_k(x_k), w_k))\right\},$$

which in view of the assumption  $\mu_k(x_k) \in \bar{U}_k(x_k)$ , yields

$$\tilde{J}_k(x_k) \geq \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}.$$

Thus, the assumption of Prop. 6.3.1 is satisfied, and it follows that the rollout algorithm performs better than the heuristic on which it is based, starting from any state and stage.

### Example 6.3.2 (Rollout Algorithm with Multiple Heuristics)

Consider a scheme that is similar to the one of the preceding example, except that  $\tilde{J}_k(x_k)$  is the minimum of the cost-to-go functions corresponding to  $m$  heuristics, i.e.,

$$\tilde{J}_k(x_k) = \min\{J_{\pi_1, k}(x_k), \dots, J_{\pi_m, k}(x_k)\},$$

where for each  $j$ ,  $J_{\pi_j, k}(x_k)$  is the cost-to-go of a policy  $\pi_j = \{\mu_{j,0}, \dots, \mu_{j,N-1}\}$ , starting from state  $x_k$  at stage  $k$ . From the DP algorithm, we have, for all  $j$ ,

$$J_{\pi_j, k}(x_k) = E\left\{g_k(x_k, \mu_{j,k}(x_k), w_k) + J_{\pi_j, k+1}(f_k(x_k, \mu_{j,k}(x_k), w_k))\right\},$$

from which, using the definition of  $\tilde{J}_k$ , it follows that

$$\begin{aligned} J_{\pi_j, k}(x_k) &\geq E\left\{g_k(x_k, \mu_{j,k}(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \mu_{j,k}(x_k), w_k))\right\} \\ &\geq \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}. \end{aligned}$$

Taking the minimum of the left-hand side over  $j$ , we obtain

$$\tilde{J}_k(x_k) \geq \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\}.$$

Thus, Prop. 6.3.1 implies that the one-step lookahead algorithm based on the heuristic algorithms' costs-to-go  $J_{\pi_1,k}(x_k), \dots, J_{\pi_m,k}(x_k)$  performs better than all these heuristics, starting from any state and stage.

Generally, the cost-to-go approximation functions  $\tilde{J}_k$  need not satisfy the assumption (6.20) of Prop. 6.3.1. The following proposition does not require this assumption. It is useful in some contexts, including the case where the minimization involved in the calculation in the one-step-lookahead policy is not exact.

**Proposition 6.3.2:** Let  $\tilde{J}_k$ ,  $k = 0, 1, \dots, N$ , be functions of  $x_k$  with  $\tilde{J}_N(x_N) = g_N(x_N)$  for all  $x_N$ , and let  $\pi = \{\bar{\mu}_0, \bar{\mu}_1, \dots, \bar{\mu}_{N-1}\}$  be a policy such that for all  $x_k$  and  $k$ , we have

$$E\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \leq \tilde{J}_k(x_k) + \delta_k, \quad (6.23)$$

where  $\delta_0, \delta_1, \dots, \delta_{N-1}$  are some scalars. Then for all  $x_k$  and  $k$ , we have

$$J_{\pi,k}(x_k) \leq \tilde{J}_k(x_k) + \sum_{i=k}^{N-1} \delta_i,$$

where  $J_{\pi,k}(x_k)$  is the cost-to-go of  $\pi$  starting from state  $x_k$  at stage  $k$ .

**Proof:** We use backwards induction on  $k$ . In particular, we have  $J_{\pi,N}(x_N) = \tilde{J}_N(x_N) = g_N(x_N)$  for all  $x_N$ . Assuming that

$$J_{\pi,k+1}(x_{k+1}) \leq \tilde{J}_{k+1}(x_{k+1}) + \sum_{i=k+1}^{N-1} \delta_i$$

for all  $x_{k+1}$ , we have

$$\begin{aligned} J_{\pi,k}(x_k) &= E\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + J_{\pi,k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \\ &\leq E\{g(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} + \sum_{i=k+1}^{N-1} \delta_i \\ &\leq \tilde{J}_k(x_k) + \delta_k + \sum_{i=k+1}^{N-1} \delta_i, \end{aligned}$$

for all  $x_k$ . The first equality above follows from the DP algorithm that defines the costs-to-go  $J_{\pi,k}$  of  $\pi$ , while the first inequality follows from the induction hypothesis, and the second inequality follows from the assumption (6.23). This completes the induction proof. **Q.E.D.**

### Example 6.3.3 (Certainty Equivalent Control)

Consider the CEC for the case of a perfect state information problem, where each disturbance  $w_k$  is fixed at a nominal value  $\bar{w}_k$ ,  $k = 0, \dots, N-1$ , which is independent of  $x_k$  and  $u_k$ . Consider the optimal value of the problem solved by the CEC at state  $x_k$  and stage  $k$ ,

$$\tilde{J}_k(x_k) = \min_{\substack{x_{i+1} = f_i(x_i, u_i, \bar{w}_i) \\ u_i \in U_i(x_i), i = k, \dots, N-1}} \left[ g_N(x_N) + \sum_{i=k}^{N-1} g_i(x_i, u_i, \bar{w}_i) \right],$$

and let  $\tilde{J}_N(x_N) = g_N(x_N)$  for all  $x_N$ . Recall that the CEC applies the control  $\bar{\mu}_k(x_k) = \bar{u}_k$  after finding an optimal control sequence  $\{\bar{u}_k, \dots, \bar{u}_{N-1}\}$  for the deterministic problem in the right-hand side above. Note also that the following DP equation

$$\tilde{J}_k(x_k) = \min_{u_k \in U_k(x_k)} \left[ g_k(x_k, u_k, \bar{w}_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, \bar{w}_k)) \right]$$

holds, and that the control  $\bar{u}_k$  applied by the CEC minimizes in the right-hand side.

Let us now apply Prop. 6.3.2 to derive a performance bound for the CEC. We have for all  $x_k$  and  $k$ ,

$$\begin{aligned} \tilde{J}_k(x_k) &= g_k(x_k, \bar{\mu}_k(x_k), \bar{w}_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), \bar{w}_k)) \\ &= E\{g(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} - \gamma_k(x_k) \end{aligned}$$

where  $\gamma_k(x_k)$  is defined by

$$\begin{aligned} \gamma_k(x_k) &= E\{g(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \\ &\quad - g_k(x_k, \bar{\mu}_k(x_k), \bar{w}_k) - \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), \bar{w}_k)). \end{aligned}$$

It follows that

$$E\{g(x_k, \bar{\mu}_k(x_k), w_k) + \tilde{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \leq \tilde{J}_k(x_k) + \delta_k,$$

where

$$\delta_k = \max_{\pi_k} \gamma_k(x_k),$$

and by Prop. 6.3.2, we obtain the following bound for the cost-to-go function  $\tilde{J}_k(x_k)$  of the CEC:

$$\tilde{J}_k(x_k) \leq \tilde{J}_k(x_k) + \sum_{i=k}^{N-1} \delta_i.$$

The preceding performance bound is helpful when it can be shown that  $\delta_k \leq 0$  for all  $k$ , in which case we have  $\tilde{J}_k(x_k) \leq \tilde{J}_k(x_k)$  for all  $x_k$  and  $k$ . This is true for example if for all  $x_k$  and  $u_k$ , we have

$$E\{g(x_k, u_k, w_k)\} \leq g_k(x_k, u_k, \bar{w}_k),$$

and

$$E\{\bar{J}_{k+1}(f_k(x_k, u_k, w_k))\} \leq \bar{J}_{k+1}(f_k(x_k, u_k, \bar{w}_k)).$$

The most common way to assert that inequalities of this type hold is via some kind of concavity assumptions; for example, the inequalities hold if the state, control, and disturbance spaces are Euclidean spaces,  $\bar{w}_k$  is the expected value of  $w_k$ , and the functions  $g(x_k, u_k, \cdot)$  and  $\bar{J}_{k+1}(f_k(x_k, u_k, \cdot))$ , viewed as functions of  $w_k$ , are concave (this is known as Jensen's inequality, and at least in the case where  $w_k$  takes a finite number of values, follows easily from the definition of concavity). It can be shown that the concavity conditions just described are guaranteed if the system is linear with respect to  $x_k$  and  $w_k$ , the cost functions  $g_k$  are concave with respect to  $x_k$  and  $w_k$  for each fixed  $u_k$ , the terminal cost function  $g_N$  is concave, and the control constraint sets  $U_k$  do not depend on  $x_k$ .

### 6.3.2 Computational Issues in Limited Lookahead

We now discuss the computation of the cost-to-go approximations and the corresponding minimization of the one-step lookahead costs.

#### Minimization Using Nonlinear Programming

One approach to obtain the control  $\bar{u}_k(x_k)$  used by the one-step lookahead policy is to exhaustively calculate and compare the one-step lookahead costs of all the controls in the set  $\bar{U}_k(x_k)$ . In some cases, there is a more efficient alternative, which is to solve a suitable nonlinear programming problem. In particular, if the control space is the Euclidean space  $\mathbb{R}^m$ , then for a one-step lookahead control calculation, we are faced with a minimization over a subset of  $\mathbb{R}^m$ , which may be approached by continuous optimization/nonlinear programming techniques.

It turns out that even a multistage lookahead control calculation can be approached by nonlinear programming. In particular, assume that the disturbance can take a finite number of values, say  $r$ . Then, it can be shown that for a given initial state, an  $l$ -stage perfect state information problem (which corresponds to an  $l$ -step lookahead control calculation) can be formulated as a nonlinear programming problem with  $m(1 + r^{l-1})$  variables. We illustrate this by means of an important example where  $l = 2$  and then discuss the general case.

#### Example 6.3.4 (Two-Stage Stochastic Programming)

Here we want to find an optimal two-stage decision rule for the following situation: In the first stage we will choose a vector  $u_0$  from a subset  $U_0 \subset \mathbb{R}^m$  with cost  $g_0(u_0)$ . Then an uncertain event represented by a random variable  $w$  will occur, where  $w$  will take one of the values  $w^1, \dots, w^r$  with corresponding probabilities  $p^1, \dots, p^r$ . We will know the value  $w^j$  once it occurs, and we must

then choose a vector  $u_1^j$  from a subset  $U_1(u_0, w^j) \subset \mathbb{R}^m$  at a cost  $g_1(u_1^j, w^j)$ . The objective is to minimize the expected cost

$$g_0(u_0) + \sum_{j=1}^r p^j g_1(u_1^j, w^j)$$

subject to

$$u_0 \in U_0, \quad u_1^j \in U_1(u_0, w^j), \quad j = 1, \dots, r.$$

This is a nonlinear programming problem of dimension  $m(1 + r)$  (the optimization variables are  $u_0, u_1^1, \dots, u_1^r$ ). It can also be viewed as a two-stage perfect state information problem, where  $x_1 = w_0$  is the state equation,  $w_0$  can take the values  $w^1, \dots, w^r$  with probabilities  $p^1, \dots, p^r$ , the cost of the first stage is  $g_0(u_0)$ , and the cost of the second stage is  $g_1(x_1, u_1)$ .

The preceding example can be generalized. Consider the basic problem of Chapter 1 for the case where there are only two stages ( $l = 2$ ) and the disturbances  $w_0$  and  $w_1$  can independently take one of the  $r$  values  $w^1, \dots, w^r$  with corresponding probabilities  $p^1, \dots, p^r$ . The optimal cost function  $J_0(x_0)$  is given by the two-stage DP algorithm

$$\begin{aligned} J_0(x_0) = \min_{u_0 \in U_0(x_0)} & \left[ \sum_{j=1}^r p^j \left\{ g_0(x_0, u_0, w^j) \right. \right. \\ & + \min_{u_1^j \in U_1(f_0(x_0, u_0, w^j))} \left[ \sum_{i=1}^r p^i \left\{ g_1(f_0(x_0, u_0, w^j), u_1^j, w^i) \right. \right. \\ & \left. \left. : g_2(f_1(f_0(x_0, u_0, w^j), u_1^j, w^i)) \right\} \right] \left. \right]. \end{aligned}$$

This DP algorithm is equivalent to solving the nonlinear programming problem

$$\begin{aligned} \text{minimize} & \sum_{j=1}^r p^j \left\{ g_0(x_0, u_0, w^j) + \sum_{i=1}^r p^i \left\{ g_1(f_0(x_0, u_0, w^j), u_1^j, w^i) \right. \right. \\ & \left. \left. - g_2(f_1(f_0(x_0, u_0, w^j), u_1^j, w^i)) \right\} \right\} \end{aligned}$$

subject to  $u_0 \in U_0(x_0)$ ,  $u_1^j \in U_1(f_0(x_0, u_0, w^j))$ ,  $j = 1, \dots, r$ .

If the controls  $u_0$  and  $u_1$  are elements of  $\mathbb{R}^m$ , the number of variables in the above problem is  $m(1 + r)$ . More generally, for an  $l$ -stage perfect state information problem a similar reformulation as a nonlinear programming problem requires  $m(1 + r^{l-1})$  variables. Thus if the number of lookahead stages is relatively small, a nonlinear programming approach may be the preferred option in calculating suboptimal limited lookahead policies.

### Choosing the Approximate Cost-to-Go

A key issue in implementing a limited lookahead policy is the selection of the cost-to-go approximation at the final step. It may appear important at first that the true cost-to-go function should be approximated well over the range of relevant states; however, this is not necessarily true. What is important is that the *cost-to-go differentials (or relative values) be approximated well*; that is, for an  $l$ -step lookahead policy it is important to have

$$\tilde{J}_{k+l}(x) - \tilde{J}_{k+l}(x') \approx J_{k+l}(x) - J_{k+l}(x'),$$

for any two states  $x$  and  $x'$  that can be generated  $l$  steps ahead from the current state. For example, if equality were to hold above for all  $x, x'$ , then  $\tilde{J}_{k+l}(x)$  and  $J_{k+l}(x)$  would differ by the same constant for each relevant  $x$  and the  $l$ -step lookahead policy would be optimal.

The manner in which the cost-to-go approximation is selected depends very much on the problem being solved. There is a wide variety of possibilities here. We will discuss three such approaches:

- (a) *Problem Approximation*: The idea here is to approximate the optimal cost-to-go with some cost derived from a related but simpler problem (for example the optimal cost-to-go of that problem). This possibility is discussed and is illustrated with examples in Sections 6.3.3 and 6.3.4.
- (b) *Parametric Cost-to-Go Approximation*: The idea here is to approximate the optimal cost-to-go with a function of a suitable parametric form, whose parameters are tuned by some heuristic or systematic scheme. This possibility is discussed in Section 6.3.5 and is illustrated using the computer chess paradigm. Additional methods of this type are discussed in Vol. II.
- (c) *Rollout Approach*: Here the optimal cost-to-go is approximated by the cost of some suboptimal policy, which is calculated either analytically, or more commonly, by simulation. Generally, if a reasonably good suboptimal policy is known (e.g., a certainty equivalent or open-loop-feedback controller, or some other problem-dependent heuristic), it can be used to obtain a cost-to-go approximation. This approach is also particularly well-suited for deterministic and combinatorial problems. It is discussed at length in Section 6.4.

#### 6.3.3 Problem Approximation - Enforced Decomposition

An often convenient approach for cost-to-go approximation is based on solution of a simpler problem that is tractable computationally or analytically. Here is an illustrative example, involving a convenient modification of the probabilistic structure of the problem.

### Example 6.3.5

Consider the problem of an unscrupulous innkeeper who charges one of  $m$  different rates  $r_1, \dots, r_m$  for a room as the day progresses, depending on whether he has many or few vacancies, so as to maximize his expected total income during the day (Exercise 1.25 in Chapter 1). A quote of a rate  $r_i$  is accepted with probability  $p_i$  and is rejected with probability  $1 - p_i$ , in which case the customer departs, never to return during that day. When the number  $y$  of customers that will ask for a room during the rest of the day (including the customer currently asking for a room) is known and the number of vacancies is  $x$ , the optimal expected income  $J(x, y)$  of the innkeeper is given by the DP recursion

$$J(x, y) = \max_{i=1, \dots, m} [p_i(r_i + J(x-1, y-1)) + (1-p_i)J(x, y-1)],$$

for all  $x \geq 1$  and  $y \geq 1$ , with initial conditions

$$J(x, 0) = J(0, y) = 0, \quad \text{for all } x \text{ and } y.$$

On the other hand, when the innkeeper does not know  $y$  at the times of decision, but instead only has a probability distribution for  $y$ , it can be seen that the problem becomes a difficult imperfect state information problem. Yet a reasonable one-step lookahead policy is based on approximating the optimal cost-to-go of subsequent decisions with  $J(x-1, \bar{y}-1)$  or  $J(x, \bar{y}-1)$ , where the function  $J$  is calculated by the above recursion and  $\bar{y}$  is the closest integer to the expected value of  $y$ . In particular, according to this one-step lookahead policy, when the innkeeper has a number of vacancies  $x \geq 1$ , he quotes to the current customer the rate that maximizes  $p_i(r_i + J(x-1, \bar{y}-1) - J(x, \bar{y}-1))$ .

The preceding example is based on replacing the problem uncertainty (the random variable  $y$ ) with a “certainty equivalent” (the scalar  $\bar{y}$ ). The next example describes a generalization of this type of approximation, based on simplifying the stochastic structure of the problem.

### Example 6.3.6 (Approximation Using Scenarios)

One possibility to approximate the optimal cost-to-go is to use certainty equivalence, in the spirit of Section 6.1. In particular, for a given state  $x_{k+1}$  at time  $k+1$ , we fix the remaining disturbances at some nominal values  $\bar{w}_{k+1}, \dots, \bar{w}_{N-1}$ , and we compute an optimal control trajectory starting from  $x_{k+1}$  at time  $k+1$ . The corresponding cost, denoted by  $\tilde{J}_{k+1}(x_{k+1})$ , is used to approximate the optimal cost-to-go  $J_{k+1}(x_{k+1})$  for the purpose of computing the corresponding one-step lookahead policy. Thus to compute the one-step lookahead control at state  $x_k$ , we need to solve a deterministic optimal control problem from all possible next states  $f_k(x_k, u_k, w_k)$  and to evaluate the corresponding optimal cost  $\tilde{J}_{k+1}(f_k(x_k, u_k, w_k))$  based on the nominal values of the uncertainty.

A simpler but less effective variant of this approach is to compute  $\tilde{J}_{k+1}(x_{k+1})$  as the cost-to-go of a given heuristic (rather than optimal) policy

for the deterministic problem that corresponds to the nominal values of the uncertainty and the starting state  $x_{k+1}$ . The advantage of using certainty equivalence here is that the potentially costly calculation of the expected value of the cost is replaced by a single state-control trajectory calculation.

The certainty equivalent approximation involves a single nominal trajectory of the remaining uncertainty. To strengthen this approach, it is natural to consider multiple trajectories of the uncertainty, called *scenarios*, and to construct an approximation to the optimal cost-to-go that involves, for every one of the scenarios, the cost of either an optimal or a given heuristic policy. Mathematically, we assume that we have a method, which at a given state  $x_{k+1}$ , generates  $M$  uncertainty sequences

$$w^m(x_{k+1}) = (w_{k+1}^m, \dots, w_{N-1}^m), \quad m = 1, \dots, M.$$

These are the scenarios considered at state  $x_{k+1}$ . The cost  $J_{k+1}(x_{k+1})$  is approximated by

$$\tilde{J}_{k+1}(x_{k+1}, r) = r_0 + \sum_{m=1}^M r_m C_m(x_{k+1}), \quad (6.24)$$

where  $r = (r_0, r_1, \dots, r_M)$  is a vector of parameters, and  $C_m(x_{k+1})$  is the cost corresponding to an occurrence of the scenario  $w^m(x_{k+1})$ , when starting from state  $x_{k+1}$  and using either an optimal or a given heuristic policy.

The parameters  $r_0, r_1, \dots, r_M$  may depend on the time index, and in more sophisticated schemes, they may depend on some characteristics of the state (see our subsequent discussion of feature-based architectures in Section 6.3.5). We may interpret the parameter  $r_m$  as an "aggregate weight" that encodes the aggregate effect on the cost-to-go function of uncertainty sequences that are similar to the scenario  $w^m(x_{k+1})$ . Note that, if  $r_0 = 0$ , the approximation (6.24) may also be viewed as a calculation by *limited simulation*, based on just the  $M$  scenarios  $w^m(x_{k+1})$ , and using the weights  $r_m$  as "aggregate probabilities." One difficulty with this approach is that we have to choose the parameters  $(r_0, r_1, \dots, r_M)$ . For this, we may either use some heuristic scheme based on trial and error, or some of the more systematic schemes of neuro-dynamic programming, discussed in Vol. II.

We finally mention a variation of the scenario-based approximation method, whereby only a portion of the future uncertain quantities are fixed at nominal scenario values, while the remaining uncertain quantities are explicitly viewed as random. The cost of scenario  $m$  at state  $x_{k+1}$  is now a random variable, and the quantity  $C_m(x_{k+1})$  used in Eq. (6.24) should be the *expected* cost of this random variable. This variation is appropriate and makes practical sense as long as the computation of the corresponding expected scenario costs  $C_m(x_{k+1})$  is convenient.

### Enforced Decomposition of Weakly Coupled Systems

The simplification/approximation approach is often well-suited for problems that involve a number of subsystems that may be coupled through

the system equation, or the cost function, or the control constraints, but the degree of coupling is "relatively weak." It is difficult to define precisely what constitutes "weak coupling," but in specific problem contexts, usually this type of structure is easily recognized. For such problems it is often sensible to introduce approximations by artificially decoupling the subsystems in some way, thereby creating either a simpler problem or a simpler cost calculation, where subsystems can be dealt with in isolation. There are a number of different ways to effect this type of artificial decomposition, and the best approach is often problem-dependent.

As an example consider a deterministic problem, where the control  $u_k$  at time  $k$  consists of  $m$  components,  $u_k = \{u_k^1, \dots, u_k^m\}$ , with  $u_k^i$  corresponding to the  $i$ th subsystem. Then to compute a cost-to-go approximation at a given state  $x_k$ , one may try a one-subsystem-at-a-time approach: first optimize over the control sequence  $\{u_k^1, u_{k+1}^1, \dots, u_{N-1}^1\}$  of the first subsystem, while keeping the controls of the remaining subsystems at some nominal values, then minimize over the controls of the second subsystem, while keeping the controls of the first subsystem at the "optimal" values just computed and the controls of subsystems  $3, \dots, m$  to the nominal values, and continue in this manner. There are several possible variations, for example to make the order in which the subsystems are considered subject to optimization as well. Let us illustrate this approach by means of an example.

### Example 6.3.7 (Vehicle Routing)

Consider  $m$  vehicles that move along the arcs of a given graph. Each node of the graph has a given "value" and the first vehicle that will pass through the node will collect its value, while vehicles that pass subsequently through the node do not collect any value. This may serve as a model of a situation where there are various valuable tasks to be performed at the nodes of a transportation network, and each task can be performed at most once and by a single vehicle. We assume that each vehicle starts at a given node and after at most a given number of arc moves, it must return to some other given node. The problem is to find a route for each vehicle satisfying these constraints, so that the total value collected by the vehicles is maximized.

This is a difficult combinatorial problem that in principle can be approached by DP. In particular, we can view as the state the set of current positions of the vehicles together with the list of nodes that have been traversed by some vehicle in the past, and have thus "lost" their value. Unfortunately, the number of these states is enormous (it increases exponentially with the number of nodes and the number of vehicles). The version of the problem involving a single vehicle, while still difficult in principle, can often be solved in reasonable time either exactly by DP or fairly accurately using a heuristic. Thus a one step-lookahead policy suggests itself, with the value-to-go approximation obtained by solving single vehicle problems.

In particular, in a one step-lookahead scheme, at a given time  $k$  and from a given state we consider all possible  $k$ th moves by the vehicles, and at

the resulting states we approximate the optimal value-to-go with the value corresponding to a suboptimal set of paths. These paths are obtained as follows: we fix an order of the vehicles and we calculate a path for the first vehicle, assuming the other vehicles do not move. (This is done either optimally by DP, or nearly optimally using some heuristic.) Then we calculate a path for the second vehicle in the order, taking into account the value collected by the first vehicle, and we similarly continue: for each vehicle, we calculate in the given order a path, taking into account the value collected by the preceding vehicles. We end up with a set of paths that have a certain total value associated with them, and which correspond to the particular order for considering the vehicles. We can also calculate other sets of paths and their corresponding total values, for other orders (possibly all orders) for considering the vehicles. We then use as the value-to-go approximation at the given state the maximal value over all the sets of paths computed from that state.

Another context where enforced decomposition may be an attractive possibility, is when the subsystems are coupled only through the disturbance. In particular, consider  $m$  subsystems of the form

$$x_{k+1}^i = f^i(x_k^i, u_k^i, w_k^i), \quad i = 1, \dots, m.$$

Here the  $i$ th subsystem has its own state  $x_k^i$ , control  $u_k^i$ , and cost per stage  $g^i(x_k^i, u_k^i, w_k^i)$ , but the probability distribution of  $w_k^i$  depends on the full state

$$x_k = (x_k^1, \dots, x_k^m).$$

A natural form of suboptimal control is to solve at each stage  $k$  and for each  $i$ , the  $i$ th subsystem optimization problem where the probability distribution of each of the future disturbances  $w_{k+1}^i, \dots, w_{N-1}^i$  is fixed at some distribution that depends only on the corresponding "local" states  $x_{k+1}^i, \dots, x_{N-1}^i$ . This distribution may be derived on the basis of some nominal values  $\bar{x}_{k+1}^j, \dots, \bar{x}_{N-1}^j$ ,  $j \neq i$ , of the future states of the other subsystems, and these nominal values may in turn depend on the full current state  $x_k$ . The first control  $u_k^i$  in the optimal policy thus obtained is applied at the  $i$ th subsystem in stage  $k$ , and the remaining portion of this policy is discarded.

Let us also discuss in some detail an example of subsystem decomposition where the coupling comes through the control constraint.

#### Example 6.3.8 (Flexible Manufacturing)

Flexible manufacturing systems (FMS) provide a popular approach for increasing productivity in manufacturing small batches of related parts. There are several workstations in an FMS, and each is capable of carrying out a variety of operations. This allows the simultaneous manufacturing of more than one part type, reduces idle time, and allows production to continue even when a workstation is out of service because of failure or maintenance.

Consider a work center in which  $n$  part types are produced. Denote

$u_k^i$ : the amount of part  $i$  produced in period  $k$ .

$d_k^i$ : a known demand for part  $i$  in period  $k$ .

$x_k^i$ : the cumulative difference of amount of part  $i$  produced and demanded up to period  $k$ .

Let us denote also by  $x_k$ ,  $u_k$ ,  $d_k$  the  $n$ -dimensional vectors with coordinates  $x_k^i$ ,  $u_k^i$ ,  $d_k^i$ , respectively. We then have

$$x_{k+1} = x_k + u_k - d_k. \quad (6.25)$$

The work center consists of a number of workstations that fail and get repaired in random fashion, thereby affecting the productive capacity of the system (i.e., the constraints on  $u_k$ ). Roughly, our problem is to schedule part production so that  $x_k$  is kept around zero to the extent possible.

The productive capacity of the system depends on a random variable  $\alpha_k$  that reflects the status of the workstations. In particular, we assume that the production vector  $u_k$  must belong to a constraint set  $U(\alpha_k)$ . We model the evolution of  $\alpha_k$  by a Markov chain with known transition probabilities  $P(\alpha_{k+1} | \alpha_k)$ . In practice, these probabilities must be estimated from individual station failure and repair rates, but we will not go into the matter further. Note also that in practice these probabilities may depend on  $u_k$ . This dependence is ignored for the purpose of development of a cost-to-go approximation. It may be taken into account when the actual suboptimal control is computed.

We select as system state the pair  $(x_k, \alpha_k)$ , where  $x_k$  evolves according to Eq. (6.25) and  $\alpha_k$  evolves according to the Markov chain described earlier. The problem is to find for every state  $(x_k, \alpha_k)$  a production vector  $u_k \in U(\alpha_k)$  such that a cost function of the form

$$J_\pi(x_0) = E \left\{ \sum_{k=0}^{N-1} \sum_{i=1}^n g^i(x_k^i) \right\}$$

is minimized. The function  $g^i$  expresses the desire to keep the current backlog or surplus of part  $i$  near zero. Two examples are  $g^i(x^i) = \beta_i |x^i|$  and  $g^i(x^i) = \beta_i (x^i)^2$ , where  $\beta_i > 0$ .

The DP algorithm for this problem is

$$\begin{aligned} J_k(x_k, \alpha_k) &= \sum_{i=1}^n g^i(x_k^i) \\ &+ \min_{u_k \in U(\alpha_k)} E \left\{ J_{k+1}(x_k + u_k - d_k, \alpha_{k+1}) | \alpha_k \right\}. \end{aligned} \quad (6.26)$$

If there is only one part type ( $n = 1$ ), the optimal policy can be fairly easily determined from this algorithm (see Exercise 6.7). However, in general, the algorithm requires a prohibitive amount of calculation for an FMS of realistic size (say for  $n > 10$  part types). We thus consider a one-step lookahead policy

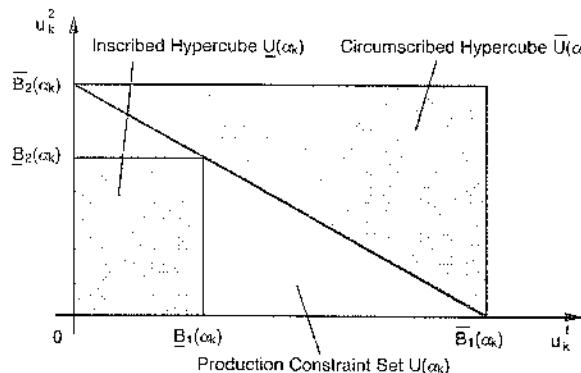


Figure 6.3.1 Inner and outer approximations of the production capacity constraint set by hypercubes in the flexible manufacturing example.

with the cost-to-go  $J_{k+1}$  replaced by an approximation  $\tilde{J}_{k+1}$  that exploits the nearly separable structure of our problem.

In particular, we note that the problem can to a large extent be decomposed with respect to individual part types. Indeed, the system equation (6.25) and the cost per stage are decoupled, and the only coupling between parts comes from the constraint  $u_k \in U(\alpha_k)$ . Suppose we approximate  $U(\alpha_k)$  by inner and outer approximating hypercubes  $\underline{U}(\alpha_k)$  and  $\overline{U}(\alpha_k)$  of the form

$$\underline{U}(\alpha_k) = \{u_k^i \mid 0 \leq u_k^i \leq \underline{B}_i(\alpha_k)\},$$

$$\overline{U}(\alpha_k) = \{u_k^i \mid 0 \leq u_k^i \leq \overline{B}_i(\alpha_k)\},$$

$$\underline{U}(\alpha_k) \subset U(\alpha_k) \subset \overline{U}(\alpha_k),$$

as shown in Fig. 6.3.1. If  $U(\alpha_k)$  is replaced for each  $\alpha_k$  by either  $\overline{U}(\alpha_k)$  or  $\underline{U}(\alpha_k)$ , then the problem is decomposed completely with respect to part types. For every part  $i$  the DP algorithm for the outer approximation is given by

$$\begin{aligned} \overline{J}_k^i(x_k^i, \alpha_k) &= g^i(x_k^i) \\ &+ \min_{0 \leq u_k^i \leq \overline{B}_i(\alpha_k)} E \left\{ \overline{J}_k^i(x_k^i + u_k^i - d_k^i, \alpha_{k+1}) \mid \alpha_k \right\}, \end{aligned} \quad (6.27)$$

and for the inner approximation it is given by

$$\begin{aligned} \underline{J}_k^i(x_k^i, \alpha_k) &= g^i(x_k^i) \\ &+ \min_{0 \leq u_k^i \leq \underline{B}_i(\alpha_k)} E \left\{ \underline{J}_k^i(x_k^i + u_k^i - d_k^i, \alpha_{k+1}) \mid \alpha_k \right\}. \end{aligned} \quad (6.28)$$

Furthermore, since  $\underline{U}(\alpha_k) \subset U(\alpha_k) \subset \overline{U}(\alpha_k)$ , the cost-to-go functions  $\overline{J}_k^i$  and  $\underline{J}_k^i$  provide lower and upper bounds to the true cost-to-go function  $J_k$ ,

$$\sum_{i=1}^n \overline{J}_k^i(x_k^i, \alpha_k) \leq J_k(x_k, \alpha_k) \leq \sum_{i=1}^n \underline{J}_k^i(x_k^i, \alpha_k),$$

and can be used to construct approximations to  $J_k$  that are suitable for a one-step lookahead policy. A simple possibility is to adopt the averaging approximation

$$\tilde{J}_k(x_k, \alpha_k) = \frac{1}{2} \sum_{i=1}^n (\overline{J}_k^i(x_k^i, \alpha_k) + \underline{J}_k^i(x_k^i, \alpha_k))$$

and use at state  $(x_k, \alpha_k)$  the control  $\tilde{u}_k$  that minimizes [cf. Eq. (6.26)]

$$E \left\{ \sum_{i=1}^n (\overline{J}_{k+1}^i(x_k^i + u_k^i - d_k^i, \alpha_{k+1}) + \underline{J}_{k+1}^i(x_k^i + u_k^i - d_k^i, \alpha_{k+1})) \mid \alpha_k \right\} \quad (6.29)$$

over all  $u_k \in U(\alpha_k)$ . Multiple upper bound approximations, based on multiple choices of  $\underline{B}_i(\alpha_k)$ , can also be used.

To implement this scheme, it is necessary to compute and store the approximate cost-to-go functions  $\overline{J}_k^i$  and  $\underline{J}_k^i$  in tables, so that they can be used in the real-time computation of the suboptimal control via the minimization of expression (6.29). The corresponding calculations [cf. the DP algorithms (6.27) and (6.28)] are nontrivial, but they can be carried out off-line, and in any case they are much less than what would be required to compute the optimal controller. The feasibility and the benefits of the overall approach have been demonstrated by simulation in the thesis by Kimemia [Kim82], on which this example is based. See also Kimemia, Gershwin, and Bertsekas [KGB82], and Tsitsiklis [Tsi84a].

For some other examples of decomposition approaches, see Wu and Bertsekas [WuB99], which deals with admission control in cellular communication networks, and Meuleau et al. [MHK98], which deals with problems of resource allocation.

#### 6.3.4 Aggregation

An alternative method for constructing a simpler and more tractable problem is based on reducing the number of states by “combining” many of them together into *aggregate states*. This results in an *aggregate problem*, with fewer states, which may be solvable by exact DP methods. The optimal cost-to-go functions of the aggregate problem is then used to construct a one-step-lookahead cost approximation for the original problem. The precise form of the aggregate problem may depend on intuition and/or heuristic reasoning, based on our understanding of the original problem.

In this subsection, we will discuss various aggregation methods, starting with the case of a finite-state problem. We will focus on defining the transition probabilities and costs of the aggregate problem, and to simplify notation, we suppress the time indexing in what follows. We generically denote:

$I, \bar{I}$ : The set of states of the original system at the current and the next stage, respectively.

$p_{ij}(u)$ : The transition probability of the original system from state  $i \in I$  to state  $j \in \bar{I}$  under control  $u$ .

$g(i, u, j)$ : The transition cost of the original system from state  $i \in I$  to state  $j \in \bar{I}$  under control  $u$ .

$S, \bar{S}$ : The set of states of the aggregate system at the current and the next stage, respectively.

$r_{st}(u)$ : The transition probability of the aggregate system from state  $s \in S$  to state  $t \in \bar{S}$  under control  $u$ .

$h(s, u)$ : The expected transition cost of the aggregate system from state  $s \in S$  under control  $u$ .

For simplicity, we assume that the control constraint set  $U(i)$  is the same for all states  $i \in I$ . This common control constraint set, denoted by  $U$ , is chosen as the control constraint set at all states  $s \in S$  of the aggregate problem.

There are several types of aggregation methods, which bring to bear intuition about the problem's structure in different ways. All these methods are based on two (somewhat arbitrary) choices of probabilities, which relate the original system states with the aggregate states:

- (1) For each aggregate state  $s \in S$  and original system state  $i \in I$ , we specify the *disaggregation probability*  $q_{si}$  (we have  $\sum_{i \in I} q_{si} = 1$  for each  $s \in S$ ). Roughly,  $q_{si}$  may be interpreted as the "degree to which  $s$  is represented by  $i$ ."
- (2) For each original system state  $j \in \bar{I}$  and aggregate state  $t \in \bar{S}$ , we specify the *aggregation probability*  $w_{jt}$  (we have  $\sum_{t \in \bar{S}} w_{jt} = 1$  for each  $j \in \bar{I}$ ). Roughly,  $w_{jt}$  may be interpreted as the "degree of membership of  $j$  in the aggregate state  $t$ ."

Note that in general, the disaggregation and aggregation probabilities may change at each stage (since the state space may change at each stage). On the other hand, for a stationary problem, where state and control spaces, system equation, and cost per stage that are the same for all stages, the disaggregation and aggregation probabilities will ordinarily also be the same for all stages.

As an illustration consider the following example of aggregation.

### Example 6.3.9 (Hard Aggregation)

We are given a partition of the original system state spaces  $I$  and  $\bar{I}$  into subsets of states (each state belongs to one and only one subset). We view each subset as an aggregate state. This corresponds to aggregation probabilities

$$w_{jt} = 1 \quad \text{if state } j \in \bar{I} \text{ belongs to aggregate state/subset } t \in \bar{S},$$

and (assuming all states that belong to aggregate state/subset  $s$  are "equally representative") disaggregation probabilities

$$q_{si} = 1/n_s \quad \text{if state } i \in I \text{ belongs to aggregate state/subset } s \in S,$$

where  $n_s$  is the number of states of  $s$ .

Given the disaggregation and aggregation probabilities,  $q_{si}$  and  $w_{jt}$ , and the original transition probabilities  $p_{ij}(u)$ , we envisage an aggregate system where state transitions occur as follows:

- (i) From aggregate state  $s$ , generate state  $i$  according to  $q_{si}$ .
- (ii) Generate a transition from  $i$  to  $j$  according to  $p_{ij}(u)$ , with cost  $g(i, u, j)$ .
- (iii) From state  $j$ , generate aggregate state  $t$  according to  $w_{jt}$ .

Then, the transition probability from aggregate state  $s$  to aggregate state  $t$  under  $u$ , and the corresponding expected transition cost, are given by

$$r_{st}(u) = \sum_{i \in I} \sum_{j \in \bar{I}} q_{si} p_{ij}(u) w_{jt},$$

$$h(s, u) = \sum_{i \in I} \sum_{j \in \bar{I}} q_{si} p_{ij}(u) g(i, u, j).$$

These transition probabilities and costs define the aggregate problem. After solving for the optimal costs-to-go  $\hat{J}(t)$ ,  $t \in \bar{S}$ , of the aggregate problem, the costs of the original problem are approximated by

$$\hat{J}(j) = \sum_{t \in \bar{S}} w_{jt} \hat{J}(t), \quad j \in \bar{I}. \quad (6.30)$$

As an illustration, for the preceding hard aggregation Example 6.3.9, the aggregate system transition process works as follows: Starting from an aggregate state/subset  $s$ , we generate with equal probability a state  $i$  in  $s$ , then a next state  $j \in \bar{I}$  according to the transition probabilities  $p_{ij}(u)$ , and then we declare as next aggregate state the subset  $t$  to which  $j$  belongs. The corresponding transition probability and expected transition cost are

$$r_{st}(u) = \frac{1}{n_s} \sum_{i \in s} \sum_{j \in t} p_{ij}(u),$$

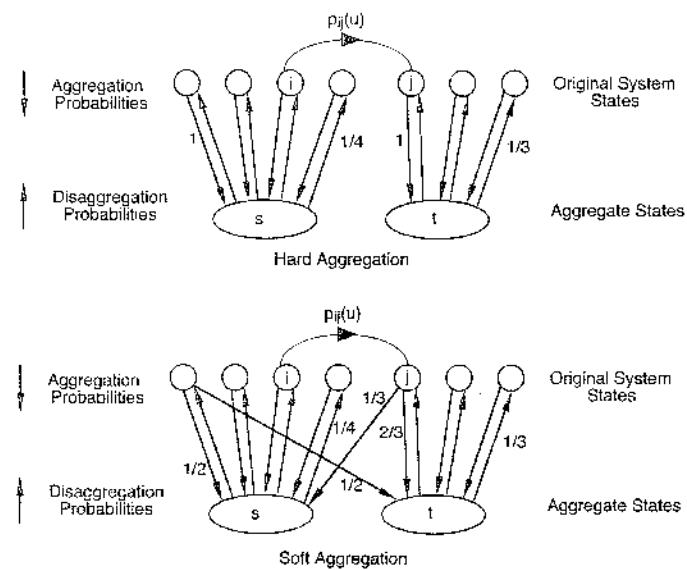


Figure 6.3.1 Disaggregation and aggregation probabilities in the hard and soft aggregation Examples 6.3.9 and 6.3.10. The difference is that in soft aggregation, some of the aggregation probabilities are strictly between 0 and 1, i.e., a state of the original system may be associated with multiple aggregate states.

$$h(s, u) = \frac{1}{n_s} \sum_{i \in s} \sum_{j \in \bar{I}} p_{ij}(u) g(i, u, j).$$

After computing the optimal costs  $\hat{J}(t)$  of the aggregate problem, we use Eq. (6.30) to obtain the approximate cost function  $\tilde{J}(j)$ , which for this hard aggregation example is piecewise constant, with all states  $j \in \bar{S}$  that belong to the same aggregate state/subset  $t$  having the same value of  $\tilde{J}(j)$ .

#### Example 6.3.10 (Soft Aggregation)

In hard aggregation, the aggregate states/subsets are disjoint, and each original system state is associated with a single aggregate state. A generalization is to allow the aggregate states/subsets to overlap, with the aggregation probabilities  $w_{jt}$  quantifying the “degree of membership” of  $j$  in the aggregate state  $t$ . Thus, an original system state  $j$  may be a member of multiple aggregate states/subsets  $t$ , and if this is so, the aggregation probabilities  $w_{jt}$  will be positive but less than 1 for all  $t$  that contain  $j$  (we should still have  $\sum_{t \in \bar{S}} w_{jt} = 1$ ; see Fig. 6.3.1).

For example, assume that we are dealing with a queue that has space for 100 customers, and that the state is the number of spaces occupied at a given time. Suppose that we introduce four aggregate states: “nearly empty”

(0-10 customers), “lightly loaded” (11-50 customers), “heavily loaded” (51-90 customers), and “nearly full” (91-100 customers). Then it makes sense to use soft aggregation, so that a state with close to 50 customers is classified neither as “lightly loaded” nor as “heavily loaded,” but is viewed instead as associated with both of these aggregate states, to some degree.

It can be seen from Eq. (6.30), that in soft aggregation, the approximate cost-to-go function  $\tilde{J}$  is not piecewise constant, as in the case of hard aggregation, and varies “smoothly” along the boundaries separating aggregate states. This is because original system states that belong to multiple aggregate states/subsets have approximate cost-to-go that is a convex combination of the costs-to-go of these aggregate states.

The choices of aggregate states, and aggregation and disaggregation probabilities are often guided by insight into the problem’s structure. The following is an example.

#### Example 6.3.11 (Admission Control in a Service Facility)

Consider a facility that serves  $m$  types of customers. At each time period, a single customer of each type arrives, and requires a level of service per unit time that is random and has a given distribution (which depends on the customer type). The facility must decide which of the arriving customers to admit at each time period. The total service capacity of the facility is given, and must at all times be no less than the sum of service levels of the customers currently within the facility. An admitted customer of a given type has a given probability of leaving the facility at each time period, independently of his required level of service and of how long he has already been in the facility. An admitted customer also pays to the facility a given amount per period, which is proportional to his required level of service (with the constant of proportionality depending on the customer type). The objective here is to maximize the expected revenue of the facility over a given horizon. Thus the tradeoff is to provide selective preference to high-paying customer types, or alternatively expressed, to avoid filling up the facility with low-paying long-staying customers, thereby potentially blocking out some high-paying customers.

In a problem of this type, the system state, just prior to making an admission decision, is the entire list of customers of each type within the facility as well as their service levels (together with the list of service levels of the customers that have just arrived). There is clearly an extremely large number of states. Intuition suggests here that it is appropriate to aggregate all customers of a given type, and represent them with their total service level. Thus the aggregate state in this approach is the list of total required service level of each type within the facility (together with the list of service levels of the customers that have just arrived - an uncontrollable state component, cf. Section 1.4). Clearly it is much easier to deal with such a space of aggregate states in a DP context.

The choice of aggregation probabilities here is clear, since any original system state maps naturally into a unique aggregate state. The rationale for specifying the disaggregation probabilities, while somewhat arbitrary, is

guided by intuition into the problem structure. Given an aggregate state (a total level of service for each customer type within the facility), we must generate through the disaggregation probabilities, a “representative” list of customers of each type. It is not difficult to devise reasonable heuristic methods for doing so. The disaggregation and aggregation probabilities, together with the transition probabilities of the original system, specify the transition probabilities and the expected cost per stage of the aggregate problem.

Some simplification techniques, aimed at reducing the complexity of the DP computation, can be interpreted in terms of aggregation. The following is an example.

#### Example 6.3.12 (Using a Coarse Grid)

A technique often used to reduce the computational requirements of DP is to select a small collection  $S$  of states from  $I$  and a small collection  $\bar{S}$  of states from  $\bar{I}$ , and define an aggregate problem whose states are those in  $S$  and  $\bar{S}$ . The aggregate problem is then solved by DP and its optimal costs are used to define approximate costs for all states in  $I$  and  $\bar{I}$ . This process, is known as *coarse grid approximation*, and is motivated by the case where the original state spaces  $I$  and  $\bar{I}$  are dense grids of points obtained by discretization of continuous state spaces, while the collections  $S$  and  $\bar{S}$  represent coarser subgrids.

The aggregate problem may be formalized by specifying the disaggregation probabilities to associate the states of  $S$  with themselves (since  $S \subset I$ ):

$$q_{ss} = 1, \quad q_{si} = 0 \text{ if } i \neq s, \quad s \in S.$$

The aggregation probabilities are used to represent each state in  $\bar{I}$  as a probabilistic mixture of states in  $\bar{S}$ , respectively, possibly using some geometrical attribute of the state space.

The aggregation methodology also applies to problems with an infinite number of states. The only difference is that for each aggregate state, the disaggregation probabilities are replaced by a *disaggregation distribution* over the original system’s state space. Among other possibilities, this type of aggregation provides methods for discretizing continuous state space problems, as illustrated by the following example.

#### Example 6.3.13 (Discretization of Continuous State Spaces)

Assume for simplicity a stationary problem, where the state space of the original problem is a bounded region of a Euclidean space. The idea here is to discretize this state space using some finite grid  $\{x^1, \dots, x^M\}$ , and then to express each nongrid state by a linear interpolation of nearby grid states. By this we mean that the grid states  $x^1, \dots, x^M$  are suitably selected within the state space, and each nongrid state  $x$  is expressed as

$$x = \sum_{m=1}^M w^m(x) x^m,$$

for some nonnegative weights  $w^m(x)$ , which add to 1 and are chosen on the basis of some geometric considerations. We view the weights  $w^m(x)$  as aggregation probabilities, and we specify the disaggregation probabilities to associate the grid states with themselves, i.e.,

$$q_{xm} = 1, \quad \text{for all } m,$$

similar to the coarse grid approach of Example 6.3.12.

The aggregation and disaggregation probabilities just given specify the aggregate problem, which has a finite state space, the set  $\{x^1, \dots, x^M\}$ , and can be solved by DP to obtain the corresponding optimal costs-to-go  $\hat{J}_k(x^m)$ ,  $m = 1, \dots, M$ , for each stage  $k$ . Then the cost-to-go of each nongrid state  $x$  at stage  $k$  is approximated by

$$\tilde{J}_k(x) = \sum_{m=1}^M w^m(x) \hat{J}_k(x^m).$$

We finally note that one may address the solution of the aggregate problem itself by some suboptimal method, thereby introducing an additional layer of approximation in the solution of the original problem.

#### 6.3.5 Parametric Cost-to-Go Approximation

The idea here is to select from within a parametric class of functions, some cost-to-go approximations  $\tilde{J}_k$ , which will be used in a limited lookahead scheme in place of the optimal cost-to-go functions  $J_k$ . Such parametric classes of functions are called *approximation architectures*, and are generically denoted by  $\tilde{J}(x, r)$ , where  $x$  is the current state and  $r = (r_1, \dots, r_m)$  is a vector of “tunable” scalar parameters, also called *weights* (to simplify notation, we suppress the time indexing in what follows). By adjusting the weights, one can change the “shape” of the approximation  $\tilde{J}$  so that it is reasonably close to the true optimal cost-to-go function.

There is an extensive methodology for the selection of the weights. The simplest and often tried approach is to do some form of semi-exhaustive or semi-random search in the space of weight vectors and adopt the weights that result in best performance of the associated one-step lookahead controller. Other more systematic approaches are based on various forms of cost-to-go evaluation and least squares fit. We will discuss such approaches briefly here and more extensively in Vol. II in the context of the methodology of neuro-dynamic programming; see also the books by Bertsekas and Tsitsiklis [BeT96], and Sutton and Barto [SuB98].

There is also a large variety of approximation architectures, involving for example polynomials, neural networks, wavelets, various types of basis functions, etc. We provide a brief discussion of architectures based on extraction of features, and we refer to the specialized literature (e.g., Bertsekas and Tsitsiklis [BeT96], Bishop [Bis95], Haykin [Hay99], Sutton and Barto [SuB98]) for detailed discussions.

### Approximation Architectures Based on Feature Extraction

Clearly, for the success of the cost function approximation approach, it is very important to select a class of functions  $\hat{J}(x, r)$  that is suitable for the problem at hand. One particularly interesting type of cost approximation is provided by *feature extraction*, a process that maps the state  $x$  into some other vector  $y(x)$ , called the *feature vector* associated with state  $x$ . The vector  $y(x)$  consists of scalar components  $y_1(x), \dots, y_m(x)$ , called *features*. These features are usually handcrafted, based on whatever human intelligence, insight, or experience is available, and are meant to capture the most important aspects of the current state  $x$ . A feature-based cost approximation has the form

$$\tilde{J}(x, r) = \hat{J}(y(x), r),$$

where  $r$  is a parameter vector. Thus, the cost approximation depends on the state  $x$  through its feature vector  $y(x)$  (see Fig. 6.3.2).

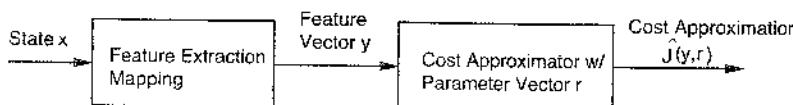


Figure 6.3.2 Using a feature extraction mapping to generate an input to a cost approximator.

The idea is that the cost-to-go function  $J$  to be approximated may be a highly complicated nonlinear map and it is sensible to try to break its complexity into smaller, less complex pieces. Ideally, the features will encode much of the nonlinearity that is inherent in  $J$ , and the approximation may be quite accurate without requiring a complicated function  $\hat{J}$ . For example, with a well-chosen feature vector  $y(x)$ , a good approximation to the cost-to-go is often provided by *linearly* weighting the features, i.e.,

$$\tilde{J}(x, r) = \hat{J}(y(x), r) = \sum_{i=1}^m r_i y_i(x),$$

where  $r_1, \dots, r_m$  is a set of tunable scalar weights.

Note that the use of a feature vector implicitly involves the grouping of states into the subsets that share the same feature vector, i.e., the subsets

$$S_v = \{x \mid y(x) = v\},$$

where  $v$  is possible value of  $y(x)$ . These subsets form a partition of the state space, and the approximate cost-to-go function  $\tilde{J}(y(x), r)$  is piecewise

constant with respect to this partition; that is, it assigns the same cost-to-go value  $\tilde{J}(v, r)$  to all states in the set  $S_v$ . This suggests that a feature extraction mapping is well-chosen if states that have the same feature vector have roughly similar optimal cost-to-go.

A feature-based architecture is also related to the aggregation methodology of Section 6.3.4. In particular, suppose that we introduce  $m$  aggregate states,  $1, \dots, m$ , and associated aggregation and disaggregation probabilities. Let  $r_i$  be the optimal cost-to-go associated with aggregate state  $i$ . Then, the aggregation methodology yields the linear parametric approximation

$$\tilde{J}(x, r) = \sum_{i=1}^m r_i y_i(x),$$

where  $y_i(x)$  is the aggregation probability associated with state  $x$  and aggregate state  $i$ . Thus, within this context, the aggregation probability  $y_i(x)$  may be viewed as a feature, which, roughly speaking, specifies the “degree of membership of  $x$  to aggregate state  $i$ .”

We now illustrate the preceding concepts with a detailed discussion of computer chess, where feature-based approximation architectures play an important role.

### Computer Chess

Chess-playing computer programs are one of the more visible successes of artificial intelligence. Their underlying methodology relies provides an interesting case study in the use of approximate DP. It involves the idea of limited lookahead, but also illustrates some DP ideas that we have not had much opportunity to look at in detail. These are the idea of *forward depth-first search*, an important memory-saving technique that was discussed in Section 2.3 in the context of label correcting methods, and the idea of *alpha-beta pruning*, which is an effective method for reducing the amount of computation needed to find optimal strategies in competitive games.

The fundamental paper on which all computer chess programs are based was written by one of the most illustrious modern-day applied mathematicians, C. Shannon [Sha50]. It was argued by Shannon that whether the starting chess position is a win, loss, or draw is a question that can be answered in principle, but the answer will probably never be known. He estimated that, based on the chess rule requiring at least one pawn advance or capture within every 50 moves (otherwise a draw is declared), there are on the order of  $10^{120}$  different possible sequences of moves in a chess game. He concluded that to examine these and select the best initial move for White would require  $10^{90}$  years of a “fast” computer’s time (fast here relates to the standards of the ‘50s, but the number  $10^{90}$  is overwhelming even by today’s standards). As an alternative, Shannon proposed a *limited lookahead* of a few moves and *evaluating the end positions by means of a*

*scoring function.* The scoring function may involve, for example, the calculation of a numerical value for each of a set of major features of a position (such as material balance, mobility, pawn structure, and other positional factors), together with a method to combine these numerical values into a single score. Thus, we may view a scoring function as a feature-based architecture for evaluating a chess position/state.

Consider first a *one-move lookahead strategy* for selecting the first move in a given position  $P$ . Let  $M_1, \dots, M_r$  be all the legal moves that can be made in position  $P$  by the side to move. Denote the resulting positions by  $M_1P, \dots, M_rP$ , and let  $S(M_1P), \dots, S(M_rP)$  be the corresponding scores (the convention here is that White is favored in positions with high score, while Black is favored in positions with low score). Then the move selected by White (Black) in position  $P$  is the move with maximum (minimum) score. This is known as the *backed-up score* of  $P$  and is given by

$$BS(P) = \begin{cases} \max\{S(M_1P), \dots, S(M_rP)\} & \text{if White is to move in } P, \\ \min\{S(M_1P), \dots, S(M_rP)\} & \text{if Black is to move in } P. \end{cases}$$

This process is illustrated in Fig. 6.3.3.

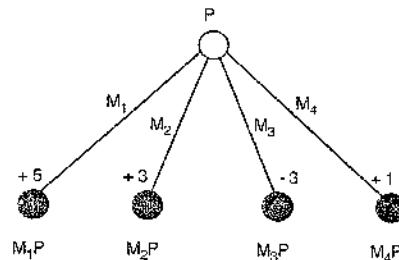


Figure 6.3.3 A one-move lookahead tree. If White moves at position  $P$ , the best move is  $M_1$  and the backed-up score is  $+5$ . If Black moves in position  $P$ , the best move is  $M_3$ , and the backed-up score of  $P$  is  $-3$ .

Consider next a *two-move lookahead strategy* in a given position  $P$ . Assume for concreteness that White moves, and let the legal moves be  $M_1, \dots, M_r$  and the corresponding positions be  $M_1P, \dots, M_rP$ . Then in each of the positions  $M_iP$ ,  $i = 1, \dots, r$ , apply the one-move lookahead strategy with Black to move. This gives a best move and a backed-up score  $BS(M_iP)$  for Black in each of the positions  $M_iP$ ,  $i = 1, \dots, r$ . Finally, based on the backed-up scores  $BS(M_1P), \dots, BS(M_rP)$ , apply a one-move lookahead strategy for White, thereby obtaining the best move at position  $P$  and a backed-up score for position  $P$  of

$$BS(P) = \max\{BS(M_1P), \dots, BS(M_rP)\}.$$

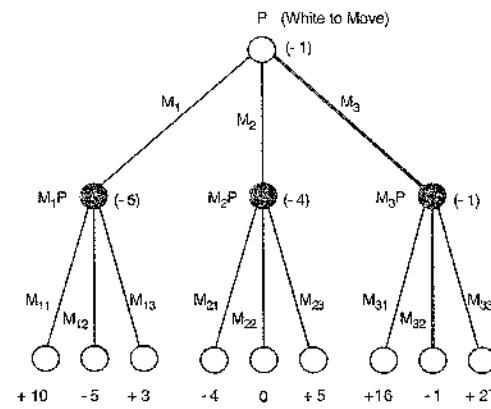


Figure 6.3.4 A two-move lookahead tree with White to move. The backed-up scores are shown in parentheses. The best initial move is  $M_3$  and the principal continuation is  $(M_3, M_{32})$ .

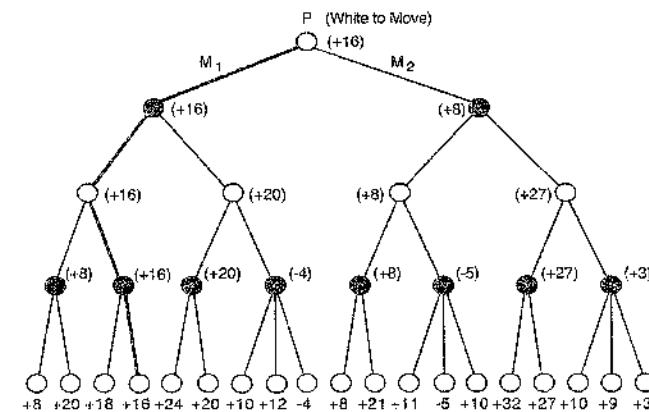


Figure 6.3.5 A four-move lookahead tree with White to move. The backed-up scores are shown in parentheses. The best initial move is  $M_1$ . The principal continuation is heavily shaded.

The sequence of best moves is known as the *principal continuation*. The process is illustrated in Fig. 6.3.4. It is clear that Shannon's method as just described can be generalized for an arbitrary number of lookahead moves (see Fig. 6.3.5).

Generally, to evaluate the best move at a given position and the corresponding backed-up score using lookahead of  $n$  moves, one can use the following DP-like procedure:

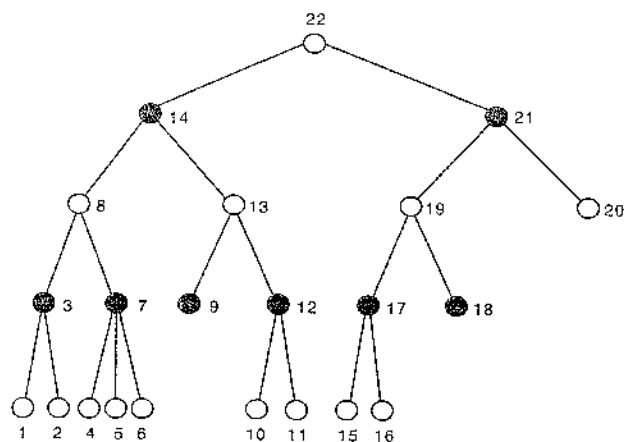


Figure 6.3.6 Traversing a tree in depth-first fashion. The numbers indicate the order in which the scores of the terminal positions and the backed-up scores of the intermediate positions are evaluated.

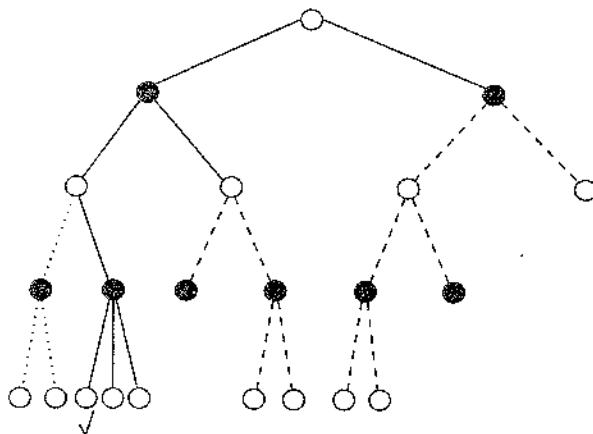


Figure 6.3.7 Storage requirements of the depth-first version of the minimax algorithm for the tree of Fig. 6.3.6. At the time that the terminal position marked by a checkmark is scored, only the solid-line moves are stored in memory. The dotted-line moves have been generated and purged from memory. The broken-line moves have not been generated as yet. The memory required grows linearly with the depth of the lookahead.

1. Evaluate the scores of all possible positions that are  $n$  moves ahead from the given position  $P$ .
2. Using the scores of the terminal positions just evaluated, calculate the backed-up scores of all possible positions that are  $n - 1$  moves ahead from  $P$ .
3. For  $k = 1, \dots, n - 1$ , using the backed-up scores of all possible positions that are  $n - k$  moves ahead from  $P$ , calculate the backed-up scores of all positions that are  $n - k - 1$  moves ahead from  $P$ .

The above procedure requires a lot of memory storage even for a modest number of lookahead moves. Shannon pointed out that with an alternative but mathematically equivalent calculation, the amount of memory required grows only *linearly* with the depth of lookahead, thereby allowing chess programs to operate in limited-memory microprocessor systems. This is accomplished by searching the tree of moves in *depth-first fashion*, and by generating new moves only when needed, as illustrated in Figs. 6.3.6 and 6.3.7. It is only necessary to store the *one* move sequence under current examination together with one list of legal moves at each level of the search tree. The precise algorithm is described by the following routine, which calls itself recursively.

#### Minimax Algorithm

To determine the backed-up score  $BS(n)$  of position  $n$ , do the following:

1. If  $n$  is a terminal position return its score. Otherwise:
2. Generate the list of legal moves at position  $n$  and let the corresponding positions be  $n_1, \dots, n_r$ . Set the tentative backed-up score  $TBS(n)$  of position  $n$  to  $\infty$  if it is White's turn to move at  $n$  and to  $-\infty$  if it is Black's turn to move at  $n$ .
3. For  $i = 1, \dots, r$ , do:
  - a. Determine the backed-up score  $BS(n_i)$  of position  $n_i$ .
  - b. If it is White's move at position  $n$ , set

$$TBS(n) := \max\{TBS(n), BS(n_i)\};$$

if it is Black's move at position  $n$ , set

$$TBS(n) := \min\{TBS(n), BS(n_i)\}.$$

4. Return  $BS(n) = TBS(n)$ .

The idea of solving one-step lookahead problems with a terminal cost (or backed-up score) that summarizes future costs is of course central in the DP algorithm. Indeed, it can be seen that the minimax algorithm just described is nothing but the DP algorithm for minimax problems (see Section 1.6). Here, positions and moves can be identified with states and controls, respectively, there are only terminal costs (the scores of the terminal positions), and the backed-up score of a position is nothing but the optimal cost-to-go at the corresponding state.

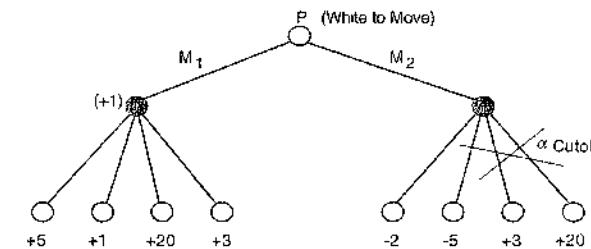
The minimax algorithm is also known as the *type A strategy*. Shannon argued that with this strategy, one could not expect a computer to seriously challenge human players of even moderate strength. In a typical chess position there are around 30 to 35 legal moves. It follows that for an  $n$ -move lookahead there will be around  $30^n$  to  $35^n$  terminal positions to be scored. Thus the number of terminal positions grows exponentially with the size of lookahead, practically limiting  $n$  to being of the order of 10 with present computers. Unfortunately, in some chess positions it is essential to look a substantially larger number of moves ahead. In particular, in dynamic positions involving many captures and countercaptures, the necessary size of lookahead can be very large.

These considerations led Shannon to consider another strategy, called *type B*, whereby the depth of the search tree is variable. He suggested that at each position the computer give all legal moves a preliminary examination and discard those that are "obviously bad." A scoring function together with some heuristic strategy can be used for this purpose. Similarly, he suggested that some positions that are dynamic, such as those involving many captures or checkmate threats, be explored further beyond the nominal depth of the search.

Chess-playing computer programs typically use a combination of Shannon's type *A* and *B* strategies. These programs use scoring functions, the forms of which have evolved by trial-and-error, but they also use sophisticated heuristics to evaluate dynamic terminal positions in detail. In particular, an effective algorithm, known as *swapoff*, is used to quickly analyze long sequences of captures and countercaptures, thereby making it possible to score realistically complex, dynamic positions (see Levy [Lev84] for a description). One may view such heuristics as either defining a sophisticated scoring function, or as implementing a type *B* strategy.

The efficiency of the minimax algorithm can be substantially improved by using the *alpha-beta pruning procedure* (denoted  $\alpha\beta$  for short), which can be used to forego some calculations involving positions that cannot affect the selection of the best move. To understand the  $\alpha\beta$  procedure, consider a chess player pondering the next move at position  $P$ . Suppose that the player has already exhaustively analyzed one relatively good move  $M_1$  with corresponding score  $BS(M_1P)$  and proceeds to examine the next move  $M_2$ . Suppose that as the opponent's replies are examined, a particularly strong response is found, which assures that the score of  $M_2$  will be

worse than that of  $M_1$ . Such a response, called a *refutation* of move  $M_2$ , makes further consideration of move  $M_2$  unnecessary (i.e., the portion of the search tree that descends from move  $M_2$  can be discarded). An example is shown in Fig. 6.3.8.



**Figure 6.3.8** The  $\alpha\beta$  procedure. White has evaluated move  $M_1$  to have backed-up score  $(+1)$ , and starts evaluating move  $M_2$ . The first reply of Black is a refutation of  $M_2$ , since it leads to a temporary score of  $-2$ , less than the backed-up score of  $M_1$ . Since the backed-up score of  $M_2$  will be  $-2$  or less,  $M_2$  will be inferior to  $M_1$ . Therefore, it is not necessary to evaluate move  $M_2$  further.

The  $\alpha\beta$  procedure can be generalized to trees of arbitrary or irregular depth and can be incorporated very simply into the minimax algorithm. Generally, if in the process of updating the backed-up score of a given position (step 3b) this score crosses a certain bound, then no further calculation is needed regarding that position. The cutoff bounds are adjusted dynamically as follows:

1. The cutoff bound in position  $n$ , where Black has to move, is denoted  $\alpha$  and equals the highest current score of all ancestor positions of  $n$  where White has to move. The exploration of position  $n$  can be terminated as soon as its temporary backed-up score equals or falls below  $\alpha$ .
2. The cutoff bound in position  $n$ , where White has to move, is denoted  $\beta$  and equals the lowest current value of all ancestor positions of  $n$  where Black has the move. The exploration of position  $n$  can be terminated as soon as its temporary backed-up score rises above  $\beta$ .

The process is illustrated in Fig. 6.3.9. *It can be shown that the backed-up score and optimal move at the starting position are unaffected by the incorporation of the  $\alpha\beta$  procedure in the minimax algorithm.* We leave the verification of this fact to the reader (Exercise 6.8). It can also be seen that *the  $\alpha\beta$  procedure will be more effective if the best moves in each position are explored first.* This tends to keep the  $\alpha$  bounds high and the  $\beta$  bounds low, thus saving a maximum amount of calculation. Current

chess programs use sophisticated techniques for ordering moves so as to maximize the effectiveness of the  $\alpha$ - $\beta$  procedure. We discuss briefly two of these techniques: *iterative deepening* and the *killer heuristic*.

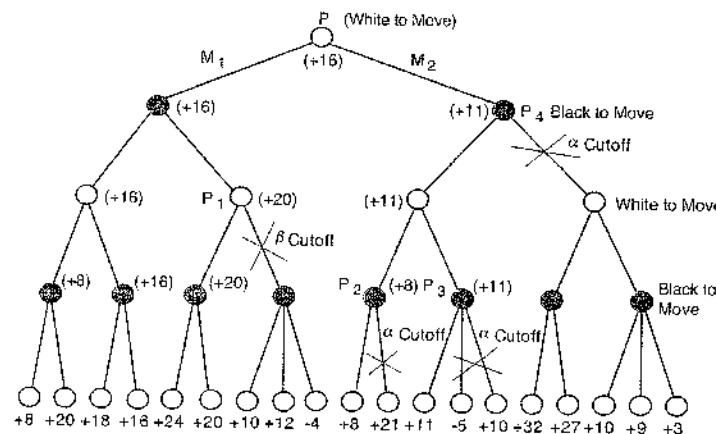


Figure 6.3.9 The  $\alpha$ - $\beta$  procedure applied to the tree of Fig. 6.3.5. For example, the  $\beta$ -cutoff in position  $P_1$  is due to the fact that its temporary score ( $+20$ ) exceeds its current  $\beta$ -bound ( $+16$ ). The  $\alpha$ -cutoffs in positions  $P_2$ ,  $P_3$ , and  $P_4$  are due to the fact that the corresponding temporary scores,  $+8$ ,  $+11$ , and  $+11$ , have fallen below the current  $\alpha$ -bound, which is  $+16$ , the current temporary score in position  $P$ .

Iterative deepening, in its pure form, consists of first conducting a search based on lookahead of one move; then carrying out (from scratch) a search based on lookahead of two moves; then carrying out a search based on lookahead of three moves and so on. This process is continued either up to a fixed level of lookahead or until some limit on computation time is exceeded. At each iteration associated with a certain level of lookahead, one obtains a best move at the starting position, which is examined first in the subsequent iteration that requires one extra move of lookahead. This enhances the power of the  $\alpha$ - $\beta$  procedure, thereby more than making up for the extra computation involved in doing a short lookahead search before doing a longer one. (Actually, given that the number of terminal positions increases on the average by a factor of the order of 30 with each additional level of lookahead, the extra computation is relatively small.) An additional benefit of this method is that a best move is maintained throughout the search and can be produced at any time as needed. This comes in handy in commercial programs that incorporate a feature whereby the computer is forced to move either upon exhausting a given time allocation or upon command by a human opponent. An improvement of the method is to

obtain a thoroughly sorted list of moves at the starting position via a one-move lookahead, and then use the improved ordering in subsequent iterations to enhance the performance of the  $\alpha$ - $\beta$  procedure.

The killer heuristic is similar to iterative deepening in that it aims at examining first the most powerful moves at each position, thereby enhancing the pruning power of the  $\alpha$ - $\beta$  procedure. To understand the idea, suppose that in some position, White selects the first move  $M_1$  from a candidate list  $\{M_1, M_2, M_3, \dots\}$ , and upon examining Black's responses to  $M_1$  finds that a particular move, which we will refer to as the *killer move*, is by far Black's best. Then it is often true that the killer move is also Black's best response to the second and subsequent moves  $M_2, M_3, \dots$  in White's list. It is therefore a good idea from the point of view of  $\alpha$ - $\beta$  pruning to consider the killer move first as a potential response to the remaining moves  $M_2, M_3, \dots$ . Of course, this does not always work as hoped, in which case it is advisable to change the killer move depending on subsequent results of the computation. In fact, some programs maintain lists of more than one killer move at each level of lookahead.

The  $\alpha$ - $\beta$  procedure is safe in the sense that searching a game tree with it and without it will produce the same result. Some computer chess programs use more drastic tree-pruning procedures, which usually require less computation for a given level of lookahead, but may miss on occasion the strongest move. There is some debate at present regarding the merits of such procedures. The books by Levy [Lev84] and Newborn [New75] consider this subject, and provide a broader discussion of the limitations of computer chess programs. A fascinating account of the development of a checkers computer program that implements many of the ideas discussed here is given by Schaeffer [Sch97].

## 6.4 ROLLOUT ALGORITHMS

We now discuss a specific type of cost-to-go approximation within the context of a limited lookahead scheme. Recall that in the one-step lookahead method, at stage  $k$  and state  $x_k$  we use the control  $\bar{p}_k(x_k)$  that attains the minimum in the expression

$$\min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k)) \right\},$$

where  $\tilde{J}_{k+1}$  is some approximation of the true cost-to-go function  $J_{k+1}$ . In the rollout algorithm, the approximating function  $\tilde{J}_{k+1}$  is the cost-to-go of some known heuristic/suboptimal policy  $\pi = \{\mu_0, \dots, \mu_{N-1}\}$ , called *base policy* (see also Example 6.3.1). The policy thus obtained is called the *rollout policy* based on  $\pi$ . Thus the *rollout policy* is a one-step lookahead

policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy.

The process of starting from some suboptimal policy and generating another policy using the one-step lookahead process described above is also called *policy improvement*. This process will be discussed in Section 7.2 and in Vol. II in the context of the *policy iteration* method, which is a primary method for solving infinite horizon problems.

Note that it is also possible to define rollout policies that make use of multistep (say,  $l$ -step) lookahead. Here we assign to every state  $x$  that can be reached in  $t$  steps, the exact cost-to-go of the base policy, as computed by Monte Carlo simulation of several trajectories that start at  $x$ . Clearly, such multistep lookahead involves much more on-line computation, but it may yield better performance than its single-step counterpart. In what follows, we concentrate on rollout policies with single-step lookahead.

The viability of a rollout policy depends on how much time is available to choose the control following the transition to state  $x$  and on how expensive the Monte Carlo evaluation of the expected value

$$E\{g_k(x_k, u_k, w_k) + \hat{J}_{k+1}(f_k(x_k, u_k, w_k))\}$$

is. In particular, it must be possible to perform the Monte Carlo simulations and calculate the rollout control within the real-time constraints of the problem. If the problem is deterministic, a single simulation trajectory suffices, and the calculations are greatly simplified, but in general, the computational overhead can be substantial.

It is possible, however, to speed up the calculation of the rollout policy if we are willing to accept some potential performance degradation. For example, we may use an approximation  $\hat{J}_{k+1}$  of  $J_{k+1}$  to identify a few promising controls through a minimization of the form

$$\min_{u_k \in U_k(x_k)} E\{g_k(x_k, u_k, w_k) + \hat{J}_{k+1}(f_k(x_k, u_k, w_k))\},$$

and then restrict attention to these controls, using fairly accurate Monte Carlo simulation. In particular, the required values of  $\hat{J}_{k+1}$  may be obtained by performing approximately the Monte Carlo simulation, using a limited number of representative trajectories. Adaptive variants of this approach are also possible, whereby we use some heuristics to adjust the accuracy of the Monte Carlo simulation depending on the results of the computation.

Generally, it is important to use as base policy one whose expected cost-to-go is conveniently calculated. The following is an example.

#### Example 6.4.1 (The Quiz Problem)

Consider the quiz problem of Example 4.5.1, where a person is given a list of  $N$  questions and can answer these questions in any order he/she chooses.

Question  $i$  will be answered correctly with probability  $p_i$ , and the person will then receive a reward  $v_i$ . At the first incorrect answer, the quiz terminates and the person is allowed to keep his or her previous rewards. The problem is to choose the ordering of questions so as to maximize the total expected reward.

We saw that the optimal sequence can be obtained using an interchange argument: questions should be answered in decreasing order of the “index of preference”  $p_i v_i / (1 - p_i)$ . We refer to this as the *index policy*. Unfortunately, with only minor changes in the structure of the problem, the index policy need not be optimal. Examples of difficult variations of the problem may involve one or more of the following characteristics:

- (a) A limit on the maximum number of questions that can be answered, which is smaller than the number of questions  $N$ . To see that the index policy is not optimal anymore, consider the case where there are two questions, only one of which may be answered. Then it is optimal to answer the question that offers the maximum expected reward  $p_i v_i$ .
- (b) A time window for each question, which constrains the set of time slots when each question may be answered. Time windows may also be combined with the option to refuse answering a question at a given period, when either no question is available during the period, or answering any one of the available questions involves excessive risk.
- (c) Precedence constraints, whereby the set of questions that can be answered in a given time slot depends on the immediately preceding question, and possibly on some earlier answered questions.
- (d) Sequence-dependent rewards, whereby the reward from answering correctly a given question depends on the immediately preceding question, and possibly on some questions answered earlier.

Nonetheless, even when the index policy is not optimal, it can conveniently be used as a base policy for the rollout algorithm. The reason is that at a given state, the index policy together with its expected reward can be easily calculated. In particular, each feasible question order  $(i_1, \dots, i_N)$  has expected reward equal to

$$p_{i_1} \left( v_{i_1} + p_{i_2} (v_{i_2} + p_{i_3} (\dots + p_{i_N} v_{i_N}) \dots) \right).$$

Thus the rollout algorithm based on the index heuristic operates as follows: at a state where a given subset of questions have already been answered, we consider the set of questions  $J$  that are eligible to be answered next. For each question  $j \in J$ , we consider a sequence of questions that starts with  $j$  and continues with the remaining questions chosen according to the index rule. We compute the expected reward of the sequence, denoted  $R(j)$ , using the above formula. Then among the questions  $j \in J$ , we choose to answer next the one with maximal  $R(j)$ . A computational study of rollout algorithms for the quiz problem and some variations, using several methods for approximating the cost-to-go of the base policy, is given by Bertsekas and Castanon [BeC99].

### Cost Improvement with a Rollout Algorithm

Rollout policies have a nice property: in their pure form, they always result in improved performance over the corresponding base policy. This is essentially a consequence of Prop. 6.3.1 (see Example 6.3.1), but for the purpose of convenient reference, we adapt the proof of that proposition to the rollout context. Let  $\bar{J}_k(x_k)$  and  $H_k(x_k)$  be the costs-to-go of the rollout and the base policies, respectively, starting from a state  $x_k$  at time  $k$ . We will show that  $\bar{J}_k(x_k) \leq H_k(x_k)$  for all  $x_k$  and  $k$ , so that the rollout policy  $\bar{\pi}$  is an improved policy over the base policy  $\pi$ . We have  $\bar{J}_N(x_N) = H_N(x_N) = g_N(x_N)$  for all  $x_N$ . Assuming that  $\bar{J}_{k+1}(x_{k+1}) \leq H_{k+1}(x_{k+1})$  for all  $x_{k+1}$ , we have

$$\begin{aligned}\bar{J}_k(x_k) &= E\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \bar{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \\ &\leq E\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + H_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\} \\ &\leq E\{g_k(x_k, \mu_k(x_k), w_k) + H_{k+1}(f_k(x_k, \mu_k(x_k), w_k))\} \\ &= H_k(x_k),\end{aligned}$$

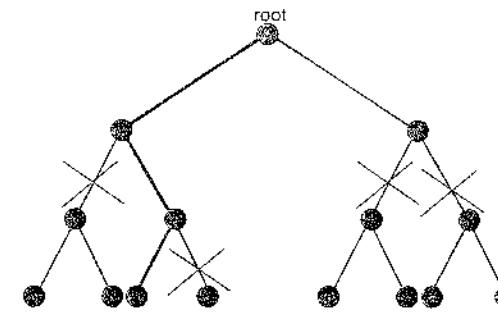
for all  $x_k$ . The first inequality above follows from the induction hypothesis, the second inequality follows from the definition of the rollout policy, and the first and second equalities follow from the DP algorithm that defines the costs-to-go of the rollout and the base policies, respectively. This completes the induction proof that  $\bar{\pi}$  is an improved policy over  $\pi$ .

Empirically, it has been observed that the rollout policy typically produces considerable (and often dramatic) cost improvement over the base policy. However, there is no solid theoretical support for this observation. The following example provides some insight into the nature of cost improvement.

### Example 6.4.2 (The Breakthrough Problem)

Consider a binary tree with  $N$  stages as shown in Fig. 6.4.1. Stage  $k$  of the tree has  $2^k$  nodes, with the node of stage 0 called *root* and the nodes of stage  $N$  called *leaves*. There are two types of tree arcs: *free* and *blocked*. A free (or blocked) arc can (cannot, respectively) be traversed in the direction from the root to the leaves. The objective is to break through the graph with a sequence of free arcs (a free path) starting from the root, and ending at one of the leaves.

One may use DP to discover a free path (if one exists) by starting from the last stage and by proceeding backwards to the root node. The  $k$ th step of the algorithm determines for each node of stage  $N - k$  whether there is a free path from that node to some leaf node, by using the results of the preceding step. The amount of calculation at the  $k$ th step is  $O(2^{N-k})$ . Adding the calculations for the  $N$  stages, we see that the total amount of calculation is  $O(N2^N)$ , so it increases exponentially with the number of stages. For this reason it is interesting to consider heuristics that require calculation that is



**Figure 6.4.1** Binary tree for the breakthrough problem. Each arc is either free or is blocked (crossed out in the figure). The problem is to find a path from the root to one of the leaves, which is free (such as the one shown with thick lines).

linear or polynomial in  $N$ , but may sometimes fail to determine a free path, even when a free path exists.

Thus, one may suboptimally use a *greedy* algorithm, which starts at the root node, selects a free outgoing arc (if one is available), and tries to construct a free path by adding successively nodes to the path. Generally, at the current node, if one of the outgoing arcs is free and the other is blocked, the greedy algorithm selects the free arc. Otherwise, it selects one of the two outgoing arcs according to some fixed rule that depends only on the current node (and not on the status of other arcs). Clearly, the greedy algorithm may fail to find a free path even if such a path exists, as can be seen from Fig. 6.4.1. On the other hand the amount of computation associated with the greedy algorithm is  $O(N)$ , which is much faster than the  $O(N2^N)$  computation of the DP algorithm. Thus we may view the greedy algorithm as a fast heuristic, which is suboptimal in the sense that there are problem instances where it fails while the DP algorithm succeeds.

Let us also consider the rollout algorithm that uses the greedy algorithm as the base heuristic. This algorithm starts at the root and tries to construct a free path by exploring alternative paths constructed by the greedy algorithm. At the current node, it proceeds according to the following two cases:

- (a) If at least one of the two outgoing arcs of the current node is blocked, the rollout algorithm adds to the current path the arc that the greedy algorithm would select at the current node.
- (b) If both outgoing arcs of the current node are free, the rollout algorithm considers the two end nodes of these arcs, and from each of them it runs the greedy algorithm. If the greedy algorithm succeeds in finding a free path that starts from at least one of these nodes, the rollout algorithm stops with a free path having been found; otherwise, the rollout algorithm moves to the node that the greedy algorithm would select at the current node.

Thus, when both outgoing arcs are free, the rollout algorithm explores further the suitability of these arcs, as in case (b) above. Because of this additional discriminatory capability, the rollout algorithm always does at least as well as the greedy (it always finds a free path when the greedy algorithm does, and it also finds a free path in some cases where the greedy algorithm does not). This is consistent with our earlier discussion of the generic cost improvement property of the rollout algorithm over the base heuristic. On the other hand, the rollout algorithm applies the greedy heuristic as many as  $2N$  times, so that it requires  $O(N^2)$  amount of computation – this is intermediate between the  $O(N)$  computation of the greedy and the  $O(N2^N)$  computation of the DP algorithm.

Let us now calculate the probabilities that the algorithms will find a free path given a randomly chosen breakthrough problem. In particular, we generate the graph of the problem randomly, by selecting each of its arcs to be free with probability  $p$ , independently of the other arcs. We then calculate the corresponding probabilities of success for the greedy and the rollout algorithms.

The probability  $G_k$  that the greedy algorithm will find a free path in a graph of  $k$  stages is the probability of a “success” in each of the  $k$  stages, where a success is counted whenever at least one of the two arcs involved is free, an event of probability  $1 - (1-p)^2$  or  $p(2-p)$ . Thus we have, using the independence of the blocked/unblocked status of the arcs,

$$G_k = (p(2-p))^k.$$

The probability  $R_k$  that the rollout algorithm will find a free path in a graph of  $k$  stages can be calculated by means of a recursion, as we now show. At a given node  $n_0$  with  $k$  stages to go, consider the path  $(n_0, n_1, \dots, n_k)$  generated by the greedy algorithm, and let  $(n_0, n_1)$  and  $(n_0, n'_1)$  denote the incident arcs of node  $n_0$ . Let  $P_1$  denote the probability that exactly one of the arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  is free, so

$$P_1 = 2p(1-p),$$

and let  $P_2$  denote the probability that both arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  are free, so

$$P_2 = p^2.$$

To calculate the probability  $R_k$  of the event that the rollout algorithm succeeds in finding a free path, we partition this event into the following four mutually exclusive events, and we calculate their probabilities:

- (1) *Event  $E_1$ .* Exactly one of the arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  is free [by necessity  $(n_0, n_1)$  since it is chosen by the greedy algorithm] and the rollout algorithm finds a free path starting from  $n_1$ . The probability of this event is  $P(E_1) = P_1 R_{k-1}$ .
- (2) *Event  $E_2$ .* Both arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  are free and the greedy algorithm finds a free path starting from  $n_1$ . The probability of this event is  $P(E_2) = P_2 G_{k-1}$ .

- (3) *Event  $E_3$ .* Both the arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  are free, and the greedy algorithm does not find a free path starting from  $n_1$  but finds a free path from  $n'_1$ . The probability of this event is  $P(E_3) = P_2(1 - G_{k-1})G_{k-1}$ .
- (4) *Event  $E_4$ .* Both the arcs  $(n_0, n_1)$  and  $(n_0, n'_1)$  are free, the greedy algorithm does not find a free path starting from either  $n_1$  or  $n'_1$ , but the rollout algorithm finds a free path from  $n_1$ . The probability of this event is  $P_2(1 - G_{k-1})^2 H_{k-1}$ , where  $H_{k-1}$  is the conditional probability that the rollout finds a free path from  $n_1$  given that the greedy does not find a free path from  $n_1$ . We have

$$R_{k-1} = G_{k-1} + (1 - G_{k-1})H_{k-1},$$

so  $(1 - G_{k-1})H_{k-1} = R_{k-1} - G_{k-1}$ , and the probability of the event  $E_4$  is

$$P(E_4) = P_2(1 - G_{k-1})^2 H_{k-1} = P_2(1 - G_{k-1})(R_{k-1} - G_{k-1}).$$

Thus, by adding the probabilities of the above mutually exclusive and collectively exhaustive events, we have

$$\begin{aligned} R_k &= P(E_1) + P(E_2) + P(E_3) + P(E_4) \\ &= P_1 R_{k-1} + P_2(G_{k-1} + (1 - G_{k-1})G_{k-1} + (1 - G_{k-1})(R_{k-1} - G_{k-1})) \\ &= (P_1 + P_2(1 - G_{k-1}))R_{k-1} + P_2G_{k-1}. \end{aligned}$$

From this, by substituting the expressions  $P_1 = 2p(1-p)$  and  $P_2 = p^2$ , we obtain

$$\begin{aligned} R_k &= (2p(1-p) + p^2(1 - G_{k-1}))R_{k-1} + p^2G_{k-1} \\ &= p(2-p)R_{k-1} + p^2G_{k-1}(1 - R_{k-1}), \end{aligned}$$

with the initial condition  $R_0 = 1$ . Since  $\lim_{k \rightarrow \infty} G_k = 0$  and  $p(2-p) < 1$ , it follows from the above equation that  $\lim_{k \rightarrow \infty} R_k = 0$ . Furthermore, by dividing with  $G_k = p(2-p)G_{k-1}$ , we have

$$\frac{R_k}{G_k} = \frac{R_{k-1}}{G_{k-1}} + \frac{p}{2-p}(1 - R_{k-1}),$$

so since  $\lim_{k \rightarrow \infty} R_k = 0$ , we obtain for large  $N$

$$\frac{R_N}{G_N} = O\left(N \frac{p}{2-p}\right).$$

Thus, asymptotically, the rollout algorithm requires  $O(N)$  times more computation, but has an  $O(N)$  times larger probability of finding a free path than the greedy algorithm. This type of tradeoff appears to be qualitatively typical: the rollout algorithm achieves a substantial performance improvement over the base heuristic at the expense of extra computation that is equal to the computation time of the base heuristic times a factor that is a low order polynomial of the problem size.

## Computational Issues in Rollout Algorithms

We now consider in more detail implementation issues and specific properties of rollout algorithms in a variety of settings. To compute the rollout control  $\bar{\mu}_k(x_k)$ , we need for all  $u_k \in U_k(x_k)$  the value of

$$Q_k(x_k, u_k) = E \left\{ g_k(x_k, u_k, w_k) + H_{k+1}(f_k(x_k, u_k, w_k)) \right\},$$

known as the *Q-factor* of  $(x_k, u_k)$  at time  $k$ . Alternatively, for the computation of  $\bar{\mu}_k(x_k)$  we need the value of the cost-to-go

$$H_{k+1}(f_k(x_k, u_k, w_k))$$

of the base policy at all possible next states  $f_k(x_k, u_k, w_k)$ , from which we can compute the required *Q*-factors.

We will focus on the case where a closed form expression of the *Q*-factor is not available. We assume instead that we can simulate the system under the base policy  $\pi$ , and in particular, that we can generate sample system trajectories and corresponding costs consistently with the probabilistic data of the problem. We will consider several cases and possibilities, we will point out their advantages and drawbacks, and we will discuss the contexts within which they are most appropriate. These cases are:

- (1) *The deterministic problem case*, where  $w_k$  takes a single known value at each stage. We provide an extensive discussion of this case, focusing not only on traditional deterministic optimal control problems, but also on quite general combinatorial optimization problems, for which the rollout approach has proved convenient and effective.
- (2) *The stochastic problem case with Q-factors evaluated by Monte-Carlo simulation*. Here, once we are at state  $x_k$ , the *Q*-factors  $Q_k(x_k, u_k)$  are evaluated on-line by Monte-Carlo simulation, for all  $u_k \in U_k(x_k)$ .
- (3) *The stochastic problem case with Q-factors approximated in some way*. One possibility is to use a certainty equivalence approximation, where the problem is genuinely stochastic, but the values  $H_k(x_k)$  are approximated by the cost-to-go of  $\pi$  that would be incurred if the system were replaced by a suitable deterministic system from state  $x_k$  and time  $k$  onward (assumed certainty equivalence). There are also other possibilities based on the use of an approximation architecture and some form of least squares.

We examine each of these three cases in the next three subsections.

### 6.4.1 Discrete Deterministic Problems

Let us assume that the problem is deterministic, i.e., that  $w_k$  can take only one value at each stage  $k$ . Then, starting from state  $x_k$  at stage  $k$ , the

base policy  $\pi$  produces deterministic sequences of states  $\{x_{k+1}, \dots, x_N\}$  and controls  $\{u_k, \dots, u_{N-1}\}$  such that

$$x_{i+1} = f(x_i, u_i), \quad i = k, \dots, N-1,$$

and a cost

$$g_k(x_k, u_k) + \dots + g_{N-1}(x_{N-1}, u_{N-1}) + g_N(x_N).$$

Thus the *Q*-factor

$$Q_k(x_k, u_k) = g_k(x_k, u_k) + H_{k+1}(f_k(x_k, u_k))$$

can be obtained by running  $\pi$  starting from state  $f_k(x_k, u_k)$  and time  $k+1$ , and recording the corresponding cost  $H_{k+1}(f_k(x_k, u_k))$ . The rollout control  $\bar{\mu}_k(x_k)$  is obtained by calculating in this manner the *Q*-factors  $Q_k(x_k, u_k)$  for all  $u_k \in U_k(x_k)$ , and setting

$$\bar{\mu}_k(x_k) = \arg \min_{u_k \in U_k(x_k)} Q_k(x_k, u_k).$$

Aside for being convenient for the deterministic special case of the basic problem of Chapter 1, this rollout method can be adapted for general discrete or combinatorial optimization problems that do not necessarily have the strong sequential character of the basic problem. For such problems the rollout approach provides a convenient and broadly applicable suboptimal solution method that goes beyond and indeed enhances the common types of heuristics, such as greedy algorithms, local search, genetic algorithms, tabu search, and others.

To illustrate the ideas involved, let us consider the problem

$$\begin{aligned} & \text{minimize } G(u) \\ & \text{subject to } u \in U \end{aligned} \tag{6.31}$$

where  $U$  is a finite set of feasible solutions and  $G(u)$  is a cost function. We assume that each solution  $u$  has  $N$  components; that is, it has the form  $u = (u_1, u_2, \dots, u_N)$ , where  $N$  is a positive integer. Under this assumption, we can view the problem as a sequential decision problem, where the components  $u_1, \dots, u_N$  are selected one-at-a-time. An  $n$ -tuple  $(u_1, \dots, u_n)$  consisting of the first  $n$  components of a solution is called an *n-solution*. We associate  $n$ -solutions with the  $n$ th stage of a DP problem. In particular, for  $n = 1, \dots, N$ , the states of the  $n$ th stage are of the form  $(u_1, \dots, u_n)$ . The initial state is a dummy (artificial) state. From this state we may move to any state  $(u_1)$ , with  $u_1$  belonging to the set

$$U_1 = \{\tilde{u}_1 \mid \text{there exists a solution of the form } (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N) \in U\}.$$

Thus  $U_1$  is the set of choices of  $u_1$  that are consistent with feasibility.

More generally, from a state of the form  $(u_1, \dots, u_{n-1})$ , we may move to any state of the form  $(u_1, \dots, u_{n-1}, u_n)$ , with  $u_n$  belonging to the set

$$U_n(u_1, \dots, u_{n-1}) = \{u_n \mid \text{there exists a solution of the form } (u_1, \dots, u_{n-1}, u_n, \dots, u_N) \in U\}. \quad (6.32)$$

The choices available at state  $(u_1, \dots, u_{n-1})$  are  $u_n \in U_n(u_1, \dots, u_{n-1})$ . These are the choices of  $u_n$  that are consistent with the preceding choices  $u_1, \dots, u_{n-1}$ , and are also consistent with feasibility. The terminal states correspond to the  $N$ -solutions  $(u_1, \dots, u_N)$ , and the only nonzero cost is the terminal cost  $G(u_1, \dots, u_N)$ .

Let  $J^*(u_1, \dots, u_n)$  denote the optimal cost starting from the  $n$ -solution  $(u_1, \dots, u_n)$ , i.e., the optimal cost of the problem over solutions whose first  $n$  components are constrained to be equal to  $u_i$ ,  $i = 1, \dots, n$ , respectively. If we knew the optimal cost-to-go function  $J^*(u_1, \dots, u_n)$ , we could construct an optimal solution by a sequence of  $N$  single component minimizations. In particular, an optimal solution  $(u_1^*, \dots, u_N^*)$  could be obtained through the algorithm

$$u_i^* = \arg \min_{u_i \in U_i(u_1^*, \dots, u_{i-1}^*)} J^*(u_1^*, \dots, u_{i-1}^*, u_i), \quad i = 1, \dots, N.$$

Unfortunately, this is seldom viable, because of the prohibitive computation required to obtain  $J^*(u_1, \dots, u_n)$ .

Assume now that we have a heuristic, which starting from an  $n$ -solution  $(u_1, \dots, u_n)$ , produces an  $N$ -solution  $(u_1, \dots, u_n, u_{n+1}, \dots, u_N)$  whose cost is denoted by  $H(u_1, \dots, u_n)$ . Such a heuristic may be viewed as a base policy for the problem, in the sense that given the current state  $(u_1, \dots, u_n)$  it generates the next decision  $u_{n+1}$  as the first component of the remaining portion  $(u_{n+1}, \dots, u_N)$  of the solution. Let us consider the corresponding rollout algorithm. It can be seen that this algorithm selects as the first solution component

$$\bar{u}_1 = \arg \min_{u_1 \in U_1} H(u_1),$$

and operates sequentially, for  $n = 1, \dots, N - 1$ , as follows:

Given a partial solution  $(\bar{u}_1, \dots, \bar{u}_n)$ , it runs the heuristic starting from the partial solutions  $(\bar{u}_1, \dots, \bar{u}_n, u_{n+1})$  corresponding to all the possible next solution components  $u_{n+1} \in U_{n+1}(\bar{u}_1, \dots, \bar{u}_n)$ , and selects as next component

$$\bar{u}_{n+1} = \arg \min_{u_{n+1} \in U_{n+1}(\bar{u}_1, \dots, \bar{u}_n)} H(\bar{u}_1, \dots, \bar{u}_n, u_{n+1}).$$

In order to analyze most economically the preceding algorithm and its variants, we will embed it within a more general and flexible framework for

discrete optimization. To this end, we introduce a graph search problem, which contains as special cases broad classes of discrete/integer optimization problems, and will serve as the context of our methodology. We will then describe and analyze a basic form of a one-step lookahead algorithm, we will discuss some of its variations, we will illustrate it by means of some examples, and we will discuss its connection with the DP context.

As we explain later (see the end of Section 6.4.1), the algorithm to be introduced is not quite a rollout algorithm in the sense discussed so far, because, strictly speaking, it does not use the cost of a heuristic policy as a one-step lookahead cost approximation, except under a special assumption (the sequential consistency assumption, to be described later). The basic idea of the algorithm is, however, very close to a rollout: it is a one-step lookahead policy with cost approximation *derived* from a heuristic. Thus, with a stretch of terminology, we will call this algorithm “rollout” as well.

#### The Basic Rollout Algorithm for Discrete Optimization

Let us introduce a graph search problem that will serve as a general model for discrete optimization. We are given a graph with set of nodes  $\mathcal{N}$ , set of arcs  $\mathcal{A}$ , and a special node  $s$ , which we call the *origin*. The arcs are directed in the sense that arc  $(i, j)$  is distinct from arc  $(j, i)$ . We are also given a subset  $\bar{\mathcal{N}}$  of nodes, called *destinations*, and a cost  $g(i)$  for each destination  $i$ . The destination nodes are terminal in the sense that they have no outgoing arcs. For simplicity, we assume that the node and arc sets,  $\mathcal{N}$  and  $\mathcal{A}$  contain a finite number of elements. However, the following analysis and discussion applies, with minor modifications in language, to the case of a countably infinite number of nodes and a finite set of outgoing arcs from each node. We want to find a path that starts at the origin  $s$ , ends at one of the destination nodes  $i \in \bar{\mathcal{N}}$ , and is such that the cost  $g(i)$  is minimized.

In the context of the discrete optimization problem (6.31), nodes  $i$  correspond to  $n$ -tuples  $(u_1, \dots, u_n)$  consisting of the first  $n$  components of a solution, where  $n = 1, \dots, N$ . Arcs lead from nodes of the form  $(u_1, \dots, u_{n-1})$  to nodes of the form  $(u_1, \dots, u_{n-1}, u_n)$ , and there is an arc for each  $u_n$  of the form (6.32). An interesting property of this special case is that its associated graph is acyclic.

In our terminology, a path is a sequence of arcs

$$(i_1, i_2), (i_2, i_3), \dots, (i_{m-1}, i_m),$$

all of which are oriented in the forward direction. The nodes  $i_1$  and  $i_m$  are called the *start node* and the *end node* of the path, respectively. For convenience, and without loss of generality,<sup>f</sup> we will assume that given an

<sup>f</sup> In the case where there are multiple arcs connecting a node pair, we can merge all these arcs to a single arc, since the set of destination nodes that can be reached from any non-destination node will not be affected.

ordered pair of nodes  $(i, j)$ , there is at most one arc with start node  $i$  and end node  $j$ , which (if it exists) will be denoted by  $(i, j)$ . In this way, a path consisting of arcs  $(i_1, i_2), (i_2, i_3), \dots, (i_{m-1}, i_m)$  is unambiguously specified as the sequence of nodes  $(i_1, i_2, \dots, i_m)$ .

Let us assume that we have a heuristic path construction algorithm, denoted  $\mathcal{H}$ , which given a non-destination node  $i \notin \bar{\mathcal{N}}$ , constructs a path  $(i, i_1, \dots, i_m, \bar{i})$  starting at  $i$  and ending at one of the destination nodes  $\bar{i}$ . Implicit in this assumption is that for every non-destination node, there exists at least one path starting at that node and ending at some destination node. We refer to the algorithm  $\mathcal{H}$  as the *base heuristic*, since we will use this algorithm as the basic building block for constructing the rollout algorithm to be introduced shortly.

The end node  $\bar{i}$  of the path constructed by the base heuristic  $\mathcal{H}$  is completely specified by the start node  $i$ . We call  $\bar{i}$  the *projection of  $i$  under  $\mathcal{H}$* , and we denote it by  $p(i)$ . We denote the corresponding cost by  $H(i)$ ,

$$H(i) = g(p(i)).$$

The projection of a destination node is the node itself by convention, so that for all  $i \in \bar{\mathcal{N}}$  we have  $i = p(i)$  and  $H(i) = g(i)$ . Note that while the base heuristic  $\mathcal{H}$  will generally yield a suboptimal solution, the path that it constructs may involve a fairly sophisticated suboptimization. For example,  $\mathcal{H}$  may construct several paths ending at destination nodes according to some heuristics, and then select the path that yields minimal cost.

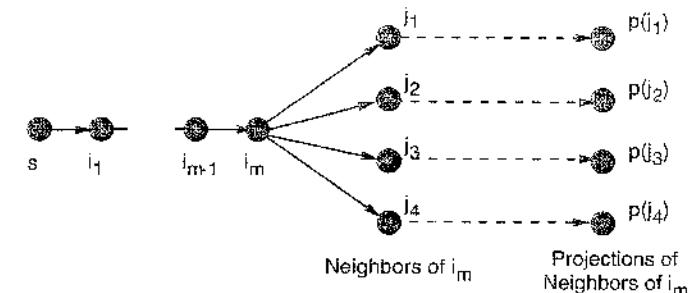
One possibility for suboptimal solution of the problem is to start at the origin  $s$  and use the base heuristic  $\mathcal{H}$  to obtain the projection  $p(s)$ . We instead propose to use  $\mathcal{H}$  to construct a path to a destination node sequentially. At the typical step of the sequence, we consider all downstream neighbors  $j$  of a node  $i$ , we run the base heuristic  $\mathcal{H}$  starting from each of these neighbors, and obtain the corresponding projections and costs. We then move to the neighbor that gives the best projection. This sequential version of  $\mathcal{H}$  is called the *rollout algorithm based on  $\mathcal{H}$* , and is denoted by  $\mathcal{RH}$ .

To formally describe the rollout algorithm, let  $N(i)$  denote the set of downstream neighbors of node  $i$ ,

$$N(i) = \{j \mid (i, j) \text{ is an arc}\}.$$

Note that  $N(i)$  is nonempty for every non-destination node  $i$ , since by assumption there exists a path starting at  $i$  and ending at its projection  $p(i)$ . The rollout algorithm  $\mathcal{RH}$  starts with the origin node  $s$ . At the typical step, given a node sequence  $(s, i_1, \dots, i_m)$ , where  $i_m$  is not a destination,  $\mathcal{RH}$  adds to the sequence a node  $i_{m+1}$  such that

$$i_{m+1} = \arg \min_{j \in N(i_m)} H(j). \quad (6.33)$$



**Figure 6.4.2** Illustration of the rollout algorithm. After  $m$  steps of the algorithm, we have the path  $(s, i_1, \dots, i_m)$ . To extend this path at the next step, we generate the set  $N(i_m)$  of neighbors of the terminal node  $i_m$ , and select from this set the neighbor that has the best projection, i.e.

$$i_{m+1} = \arg \min_{j \in N(i_m)} H(j) = \arg \min_{j \in N(i_m)} g(p(j)).$$

If  $i_{m+1}$  is a destination node,  $\mathcal{RH}$  terminates. Otherwise, the process is repeated with the sequence  $(s, i_1, \dots, i_m, i_{m+1})$  replacing  $(s, i_1, \dots, i_m)$ ; see Fig. 6.4.2.

Note that once  $\mathcal{RH}$  has terminated with a path  $(s, i_1, \dots, i_m)$ , we will have obtained the projection  $p(i_k)$  of each of the nodes  $i_k$ ,  $k = 1, \dots, m$ . The best of these projections yields a cost

$$\min_{k=1, \dots, m} H(i_k) = \min_{k=1, \dots, m} g(p(i_k)),$$

and the projection that corresponds to the minimum above may be taken as the final (suboptimal) solution produced by the rollout algorithm. We may also compare the above minimal cost with the cost  $g(p(s))$  of the projection  $p(s)$  of the origin, and use  $p(s)$  as the final solution if it produces a smaller cost. This will ensure that the rollout algorithm will produce a solution that is no worse than the one produced by the base heuristic.

### Example 6.4.3 (Traveling Salesman Problem)

Let us consider the traveling salesman problem, whereby a salesman wants to find a minimum mileage/cost tour that visits each of  $N$  given cities exactly once and returns to the city he started from. We associate a node with each city  $i = 1, \dots, N$ , and we introduce an arc  $(i, j)$  with traversal cost  $a_{ij}$  for each ordered pair of nodes  $i$  and  $j$ . Note that we assume that the graph is complete; that is, there exists an arc for each ordered pair of nodes. There is no loss of generality in doing so because we can assign a very high cost  $a_{ij}$  to

an arc  $(i, j)$  that is precluded from participation in the solution. The problem is to find a cycle that goes through all the nodes exactly once and whose sum of arc costs is minimum.

There are many heuristic approaches for solving the traveling salesman problem. For illustration purposes, let us restrict attention to the simple *nearest neighbor* heuristic. Here, we start from a path consisting of just a single node  $i_1$  and at each iteration, we enlarge the path with a node that does not close a cycle and minimizes the cost of the enlargement. In particular, after  $k$  iterations, we have a path  $\{i_1, \dots, i_k\}$  consisting of distinct nodes, and at the next iteration, we add an arc  $(i_k, i_{k+1})$  that minimizes  $a_{i_k i_{k+1}}$  over all arcs  $(i_k, i)$  with  $i \neq i_1, \dots, i_k$ . After  $N - 1$  iterations, all nodes are included in the path, which is then converted to a tour by adding the final arc  $(i_N, i_1)$ .

We can formulate the traveling salesman problem as a graph search problem as follows: There is a chosen starting city, say  $i_1$  corresponding to the origin of the graph search problem. Each node of the graph search problem corresponds to a path  $(i_1, i_2, \dots, i_k)$ , where  $i_1, i_2, \dots, i_k$  are distinct cities. The neighbor nodes of the path  $(i_1, i_2, \dots, i_k)$  are paths of the form  $(i_1, i_2, \dots, i_k, i_{k+1})$  which correspond to adding one more unvisited city  $i_{k+1} \neq i_1, i_2, \dots, i_k$  at the end of the path. The destinations are the cycles of the form  $(i_1, i_2, \dots, i_N)$ , and the cost of a destination in the graph search problem is the cost of the corresponding cycle. Thus a path from the origin to a destination in the graph search problem corresponds to constructing a cycle in  $N - 1$  arc addition steps, and at the end incurring the cost of the cycle.

Let us now use as base heuristic the nearest neighbor method. The corresponding rollout algorithm operates as follows: After  $k$  iterations, we have a path  $\{i_1, \dots, i_k\}$  consisting of distinct nodes. At the next iteration, we run the nearest neighbor heuristic starting from each of the paths of the form  $\{i_1, \dots, i_k, i\}$  where  $i \neq i_1, \dots, i_k$ , and obtain a corresponding cycle. We then select as next node  $i_{k+1}$  of the path the node  $i$  that corresponds to the best cycle thus obtained.

### Termination and Sequential Consistency

We say that the rollout algorithm  $\mathcal{RH}$  is *terminating* if it is guaranteed to terminate finitely starting from any node. Contrary to the base heuristic  $\mathcal{H}$ , which by definition, has the property that it yields a path terminating at a destination starting from any node, the rollout algorithm  $\mathcal{RH}$  need not have this property in the absence of additional conditions. The termination question can usually be resolved quite easily, and we will now discuss a few different methods by which this can be done.

One important case where  $\mathcal{RH}$  is terminating is *when the graph is acyclic*, since then the nodes of the path generated by  $\mathcal{RH}$  cannot be repeated within the path, and their number is bounded by the number of nodes in  $\mathcal{N}$ . As a first step towards developing another case where  $\mathcal{RH}$  is terminating, we introduce the following definition, which will also set the stage for further analysis of the properties of  $\mathcal{RH}$ .

**Definition 6.4.1:** We say that the base heuristic  $\mathcal{H}$  is *sequentially consistent* if for every node  $i$ , it has the following property: If  $\mathcal{H}$  generates the path  $(i, i_1, \dots, i_m, \bar{i})$  when it starts at  $i$ , it generates the path  $(i_1, \dots, i_m, \bar{i})$  when it starts at the node  $i_1$ .

Thus  $\mathcal{H}$  is sequentially consistent if all the nodes of a path that it generates have the same projection. There are many examples of sequentially consistent algorithms that are used as heuristics in combinatorial optimization, including the following.

### Example 6.4.4 (Greedy Algorithms as Base Heuristics)

Suppose that we have a function  $F$ , which for each node  $i$ , provides a scalar estimate  $F(i)$  of the optimal cost starting from  $i$ , that is, the minimal cost  $g(\bar{i})$  that can be obtained with a path that starts at  $i$  and ends at one of the destination nodes  $\bar{i} \in \bar{\mathcal{N}}$ . Then  $F$  can be used to define a base heuristic, called the *greedy algorithm with respect to  $F$* , as follows:

The greedy algorithm starts at a node  $i$  with the (degenerate) path that consists of just node  $i$ . At the typical step, given a path  $(i, i_1, \dots, i_m)$ , where  $i_m$  is not a destination, the algorithm adds to the path a node  $i_{m+1}$  such that

$$i_{m+1} = \arg \min_{j \in N(i_m)} F(j). \quad (6.34)$$

In the case where  $i_{m+1}$  is a destination, the algorithm terminates with the path  $(i, i_1, \dots, i_m, i_{m+1})$ . Otherwise, the process is repeated with the path  $(i, i_1, \dots, i_m, i_{m+1})$  replacing  $(i, i_1, \dots, i_m)$ .

An example of a greedy algorithm is the nearest neighbor heuristic for the traveling salesman problem (cf. Example 6.4.3). Recall from that example that nodes of the graph search problem correspond to paths (sequences of distinct cities), and a transition to a neighbor node corresponds to adding one more unvisited city to the end of the current path. The function  $F$  in the nearest neighbor heuristic specifies the cost of the addition of the new city.

It is also interesting to note that by viewing  $F$  as a cost-to-go approximation, we may consider the greedy algorithm to be a special type of one-step lookahead policy. Furthermore, if  $F(j)$  is chosen to be the cost obtained by some base heuristic starting from  $j$ , then the greedy algorithm becomes the corresponding rollout algorithm. Thus, it may be said that the rollout algorithm is a special case of a greedy algorithm. However, the particular choice of  $F$  used in the rollout algorithm is responsible for special properties that are not shared by other types of greedy algorithms.

Let us denote by  $\mathcal{H}$  the greedy algorithm described above and assume that it terminates starting from every node (this has to be verified independently). Let us also assume that whenever there is a tie in the minimization of Eq. (6.34),  $\mathcal{H}$  resolves the tie in a manner that is fixed and independent of the starting node  $i$  of the path, e.g., by resolving the tie in favor of the

numerically smallest node  $j$  that attains the minimum in Eq. (6.34). Then it can be seen that  $\mathcal{H}$  is sequentially consistent, since by construction, every node on a path generated by  $\mathcal{H}$  has the same projection.

For a sequentially consistent base heuristic  $\mathcal{H}$ , we will assume a restriction in the way the rollout algorithm  $\mathcal{RH}$  resolves ties in selecting the next node on its path: this restriction will guarantee that  $\mathcal{RH}$  is terminating. In particular, suppose that after  $m$  steps,  $\mathcal{RH}$  has produced the node sequence  $(s, i_1, \dots, i_m)$ , and that the path generated by  $\mathcal{H}$  starting from  $i_m$  is  $(i_m, i_{m+1}, i_{m+2}, \dots, \bar{i})$ . Suppose that among the neighbor set  $N(i_m)$ , the node  $i_{m+1}$  attains the minimum in the selection test

$$\min_{j \in N(i_m)} H(j). \quad (6.35)$$

but there are also some other nodes, in addition to  $i_{m+1}$ , that attain this minimum. Then, we require that the tie is broken in favor of  $i_{m+1}$ , i.e., that the next node added to the current sequence  $(s, i_1, \dots, i_m)$  is  $i_{m+1}$ . Under this convention for tie-breaking, we show in the following proposition that the rollout algorithm  $\mathcal{RH}$  terminates at a destination and yields a cost that is no larger than the cost yielded by the base heuristic  $\mathcal{H}$ .†

**Proposition 6.4.1:** Let the base heuristic  $\mathcal{H}$  be sequentially consistent. Then the rollout algorithm  $\mathcal{RH}$  is terminating. Furthermore, if  $(i_1, \dots, i_m)$  is the path generated by  $\mathcal{RH}$  starting from a non-destination node  $i_1$  and ending at a destination node  $i_m$ , the cost of  $\mathcal{RH}$  starting from  $i_1$  is less or equal to the cost of  $\mathcal{H}$  starting from  $i_1$ . In particular, we have

$$H(i_1) \geq H(i_2) \geq \dots \geq H(i_{m-1}) \geq H(i_m). \quad (6.36)$$

† For an example where this convention for tie-breaking is not observed and as a consequence  $\mathcal{RH}$  does not terminate, assume that there is a single destination  $d$  and that all other nodes are arranged in a cycle. Each non-destination node  $i$  has two outgoing arcs: one arc that belongs to the cycle, and another arc which is  $(i, d)$ . Let  $\mathcal{H}$  be the (sequentially consistent) base heuristic that starting from a node  $i \neq d$ , generates the path  $(i, d)$ . When the terminal node of the path is node  $i$ , the rollout algorithm  $\mathcal{RH}$  compares the two neighbors of  $i$ , which are  $d$  and the node next to  $i$  on the cycle, call it  $j$ . Both neighbors have  $d$  as their projection, so there is tie in Eq. (6.35). It can be seen that if  $\mathcal{RH}$  breaks ties in favor of the neighbor  $j$  that lies on the cycle, then  $\mathcal{RH}$  continually repeats the cycle and never terminates.

Furthermore, for all  $m = 1, \dots, \tilde{m}$ ,

$$H(i_m) = \min \left\{ H(i_1), \min_{j \in N(i_1)} H(j), \dots, \min_{j \in N(i_{m-1})} H(j) \right\}. \quad (6.37)$$

**Proof:** Let  $i_m$  and  $i_{m+1}$  be two successive nodes generated by  $\mathcal{RH}$ , and let  $(i_m, i'_{m+1}, i''_{m+2}, \dots, \bar{i}_m)$  be the path generated by  $\mathcal{H}$  starting from  $i_m$ , where  $\bar{i}_m$  is the projection of  $i_m$ . Then, since  $\mathcal{H}$  is sequentially consistent, we have

$$H(i_m) = H(i'_{m+1}) = g(\bar{i}_m).$$

Furthermore, since  $i'_{m+1} \in N(i_m)$ , we have using the definition of  $\mathcal{RH}$  [cf. Eq. (6.33)]

$$H(i'_{m+1}) > \min_{j \in N(i_m)} H(j) = H(i_{m+1}).$$

Combining the last two relations, we obtain

$$H(i_m) \geq H(i_{m+1}) = \min_{j \in N(i_m)} H(j). \quad (6.38)$$

To show that  $\mathcal{RH}$  is terminating, note that in view of Eq. (6.38), either  $H(i_m) > H(i_{m+1})$ , or else  $H(i_m) = H(i_{m+1})$ . In the latter case, in view of the convention for breaking ties that occur in Eq. (6.35) and the sequential consistency of  $\mathcal{H}$ , the path generated by  $\mathcal{H}$  starting from  $i_{m+1}$  is the tail portion of the path generated by  $\mathcal{H}$  starting from  $i_m$ , and has one arc less. Thus the number of nodes generated by  $\mathcal{RH}$  between successive times that the inequality  $H(i_m) > H(i_{m+1})$  holds is finite. On the other hand, the inequality  $H(i_m) > H(i_{m+1})$  can occur only a finite number of times, since the number of destination nodes is finite, and the destination node of the path generated by  $\mathcal{H}$  starting from  $i_m$  cannot be repeated if the inequality  $H(i_m) > H(i_{m+1})$  holds. Therefore,  $\mathcal{RH}$  is terminating.

If  $(i_1, \dots, i_m)$  is the path generated by  $\mathcal{RH}$ , the relation (6.38) implies the desired relations (6.36) and (6.37). Q.E.D.

Proposition 6.4.1 shows that in the sequentially consistent case, the rollout algorithm  $\mathcal{RH}$  has an important “automatic cost sorting” property, whereby it follows the best path generated by the base heuristic  $\mathcal{H}$ . In particular, when  $\mathcal{RH}$  generates a path  $(i_1, \dots, i_m)$ , it does so by using  $\mathcal{H}$  to generate a collection of other paths and corresponding projections starting from all the successor nodes of the intermediate nodes  $i_1, \dots, i_{m-1}$ . However,  $(i_1, \dots, i_m)$  is guaranteed to be the best among this path collection and  $i_m$  has minimal cost among all generated projections [cf. Eq. (6.37)]. Of course this does not guarantee that the path generated by  $\mathcal{RH}$  will be a

near-optimal path, because the collection of paths generated by  $\mathcal{H}$  may be “poor.” Still, the property whereby  $\mathcal{RH}$  at all times follows the best path found so far is intuitively reassuring.

The following example illustrates the preceding concepts.

#### Example 6.4.5 (One-Dimensional Walk)

Consider a person who walks on a straight line and at each time period takes either a unit step to the left or a unit step to the right. There is a cost function assigning cost  $g(i)$  to each integer  $i$ . Given an integer starting point on the line, the person wants to minimize the cost of the point where he will end up after a given and fixed number  $N$  of steps.

We can formulate this problem as a graph search problem of the type discussed in the preceding section. In particular, without loss of generality, let us assume that the starting point is the origin, so that the person's position after  $n$  steps will be some integer in the interval  $[-n, n]$ . The nodes of the graph are identified with pairs  $(k, m)$ , where  $k$  is the number of steps taken so far ( $k = 1, \dots, N$ ) and  $m$  is the person's position ( $m \in [-k, k]$ ). A node  $(k, m)$  with  $k < N$  has two outgoing arcs with end nodes  $(k+1, m-1)$  (corresponding to a left step) and  $(k+1, m+1)$  (corresponding to a right step). The starting state is  $(0, 0)$  and the terminating states are of the form  $(N, m)$ , where  $m$  is of the form  $N - 2l$  and  $l \in [0, N]$  is the number of left steps taken.

Let the base heuristic  $\mathcal{H}$  be defined as the algorithm, which, starting at a node  $(k, m)$ , takes  $N - k$  successive steps to the right and terminates at the node  $(N, m + N - k)$ . Note that  $\mathcal{H}$  is sequentially consistent. The rollout algorithm  $\mathcal{RH}$ , at node  $(k, m)$  compares the cost of the destination node  $(N, m + N - k)$  (corresponding to taking a step to the right and then following  $\mathcal{H}$ ) and the cost of the destination node  $(N, m + N - k - 2)$  (corresponding to taking a step to the left and then following  $\mathcal{H}$ ).

Let us say that an integer  $i \in [-N + 2, N - 2]$  is a *local minimum* if  $g(i - 2) \geq g(i)$  and  $g(i) \leq g(i + 2)$ . Let us also say that  $N$  (or  $-N$ ) is a local minimum if  $g(N - 2) \leq g(N)$  [or  $g(-N) \leq g(-N + 2)$ , respectively]. Then it can be seen that starting from the origin  $(0, 0)$ ,  $\mathcal{RH}$  obtains the local minimum that is closest to  $N$ , (see Fig. 6.4.3). This is no worse (and typically better) than the integer  $N$  obtained by  $\mathcal{H}$ . Note that if  $g$  has a unique local minimum in the set of integers in the range  $[-N, N]$ , the minimum must also be global, and it will be found by  $\mathcal{RH}$ . This example illustrates how  $\mathcal{RH}$  may exhibit “intelligence” that is totally lacking from  $\mathcal{H}$ , and is in agreement with the result of Prop. 6.4.1.

#### Sequential Improvement

It is possible to show that the rollout algorithm improves on the base heuristic (cf. Prop. 6.4.1) under weaker conditions. To this end we introduce the following definition.

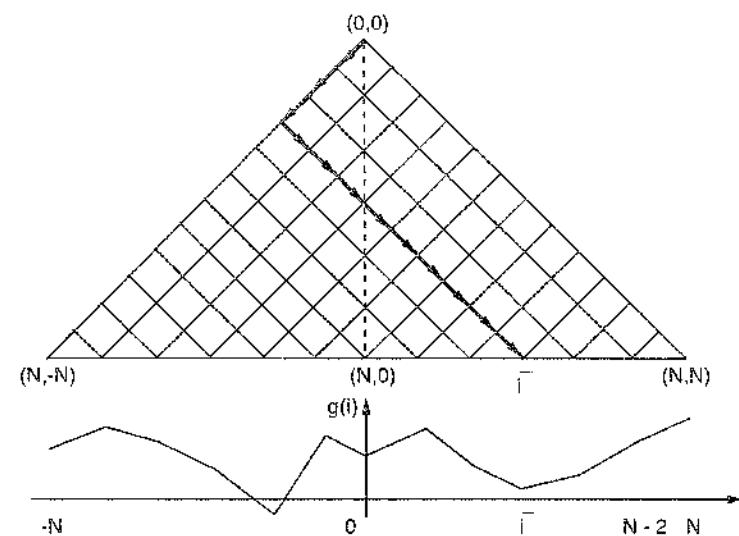


Figure 6.4.3 Illustration of the path generated by the rollout algorithm  $\mathcal{RH}$  in Example 6.4.5. The algorithm keeps moving to the left up to the time where the base heuristic  $\mathcal{H}$  generates two destinations  $(N, \bar{i})$  and  $(N, \bar{i} - 2)$  with  $g(\bar{i}) \leq g(\bar{i} - 2)$ . Then it continues to move to the right ending at the destination  $(N, \bar{i})$ , which corresponds to the local minimum closest to  $N$ .

**Definition 6.4.2:** We say that the base heuristic  $\mathcal{H}$  is *sequentially improving* if for every non-destination node  $i$ , we have

$$H(i) \geq \min_{j \in N(i)} H(j). \quad (6.39)$$

It can be seen that a sequentially consistent  $\mathcal{H}$  is also sequentially improving, since sequential consistency implies that  $H(i)$  is equal to one of the values  $H(j)$ ,  $j \in N(i)$ . We have the following generalization of Prop. 6.4.1, which also bears a relation with the general cost estimate for one-step lookahead policies of Prop. 6.3.1.

**Proposition 6.4.2:** Let the base heuristic  $\mathcal{H}$  be sequentially improving, and assume that the rollout algorithm  $\mathcal{RH}$  is terminating. Let  $(i_1, \dots, i_m)$  be the path generated by  $\mathcal{RH}$  starting from a non-destination node  $i_1$  and ending at a destination node  $i_m$ . Then the

cost of  $\mathcal{RH}$  starting from  $i_1$  is less or equal to the cost of  $\mathcal{H}$  starting from  $i_1$ . In particular, we have for all  $m = 1, \dots, \tilde{m}$ ,

$$H(i_m) = \min \left\{ H(i_1), \min_{j \in N(i_1)} H(j), \dots, \min_{j \in N(i_{m-1})} H(j) \right\}. \quad (6.40)$$

**Proof:** For each  $m = 1, \dots, \tilde{m} - 1$ , we have

$$H(i_m) \geq \min_{j \in N(i_m)} H(j),$$

by the sequential improvement assumption, while we have

$$\min_{j \in N(i_m)} H(j) = H(i_{m+1}),$$

by the definition of the rollout algorithm. These two relations imply Eq. (6.40). Since the cost of  $\mathcal{RH}$  starting from  $i_1$  is  $H(i_{\tilde{m}})$ , the result follows. **Q.E.D.**

#### Example 6.4.6 (One-Dimensional Walk -- Continued)

Consider the one-dimensional walk problem of Example 6.4.5, and let  $\mathcal{H}$  be defined as the algorithm that, starting at a node  $(k, m)$ , compares the cost  $g(m + N - k)$  (corresponding to taking all of the remaining  $N - k$  steps to the right) and the cost  $g(m - N + k)$  (corresponding to taking all of the remaining  $N - k$  steps to the left), and accordingly moves to node

$$(N, m + N - k) \quad \text{if} \quad g(m + N - k) \leq g(m - N + k),$$

or to node

$$(N, m - N + k) \quad \text{if} \quad g(m - N + k) < g(m + N - k).$$

It can be seen that  $\mathcal{H}$  is not sequentially consistent, but is instead sequentially improving. Using Eq. (6.40), it follows that starting from the origin  $(0, 0)$ ,  $\mathcal{RH}$  obtains the global minimum of  $g$  in the interval  $[-N, N]$ , while  $\mathcal{H}$  obtains the better of the two points  $-N$  and  $N$ .

Proposition 6.4.2 actually follows from a general equation for the cost of the path generated by the rollout algorithm, which holds for any base heuristic (not necessarily one that is sequentially improving). This is given in the following proposition, which is related to Prop. 6.3.2.

**Proposition 6.4.3:** Assume that the rollout algorithm  $\mathcal{RH}$  is terminating. Let  $(i_1, \dots, i_{\tilde{m}})$  be the path generated by  $\mathcal{RH}$  starting from a non-destination node  $i_1$  and ending at a destination node  $i_{\tilde{m}}$ . Then the cost of  $\mathcal{RH}$  starting from  $i_1$  is equal to

$$H(i_1) + \delta_{i_1} + \dots + \delta_{i_{\tilde{m}-1}},$$

where for every non-destination node  $i$ , we denote

$$\delta_i = \min_{j \in N(i)} H(j) - H(i).$$

**Proof:** We have by the definition of the rollout algorithm

$$H(i_m) + \delta_{i_m} = \min_{j \in N(i_m)} H(j) = H(i_{m+1}), \quad m = 1, \dots, \tilde{m} - 1.$$

By adding these equations over  $m$ , we obtain

$$H(i_1) + \delta_{i_1} + \dots + \delta_{i_{\tilde{m}-1}} = H(i_{\tilde{m}}).$$

Since the cost of  $\mathcal{RH}$  starting from  $i_1$  is  $H(i_{\tilde{m}})$ , the result follows. **Q.E.D.**

If the base heuristic is sequentially improving, we have  $\delta_i \leq 0$  for all non-destination nodes  $i$ , so it follows from Prop. 6.4.3 that the cost of the rollout algorithm is less or equal to the cost of the base heuristic (cf. Prop. 6.4.2).

#### The Fortified Rollout Algorithm

We now describe a variant of the rollout algorithm that implicitly uses a sequentially improving base heuristic, so that it has the cost improvement property of Prop. 6.4.2. This variant, called the *fortified rollout algorithm*, and denoted by  $\mathcal{RH}$ , starts at the origin  $s$ , and after  $m$  steps, maintains, in addition to the current sequence of nodes  $(s, i_1, \dots, i_m)$ , a path

$$P(i_m) = (i_m, i'_{m+1}, \dots, i'_k),$$

ending at a destination  $i'_k$ . Roughly speaking, the path  $P(i_m)$  is the tail portion of the best path found after the first  $m$  steps of the algorithm, in the sense that the destination  $i'_k$  has minimal cost over all the projections of nodes calculated thus far.

In particular, initially  $P(s)$  is the path generated by the base heuristic  $\mathcal{H}$  starting from  $s$ . At the typical step of the fortified rollout algorithm  $\mathcal{RH}$ ,

we have a node sequence  $(s, i_1, \dots, i_m)$ , where  $i_m$  is not a destination, and the path  $P(i_m) = (i_m, i'_{m-1}, \dots, i'_k)$ . Then, if

$$\min_{j \in N(i_m)} H(j) < g(i'_k), \quad (6.41)$$

$\mathcal{RH}$  adds to the node sequence  $(s, i_1, \dots, i_m)$  the node

$$i_{m+1} = \arg \min_{j \in N(i_m)} H(j),$$

and sets  $P(i_{m+1})$  to the path generated by  $\mathcal{H}$ , starting from  $i_{m+1}$ . On the other hand, if

$$\min_{j \in N(i_m)} H(j) \geq g(i'_k), \quad (6.42)$$

$\mathcal{RH}$  adds to the node sequence  $(s, i_1, \dots, i_m)$  the node

$$i_{m+1} = i'_{m+1},$$

and sets  $P(i_{m+1})$  to the path  $(i_{m+1}, i'_{m-2}, \dots, i'_k)$ . If  $i_{m+1}$  is a destination,  $\mathcal{RH}$  terminates, and otherwise  $\mathcal{RH}$  repeats the process with  $(s, i_1, \dots, i_{m+1})$  replacing  $(s, i_1, \dots, i_m)$ , and  $P(i_{m+1})$  replacing  $P(i_m)$ , respectively.

The idea behind the construction of  $\mathcal{RH}$  is to follow the path  $P(i_m)$  unless a path of lower cost is discovered through Eq. (6.41). We can show that  $\mathcal{RH}$  may be viewed as the rollout algorithm  $\mathcal{R}\bar{\mathcal{H}}$  corresponding to a modified version of  $\mathcal{H}$ , called *fortified*  $\mathcal{H}$ , and denoted  $\bar{\mathcal{H}}$ . This algorithm is applied to a slightly modified version of the original problem, which involves an additional downstream neighbor for each node  $i_m$  that is generated in the course of the algorithm  $\mathcal{RH}$  and for which the condition (6.42) holds. For every such node  $i_m$ , the additional neighbor is a copy of  $i'_{m+1}$ , and the path generated by  $\bar{\mathcal{H}}$  starting from this copy is  $(i'_{m+1}, \dots, i'_k)$ . From every other node, the path generated by  $\bar{\mathcal{H}}$  is the same as the path generated by  $\mathcal{H}$ .

It can be seen that  $\bar{\mathcal{H}}$  is sequentially improving, so that  $\mathcal{RH}$  is terminating and has the automatic cost sorting property of Prop. 6.4.2; that is,

$$H(i_m) = \min \left\{ H(i_1), \min_{j \in N(i_1)} H(j), \dots, \min_{j \in N(i_{m-1})} H(j) \right\}.$$

The above property can also be easily verified directly, using the definition of  $\mathcal{RH}$ . Finally, it can be seen that when  $\mathcal{H}$  is sequentially consistent, the rollout algorithm  $\mathcal{RH}$  and its fortified version  $\mathcal{RH}$  coincide.

### Using Multiple Path Construction Algorithms

In many problems, several promising path construction heuristics may be available. It is then possible to use all of these heuristics in the rollout framework. In particular, let us assume that we have  $K$  algorithms  $\mathcal{H}_1, \dots, \mathcal{H}_K$ . The  $k$ th of these algorithms, given a non-destination node  $i$ , produces a path  $(i, i_1, \dots, i_m, \bar{i})$  that ends at a destination node  $\bar{i}$ , and the corresponding cost is denoted by  $H_k(i) = g(\bar{i})$ . We can incorporate the  $K$  algorithms in a generalized version of the rollout algorithm, which uses the minimal cost

$$H(i) = \min_{k=1, \dots, K} H_k(i), \quad (6.43)$$

in place of the cost obtained by any one of the  $K$  algorithms  $\mathcal{H}_1, \dots, \mathcal{H}_K$ .

In particular, the algorithm starts with the origin node  $s$ . At the typical step, given a node sequence  $(s, i_1, \dots, i_m)$ , where  $i_m$  is not a destination, the algorithm adds to the sequence a node  $i_{m+1}$  such that

$$i_{m+1} = \arg \min_{j \in N(i_m)} H(j).$$

If  $i_{m+1}$  is a destination node, the algorithm terminates, and otherwise the process is repeated with the sequence  $(s, i_1, \dots, i_m, i_{m+1})$  replacing  $(s, i_1, \dots, i_m)$ .

An interesting property, which can be readily verified by using the definitions, is that if all the algorithms  $\mathcal{H}_1, \dots, \mathcal{H}_K$  are sequentially improving, the same is true for  $\mathcal{H}$ . This is consistent with the analysis of Example 6.3.2.

The fortified version of the rollout algorithm  $\mathcal{RH}$  easily generalizes for the case of Eq. (6.43), by defining the path generated starting from a node  $i$  as the path generated by the path construction algorithm, which attains the minimum in Eq. (6.43).

In an alternative version of the rollout algorithm that uses multiple path construction heuristics, the results of the  $K$  algorithms  $\mathcal{H}_1, \dots, \mathcal{H}_K$  are weighted with some fixed scalar weights  $r_k$  to compute  $H(i)$  for use in Eq. (6.33):

$$H(i) = \sum_{k=1}^K r_k H_k(i). \quad (6.44)$$

The weights  $r_k$  may be adjusted by trial and error. An alternative and more sophisticated possibility, is to use weights that depend on the node  $i$  and which are obtained by training using the neuro-dynamic programming methodology described in Vol. II.

### Extension for Intermediate Arc Costs

Let us consider a variant of the graph search problem where in addition to the terminal cost  $g(i)$ , there is a cost  $c(i, j)$  for a path to traverse an arc

$(i, j)$ . Within this context, the cost of a path  $(i_1, i_2, \dots, i_n)$  that starts at  $i_1$  and ends at a destination node  $i_n$  is redefined to be

$$g(i_n) + \sum_{k=1}^{n-1} c(i_k, i_{k+1}). \quad (6.45)$$

Note that when the cost  $g(i)$  is zero for all destination nodes  $i$ , this is the problem of finding a shortest path from the origin node  $s$  to one of the destination nodes, with  $c(i, j)$  viewed as the length of arc  $(i, j)$ . We have seen in Chapter 2 that there are efficient algorithms for solving this problem. However, here we are interested in problems where the number of nodes is very large, and the use of the shortest path algorithms of Chapter 2 is impractical.

One way to transform the problem with arc costs into one involving a terminal cost only is to redefine the graph of the problem so that nodes correspond to sequences of nodes in the original problem graph. Thus if we have arrived at node  $i_k$  using path  $(i_1, \dots, i_k)$ , the choice of  $i_{k+1}$  as the next node is viewed as a transition from  $(i_1, \dots, i_k)$  to  $(i_1, \dots, i_k, i_{k+1})$ . Both nodes  $(i_1, \dots, i_k)$  and  $(i_1, \dots, i_k, i_{k+1})$  are viewed as nodes of a redefined graph. Furthermore, in this redefined graph, a destination node has the form  $(i_1, i_2, \dots, i_n)$ , where  $i_n$  is a destination node of the original graph, and has a cost given by Eq. (6.45).

After the details are worked out, we see that to recover our earlier algorithms and analysis, we need to modify the cost of the heuristic algorithm  $\mathcal{H}$  as follows: If the path  $(i_1, \dots, i_n)$  is generated by  $\mathcal{H}$  starting at  $i_1$ , then

$$H(i_1) = g(i_n) + \sum_{k=1}^{n-1} c(i_k, i_{k+1}).$$

Furthermore, the rollout algorithm  $\mathcal{RH}$  at node  $i_m$  selects as next node  $i_{m+1}$  the node

$$i_{m+1} = \arg \min_{j \in N(i_m)} [c(i_m, j) + H(j)];$$

[cf. Eq. (6.33)]. The definition of a sequentially consistent algorithm remains unchanged. Furthermore, Prop. 6.4.1 remains unchanged except that Eqs. (6.36) and (6.37) are modified to read

$$H(i_k) \geq c(i_k, i_{k+1}) + H(i_{k+1}) = \min_{j \in N(i_k)} [c(i_k, j) + H(j)], \quad k = 1, \dots, m-1.$$

A sequentially improving algorithm should now be characterized by the property

$$H(i_k) \geq c(i_k, i_{k-1}) + H(i_{k+1}),$$

where  $i_{k+1}$  is the next node on the path generated by  $\mathcal{H}$  starting from  $i_k$ . Furthermore, Prop. 6.4.2 remains unchanged, except that Eq. (6.40) is modified to read

$$H(i_k) \geq \min_{j \in N(i_k)} [c(i_k, j) + H(j)], \quad k = 1, \dots, m-1.$$

Finally, the criterion  $\min_{j \in N(i_m)} H(j) < g(i'_k)$  [cf. Eq. (6.41)] used in the fortified rollout algorithm, given the sequence  $(s, i_1, \dots, i_m)$ , where  $i_m \notin \bar{\mathcal{N}}$ , and the path  $P(i_m) = (i_m, i'_{m+1}, \dots, i'_k)$ , should be replaced by

$$\min_{j \in N(i_m)} [c(i_m, j) + H(j)] < g(i'_k) + c(i_m, i'_{m+1}) + \sum_{l=m+1}^{k-1} c(i'_l, i'_{l+1}).$$

### Rollout Algorithms with Multistep Lookahead

We may incorporate *multistep lookahead* into the rollout framework. To describe the case of 2-step lookahead, suppose that after  $m$  steps of the rollout algorithm, we have the current node sequence  $(s, i_1, \dots, i_m)$ . We then consider the set of all 2-step-ahead neighbors of  $i_m$ , defined as

$$N_2(i_m) = \{j \in \mathcal{N} \mid j \in N(i_m) \text{ and } j \in \bar{\mathcal{N}}, \text{ or } j \in N(n) \text{ for some } n \in N(i_m)\}.$$

We run the base heuristic  $\mathcal{H}$  starting from each  $j \in N_2(i_m)$  and we find the node  $\bar{j} \in N_2(i_m)$  that has projection of minimum cost. Let  $i_{m+1} \in N(i_m)$  be the node next to  $i_m$  on the (one- or two-arc) path from  $i_m$  to  $\bar{j}$ . If  $i_{m+1}$  is a destination node, the algorithm terminates. Otherwise, the process is repeated with the sequence  $(s, i_1, \dots, i_m, i_{m+1})$  replacing  $(s, i_1, \dots, i_m)$ .

Note that a fortified version of the rollout algorithm described above is possible along the lines described earlier. Also, it is possible to eliminate from the set  $N_2(i_m)$  some of the 2-step neighbors of  $i_m$  that are judged less promising according to some heuristic criterion, in order to limit the number of applications of the base heuristic. This may be viewed as *selective depth lookahead*. Finally, the extension of the algorithm to lookahead of more than two steps is straightforward: we simply replace the 2-step-ahead neighbor set  $N_2(i_m)$  with a suitably defined  $k$ -step ahead neighbor set  $N_k(i_m)$ .

### Interpretation in Terms of DP

Let us now reinterpret the graph-based rollout algorithm within the context of deterministic DP. We will aim to view the base heuristic as a suboptimal

policy, and to view the rollout algorithm as a policy obtained by a process of policy improvement, provided the base heuristic is sequentially consistent.

To this end, we cast the graph search problem as a sequential decision problem, where each node corresponds to a state of a dynamic system. At each non-destination node/state  $i$ , a node  $j$  must be selected from the set of neighbors  $N(i)$ ; then if  $j$  is a destination, the process terminates with cost  $g(j)$ , and otherwise the process is repeated with  $j$  becoming the new state. The DP algorithm calculates for every node  $i$ , the minimal cost that can be achieved starting from  $i$ , that is, the smallest value of  $g(i)$  that can be obtained using paths that start from  $i$  and end at destination nodes  $\bar{i}$ . This value, denoted  $J^*(i)$ , is the optimal cost-to-go starting at node  $i$ . Once  $J^*(i)$  is computed for all nodes  $i$ , an optimal path  $(i_1, i_2, \dots, i_m)$  can be constructed starting from any initial node/state  $i_1$  by successively generating nodes using the relation

$$i_{k+1} = \arg \min_{j \in N(i_k)} J^*(j), \quad k = 1, \dots, m-1, \quad (6.46)$$

up to the point where a destination node  $i_m$  is encountered.<sup>†</sup>

A base heuristic  $\mathcal{H}$  defines a policy  $\pi$ , i.e., an assignment of a successor node to any non-destination node. However, starting from a given node  $i$ , the cost of  $\pi$  need not be equal to  $H(i)$  because if a path  $(i_1, i_2, i_3, \dots, i_m)$  is generated by  $\mathcal{H}$  starting from node  $i_1$ , it is not necessarily true that the path  $(i_2, i_3, \dots, i_m)$  is generated by the base heuristic starting from  $i_2$ . Thus the successor node chosen at node  $i_2$  by policy  $\pi$  may be different than the one used in the calculation of  $H(i_1)$ . On the other hand, if  $\mathcal{H}$  is sequentially consistent, the cost of policy  $\pi$  starting from a node  $i$  is  $H(i)$ , since sequential consistency implies that the path that the base heuristic generates starting at the successor node is part of the path it generates at the predecessor node. Thus the cost improvement property of the rollout algorithm in the sequentially consistent case also follows from the cost improvement property shown earlier in the DP context.

Generally, we can view the rollout algorithm  $\mathcal{RH}$  as a one-step lookahead policy that uses  $H(j)$  as a cost-to-go approximation from state  $j$ . In some cases,  $H(j)$  is the cost of some policy (in the DP sense), such as for example when  $\mathcal{H}$  is sequentially consistent, as explained above. In general, however, this need not be so, in which case we can view  $H(j)$  as a convenient cost-to-go approximation that is derived from the base heuristic. Still, the rollout algorithm  $\mathcal{RH}$  may improve on the cost of the base heuristic (e.g., when  $\mathcal{H}$  is sequentially improving, cf. Prop. 6.4.2) just as a general one-step lookahead policy may improve on the corresponding one-step lookahead cost approximation (cf. Prop. 6.3.1).

<sup>†</sup> We assume here that there are no termination/cycling difficulties of the type illustrated in the footnote following Example 6.4.4.

#### 6.4.2 $Q$ -Factors Evaluated by Simulation

We now consider a stochastic problem and some computational issues regarding the implementation of the rollout policy based on a given heuristic policy. A conceptually straightforward approach to compute the rollout control at a given state  $x_k$  and time  $k$  is to use Monte-Carlo simulation. To implement this algorithm, we consider all possible controls  $u_k \in U_k(x_k)$  and we generate a “large” number of simulated trajectories of the system starting from  $x_k$ , using  $u_k$  as the first control, and using the policy  $\pi$  thereafter. Thus a simulated trajectory has the form

$$x_{i-1} = f_i(x_i, \mu_i(x_i), w_i), \quad i = k+1, \dots, N-1,$$

where the first generated state is

$$x_{k+1} = f_k(x_k, u_k, w_k),$$

and each of the disturbances  $w_k, \dots, w_{N-1}$  is an independent random sample from the given distribution. The costs corresponding to these trajectories are averaged to compute an approximation  $\tilde{Q}_k(x_k, u_k)$  to the  $Q$ -factor

$$E\left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}.$$

Here,  $\tilde{Q}_k(x_k, u_k)$  is an approximation to  $Q_k(x_k, u_k)$  because of the simulation error resulting from the use of a limited number of trajectories. The approximation becomes increasingly accurate as the number of simulated trajectories increases. Once the approximate  $Q$ -factor  $\tilde{Q}_k(x_k, u_k)$  corresponding to each control  $u \in U_k(x_k)$  is computed, we can obtain the (approximate) rollout control  $\bar{\mu}_k(x_k)$  by the minimization

$$\bar{\mu}_k(x_k) = \arg \min_{u_k \in U_k(x_k)} \tilde{Q}_k(x_k, u_k).$$

There is a serious flaw with this approach, due to the simulation error involved in the calculation of the  $Q$ -factors. In particular, for the calculation of  $\bar{\mu}_k(x_k)$  to be accurate, the  $Q$ -factor differences

$$Q_k(x_k, u_k) - Q_k(x_k, \hat{u}_k)$$

must be computed accurately for all pairs of controls  $u_k$  and  $\hat{u}_k$ , so that these controls can be accurately compared. On the other hand, the simulation/approximation errors in the computation of the individual  $Q$ -factors  $Q_k(x_k, u_k)$  may be magnified through the preceding differencing operation.

An alternative approach is to approximate by simulation the  $Q$ -factor difference  $Q_k(x_k, u_k) - Q_k(x_k, \hat{u}_k)$  by sampling the difference

$$C_k(x_k, u_k, w_k) - C_k(x_k, \hat{u}_k, w_k),$$

where  $\mathbf{w}_k = (w_k, w_{k+1}, \dots, w_{N-1})$  and

$$C_k(x_k, u_k, \mathbf{w}_k) = g_N(x_N) + g_k(x_k, u_k, w_k) + \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i), w_i).$$

This approximation may be far more accurate than the one obtained by differencing independent samples of  $C_k(x_k, u_k, \mathbf{w}_k)$  and  $C_k(x_k, \hat{u}_k, \mathbf{w}_k)$ . Indeed, by introducing the zero mean sample errors

$$D_k(x_k, u_k, \mathbf{w}_k) = C_k(x_k, u_k, \mathbf{w}_k) - Q_k(x_k, u_k),$$

it can be seen that the variance of the error in estimating  $Q_k(x_k, u_k) - Q_k(x_k, \hat{u}_k)$  with the former method will be smaller than with the latter method if and only if

$$\begin{aligned} E_{\mathbf{w}_k, \hat{\mathbf{w}}_k} \{ |D_k(x_k, u_k, \mathbf{w}_k) - D_k(x_k, \hat{u}_k, \hat{\mathbf{w}}_k)|^2 \} \\ > E_{\mathbf{w}_k} \{ |D_k(x_k, u_k, \mathbf{w}_k) - D_k(x_k, \hat{u}_k, \mathbf{w}_k)|^2 \}, \end{aligned}$$

or equivalently

$$E \{ D_k(x_k, u_k, \mathbf{w}_k) D_k(x_k, \hat{u}_k, \mathbf{w}_k) \} > 0; \quad (6.47)$$

i.e., if and only if the correlation between the errors  $D_k(x_k, u_k, \mathbf{w}_k)$  and  $D_k(x_k, \hat{u}_k, \mathbf{w}_k)$  is positive. A little thought should convince the reader that this property is likely to hold in many types of problems. Roughly speaking, the relation (6.47) holds if changes in the value of  $u_k$  (at the first stage) have little effect on the value of the error  $D_k(x_k, u_k, \mathbf{w}_k)$  relative to the effect induced by the randomness of  $\mathbf{w}_k$ . In particular, suppose that there exists a scalar  $\gamma < 1$  such that, for all  $x_k, u_k$ , and  $\hat{u}_k$ , there holds

$$E \{ |D_k(x_k, u_k, \mathbf{w}_k) - D_k(x_k, \hat{u}_k, \mathbf{w}_k)|^2 \} \leq \gamma E \{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \}. \quad (6.48)$$

Then we have

$$\begin{aligned} D_k(x_k, u_k, \mathbf{w}_k) D_k(x_k, \hat{u}_k, \mathbf{w}_k) \\ = |D_k(x_k, u_k, \mathbf{w}_k)|^2 \\ + D_k(x_k, u_k, \mathbf{w}_k) (D_k(x_k, \hat{u}_k, \mathbf{w}_k) - D_k(x_k, u_k, \mathbf{w}_k)) \\ \geq |D_k(x_k, u_k, \mathbf{w}_k)|^2 \\ - |D_k(x_k, u_k, \mathbf{w}_k)| \cdot |D_k(x_k, \hat{u}_k, \mathbf{w}_k) - D_k(x_k, u_k, \mathbf{w}_k)|, \end{aligned}$$

from which we obtain, using also Eq. (6.48),

$$\begin{aligned} E \{ D_k(x_k, u_k, \mathbf{w}_k) D_k(x_k, \hat{u}_k, \mathbf{w}_k) \} \\ \geq E \left\{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\} \\ - E \left\{ |D_k(x_k, u_k, \mathbf{w}_k) - D_k(x_k, \hat{u}_k, \mathbf{w}_k)|^2 \right\} \\ \geq E \left\{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\} - \frac{1}{2} E \left\{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\} \\ - \frac{1}{2} E \left\{ |D_k(x_k, \hat{u}_k, \mathbf{w}_k) - D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\} \\ \geq \frac{1-\gamma}{2} E \left\{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\}. \end{aligned}$$

Thus, under the assumption (6.48) and the assumption

$$E \left\{ |D_k(x_k, u_k, \mathbf{w}_k)|^2 \right\} > 0,$$

the condition (6.47) holds and guarantees that by averaging cost difference samples rather than differencing (independently obtained) averages of cost samples, the simulation error variance decreases.

#### 6.4.3 Q-Factor Approximation

Let us now consider the case of a stochastic problem and various possibilities for approximating the costs-to-go  $H_k(x_k)$ ,  $k = 1, \dots, N-1$ , of the base policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , rather than calculating them by Monte-Carlo simulation. For example, in a certainty equivalence approach, given a state  $x_k$  at time  $k$ , we fix the remaining disturbances at some "typical" values  $\bar{w}_{k+1}, \dots, \bar{w}_{N-1}$ , and we approximate the true  $Q$ -factor

$$Q_k(x_k, u_k) = E \{ g_k(x_k, u_k, w_k) + H_{k+1}(f_k(x_k, u_k, w_k)) \}$$

with

$$\tilde{Q}_k(x_k, u_k) = E \{ g_k(x_k, u_k, w_k) + \tilde{H}_{k+1}(f_k(x_k, u_k, w_k)) \}, \quad (6.49)$$

where  $\tilde{H}_{k+1}(f_k(x_k, u_k, w_k))$  is obtained by

$$\tilde{H}_{k+1}(f_k(x_k, u_k, w_k)) = g_N(\bar{x}_N) + \sum_{i=k+1}^{N-1} g_i(\bar{x}_i, \mu_i(x_i), \bar{w}_i),$$

the initial state is

$$\bar{x}_{k+1} = f_k(x_k, u_k, w_k),$$

and the intermediate states are given by

$$\bar{x}_{i+1} = f_i(\bar{x}_i, \mu_i(x_i), \bar{w}_i), \quad i = k+1, \dots, N-1.$$

Thus, in this approach, the rollout control is approximated by

$$\tilde{\mu}_k(x_k) = \arg \min_{u_k \in U_k(x_k)} \tilde{Q}_k(x_k, u_k).$$

Note that the approximate cost-to-go  $\tilde{H}_{k+1}(x_{k+1})$  represents an approximation of the true cost-to-go  $H_{k+1}(x_{k+1})$  of the base policy based on a single sample (the nominal disturbances  $\bar{w}_{k+1}, \dots, \bar{w}_{N-1}$ ). A potentially more accurate approximation is obtained using multiple nominal disturbance sequences and averaging the corresponding costs with appropriate nominal probabilities, similar to the scenario approximation approach of Example 6.3.6.

Let us also mention another approach for approximation of the cost-to-go  $H_{k+1}$  of the base policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , using an approximation architecture. Here we calculate (possibly approximate) values for the cost-to-go of the base policy at a finite set of state-time pairs, and then we select the weights through a “least-squares fit” of these values.

In particular, suppose that we have calculated the correct value of the cost-to-go  $H_{N-1}(x^i)$  at the next-to-last stage for  $s$  states  $x^i$ ,  $i = 1, \dots, s$ , through the DP formula

$$\begin{aligned} H_{N-1}(x_{N-1}) &= E \left\{ g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) \right. \\ &\quad \left. + g_N(f_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1})) \right\}, \end{aligned}$$

and the given terminal cost function  $g_N$ . We can then approximate the entire function  $H_{N-1}(x_{N-1})$  by a function of some given form

$$\tilde{H}_{N-1}(x_{N-1}, r_{N-1}),$$

where  $r_{N-1}$  is a vector of weights, which can be obtained by solving the problem

$$\min_r \sum_{i=1}^s |H_{N-1}(x^i) - \tilde{H}_{N-1}(x^i, r)|^2. \quad (6.50)$$

For example if  $\tilde{H}_{N-1}$  is specified to be a linear function of  $m$  features  $y_1(x), \dots, y_m(x)$ ,

$$\tilde{H}_{N-1}(x, r) = \sum_{j=1}^m r_j y_j(x),$$

the least squares problem (6.50) is

$$\min_r \sum_{i=1}^s |H_{N-1}(x^i) - \sum_{j=1}^m r_j y_j(x^i)|^2.$$

This is a linear least squares problem that can be solved in closed form (its cost function is convex quadratic in the vector  $r$ ).

Note that this approximation procedure can be enhanced if we have additional information on the true cost-to-go function  $H_{N-1}(x_{N-1})$ . For example, if we know that  $H_{N-1}(x_{N-1}) \geq 0$  for all  $x_{N-1}$ , we may first compute the approximation  $\tilde{H}_{N-1}(x_{N-1}, r_{N-1})$  by solving the least-squares problem (6.50) and then replace this approximation by

$$\max \{0, \tilde{H}_{N-1}(x_{N-1}, r_{N-1})\}.$$

Once an approximating function  $\tilde{H}_{N-1}(x_{N-1}, r_{N-1})$  for the next-to-last stage has been obtained, it can be used to similarly obtain an approximating function  $\tilde{H}_{N-2}(x_{N-2}, r_{N-2})$ . In particular, (approximate) cost-to-go function values  $\tilde{H}_{N-2}(x^i)$  are obtained for  $s$  states  $x^i$ ,  $i = 1, \dots, s$ , through the (approximate) DP formula

$$\begin{aligned} \hat{H}_{N-2}(x_{N-2}) &= E \left\{ g_{N-2}(x_{N-2}, \mu_{N-2}(x_{N-2}), w_{N-2}) \right. \\ &\quad \left. + \tilde{H}_{N-1}(f_{N-2}(x_{N-2}, \mu_{N-2}(x_{N-2}), w_{N-2}), r_{N-1}) \right\}. \end{aligned}$$

These values are used to approximate the cost-to-go function  $H_{N-2}(x_{N-2})$  by a function of some given form

$$\tilde{H}_{N-2}(x_{N-2}, r_{N-2}),$$

where  $r_{N-2}$  is a vector of parameters, which is obtained by solving the problem

$$\min_r \sum_{i=1}^s |\hat{H}_{N-2}(x^i) - \tilde{H}_{N-2}(x^i, r)|^2.$$

The process can be similarly continued to obtain  $\tilde{H}_k(x_k, r_k)$  up to  $k = 0$  by solving for each  $k$  the problem

$$\min_r \sum_{i=1}^s |\hat{H}_k(x^i) - \tilde{H}_k(x^i, r)|^2. \quad (6.51)$$

Given the approximations  $\tilde{H}_0(x_0, r_0), \dots, \tilde{H}_{N-1}(x_{N-1}, r_{N-1})$  to the cost-to-go of the base policy, one may obtain a suboptimal policy by using at state-time pair  $(x_k, k)$  the one-step lookahead control

$$\bar{\mu}_k(x_k) = \arg \min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{H}_{k+1}(f_k(x_k, u_k, w_k), r_{k+1}) \right\}.$$

This control must be calculated on-line once the state  $x_k$  at time  $k$  becomes known.

## 6.5 MODEL PREDICTIVE CONTROL AND RELATED METHODS

In many control problems where the objective is to keep the state of a system near some desired point, the linear-quadratic models of Sections 4.1 and 5.3 are not satisfactory. There are two main reasons for this:

- (a) The system may be nonlinear, and using for control purposes a model that is linearized around the desired point may be inappropriate.
- (b) There may be control and/or state constraints, which are not handled adequately through a quadratic penalty on state and control. The reason may be special structure of the problem dictating that, for efficiency purposes, the system should often be operated at the boundary of its constraints. The solution obtained from a linear-quadratic model is not suitable for this, because the quadratic penalty on state and control tends to “blur” the boundaries of the constraints.

These inadequacies of the linear-quadratic model have motivated a form of suboptimal control, called *model predictive control* (MPC), which combines elements of several ideas that we have discussed so far: certainty equivalent control, multistage lookahead, and rollout algorithms. We will focus primarily on the most common form of MPC, where the system is either deterministic, or else it is stochastic, but it is replaced with a deterministic version by using typical values in place of all uncertain quantities, as in the certainty equivalent control approach. At each stage, a (deterministic) optimal control problem is solved over a fixed length horizon, starting from the current state. The first component of the corresponding optimal policy is then used as the control of the current stage, while the remaining components are discarded. The process is then repeated at the next stage, once the next state is revealed. We will also briefly discuss a version of MPC where there is uncertainty with a set-membership description.

The primary objective in MPC, aside from fulfilling the state and control constraints of the problem, is to obtain a stable closed-loop system. Note here that we may only be able to guarantee the stability of the deterministic model that forms the basis for the calculations of the MPC. This is consistent with a common practice in control theory: design a stable controller for a deterministic model of the system, and expect that it will provide some form of stability in a realistic stochastic environment as well.

In Section 6.5.2, we will discuss the mechanism by which stability is achieved by MPC under some reasonable conditions. We will first discuss in the next subsection some issues of multistage lookahead, which are relevant to MPC, but are also important in a broader context. In Section 6.5.3, we provide a general unifying framework for suboptimal control, which includes as special cases several approaches discussed in this chapter, OLFC, rollout, and MPC, and captures the mathematical essence of their attractive properties.

### 6.5.1 Rolling Horizon Approximations

Let us consider the  $l$ -step lookahead policy when the cost-to-go approximation is just zero. With this policy, at each stage we apply a control that would be optimal if the remaining horizon length were  $l$  and there were no terminal cost. Thus at the typical stage  $k$  we ignore the costs incurred in stages  $k + l + 1$  and beyond, and accordingly we neglect the corresponding long-range effects of our action. We call this the *rolling horizon* approach. In a variant of this approach, following the  $l$  steps of lookahead, we use a cost-to-go approximation that is equal to the terminal cost function  $g_N$ . This is essential if  $g_N$  is significant relative to the costs per stage accumulated over  $l$  stages.

We may also use a rolling horizon approach for infinite horizon problems. Then the length of the horizon of the problem solved at each stage stays the same at all stages. As a result, for a time-invariant system and cost per stage, the rolling horizon approach produces a stationary policy (the controls applied at the same state but in different stages are the same). This is a generic characteristic of infinite horizon control, as we have seen in the context of linear-quadratic problems (see also the discussion of Vol. II).

Naturally, a policy obtained using a rolling horizon is typically not optimal. One is tempted to conjecture that if the size of the lookahead  $l$  is increased, then the performance of the rolling horizon policy is improved. This, however, need not be true as the following example shows.

#### Example 6.5.1

This is an oversimplified problem, which, however, demonstrates the basic pitfall of the rolling horizon approach.

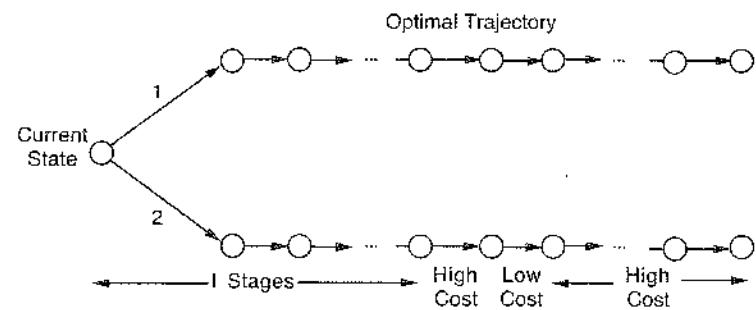


Figure 6.5.1 The problem of Example 6.5.1.

Consider a deterministic setting where at the initial state there are two possible controls, say 1 and 2 (see Fig. 6.5.1). At all other states there is only one available control, so a policy consists of just the initial choice between controls 1 and 2. Suppose that (based on the cost of the subsequent  $N$  stages) control 1 is optimal. Suppose also that if control 2 is chosen, an "unfavorable" (high cost) state results after  $l$  transitions, followed by a "particularly favorable" state, which is then followed by other "unfavorable" states. Then, in contrast with the  $l$ -step lookahead policy, the  $(l-1)$ -step lookahead policy may view the inferior control 2 as being better, because it may be "fooled" by the presence of the "particularly favorable" state  $l+1$  transitions ahead.

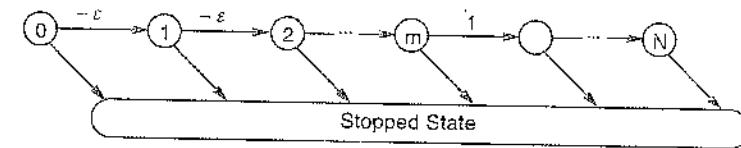
The rolling horizon approach is also interesting in the context of a rollout algorithm, where we need to calculate the cost-to-go of the base policy at various states. It is possible to use a rolling horizon approximation in the calculation of this cost-to-go. Thus, from the given state, we calculate the cost of the base policy over a fixed number of stages, rather than over the entire remaining horizon. This can result in significant computational savings. Furthermore, there may also be an *improvement* in the performance of the rollout policy if a rolling horizon approximation is used. One reason is the phenomenon illustrated in the preceding example. In fact, because of the suboptimality of the base policy, this phenomenon can get exaggerated, as shown in the following example.

### Example 6.5.2

Consider an  $N$ -stage stopping problem where at each stage we may either stop with a stopping cost equal to 0, or continue at a certain cost that is either  $-\epsilon$  or 1, where  $0 < \epsilon < 1/N$  (see Fig. 6.5.2). Let the first state with continuation cost equal to 1 be state  $m$ . Then the optimal policy is to stop after  $m$  steps at state  $m$ . The corresponding optimal cost is  $-me$ . It can also be seen that an  $l$ -step rolling horizon approach with the cost evaluated optimally over the  $l$  steps (rather than suboptimally using a base heuristic) is optimal.

Consider now the rollout policy where the base heuristic is to continue at every state (except the last where stopping is mandatory). It can be seen that this policy will stop at the initial state at a cost of 0, since it will evaluate the continuation action as having positive cost, in view of the fact  $1 - N\epsilon > 0$ , and will thus prefer the stopping action. However, the rollout policy that uses a rolling horizon of  $l$  stages, with  $l \leq m$ , will continue up to the first  $m - l + 1$  stages, thus compiling a cost of  $-(m - l + 1)\epsilon$ . Thus, as the length  $l$  of the rolling horizon becomes shorter, the performance of the rollout policy improves!

Another example of a rollout algorithm whose performance can be improved by using a rolling horizon approximation is the breakthrough problem of Example 6.4.2. In this case, the evolution of the system under



**Figure 6.5.2** The problem of Example 6.5.2

a rollout algorithm, where the greedy heuristic is evaluated using an  $l$ -step rolling horizon approximation, can be modeled using a Markov chain with  $l + 1$  states (see Exercise 6.18). Using this Markov chain, it is possible to ascertain that for a problem with a large number of steps  $N$ , the length of the rolling horizon that maximizes the breakthrough probability approaches an optimal value that is essentially independent of  $N$ .

### 6.5.2 Stability Issues in Model Predictive Control

As mentioned earlier, model predictive control (MPC) was initially motivated by the desire to introduce nonlinearities, and control and/or state constraints into the linear-quadratic framework, and obtain a suboptimal but stable closed-loop system. With this in mind, we will describe MPC for the case of a stationary, possibly nonlinear, deterministic system, where state and control belong to some Euclidean spaces. The system is

$$x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \dots$$

and the cost per stage is quadratic:

$$x'_k Q x_k + u'_k R u_k, \quad k = 0, 1, \dots$$

where  $Q$  and  $R$  are positive definite symmetric matrices. We impose state and control constraints

$$x_k \in X, \quad u_k \in U(x_k), \quad k = 0, 1, \dots$$

and we assume that the set  $X$  contains the origin of the corresponding Euclidean space. Furthermore, if the system is at the origin, it can be kept there at no cost with control equal to 0, i.e.,  $0 \in U(0)$  and  $f(0, 0) = 0$ . We want to derive a stationary feedback controller that applies control  $\bar{u}(x)$  at state  $x$ , and is such that, for all initial states  $x_0 \in X$ , the state of the closed-loop system

$$x_{k+1} = f(x_k, \bar{\mu}(x_k))$$

satisfies the state and control constraints, and the total cost over an infinite number of stages is finite:

$$\sum_{k=0}^{\infty} (x'_k Q x_k + \bar{\mu}(x_k)' R \bar{\mu}(x_k)) < \infty. \quad (6.52)$$

Note that because of the positive definiteness of  $Q$  and  $R$ , the feedback controller  $\bar{\mu}$  is stable in the sense that  $x_k \rightarrow 0$  and  $\bar{\mu}(x_k) \rightarrow 0$  for all initial states  $x_0 \in X$ . (In the case of a linear system, the assumption of positive definiteness of  $Q$  may be relaxed to positive semidefiniteness, together with an observability assumption of the type introduced in Section 4.1 and Prop. 4.1.1.)

In order for such a controller to exist, it is evidently sufficient [in view of the assumption that  $f(0, 0) = 0$ ] that there exists a positive integer  $m$  such that for every initial state  $x_0 \in X$ , one can find a sequence of controls  $u_k$ ,  $k = 0, 1, \dots, m-1$ , which drive to 0 the state  $x_m$  of the system at time  $m$ , while keeping all the preceding states  $x_1, x_2, \dots, x_{m-1}$  within  $X$  and satisfying the control constraints  $u_0 \in U(x_0), \dots, u_{m-1} \in U(x_{m-1})$ . We refer to this as the *constrained controllability assumption* (cf. the corresponding assumption of Section 4.1). In practical applications, this assumption can often be checked easily. Alternatively, the state and control constraints can be constructed in a way that the assumption is satisfied; the methodology of reachability of target tubes, discussed in Section 4.6.2, can be used for this purpose.

Let us now describe a form of MPC under the preceding assumption. At each stage  $k$  and state  $x_k \in X$ , it solves an  $m$ -stage deterministic optimal control problem involving the same quadratic cost and the requirement that the state after  $m$  stages be exactly equal to 0. This is the problem of minimizing

$$\sum_{i=k}^{k+m-1} (x_i' Q x_i + u_i' R u_i),$$

subject to the system equation constraints

$$x_{i+1} = f(x_i, u_i), \quad i = k, k+1, \dots, k+m-1,$$

the state and control constraints

$$x_i \in X, \quad u_i \in U(x_i), \quad i = k, k+1, \dots, k+m-1,$$

and the terminal state constraint

$$x_{k+m} = 0.$$

By the constrained controllability assumption, this problem has a feasible solution. Let  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{k+m-1}\}$  be a corresponding optimal control sequence. The MPC applies at stage  $k$  the first component of this sequence,

$$\bar{\mu}(x_k) = \bar{u}_k,$$

and discards the remaining components.

### Example 6.5.3

Consider the scalar linear system

$$x_{k+1} = x_k + u_k,$$

and the state and control constraints

$$x_k \in X = \{x \mid |x| \leq 1.5\}, \quad u_k \in U(x_k) = \{u \mid |u| \leq 1\}.$$

Let also  $Q = R = 1$ . We select  $m = 2$ . For this value of  $m$ , the constrained controllability assumption is satisfied.

When at state  $x_k \in X$ , the MPC minimizes the two-stage cost

$$x_k^2 + u_k^2 + (x_k + u_k)^2 + u_{k+1}^2,$$

subject to the control constraints

$$|u_k| \leq 1, \quad |u_{k+1}| \leq 1,$$

and the state constraints

$$x_{k+1} \in X, \quad x_{k+2} = x_k + u_k + u_{k+1} = 0.$$

It is easily verified that this minimization yields  $u_{k+1} = -(x_k + u_k)$  and  $u_k = -(2/3)x_k$ . Thus the MPC takes the form

$$\bar{\mu}(x_k) = -\frac{2}{3}x_k,$$

and the closed-loop system is

$$x_{k+1} = \frac{1}{3}x_k, \quad k = 0, 1, \dots$$

Note that while the closed-loop system is stable, its state is never driven to 0 if started from  $x_0 \neq 0$ .

We now show that the MPC satisfies the stability condition (6.52). Let  $x_0, u_0, x_1, u_1, \dots$  be the state and control sequence generated by MPC, so that

$$u_k = \bar{\mu}(x_k), \quad x_{k+1} = f(x_k, \bar{\mu}(x_k)), \quad k = 0, 1, \dots$$

Denote  $\hat{J}(x)$  the optimal cost of the  $m$ -stage problem solved by MPC when at a state  $x \in X$ . Let also  $\hat{J}(x)$  be the optimal cost starting at  $x$  of a corresponding  $(m-1)$ -stage problem, i.e., the optimal value of the quadratic cost

$$\sum_{k=0}^{m-2} (x_k' Q x_k + u_k' R u_k),$$

where  $x_0 = x$ , subject to the constraints

$$x_k \in \bar{X}, \quad u_k \in \bar{U}(x_k), \quad k = 0, 1, \dots, m - 2,$$

and

$$x_{m-1} = 0.$$

[For states  $x \in X$  for which this problem does not have a feasible solution, we write  $\tilde{J}(x) = \infty$ .] Since having one less stage in our disposal to drive the state to 0 cannot decrease the optimal cost, we have for all  $x \in X$

$$\hat{J}(x) \leq \tilde{J}(x). \quad (6.53)$$

From the definitions of  $\hat{J}$  and  $\tilde{J}$ , we have for all  $k$ ,

$$\min_{u \in U(x)} [x'_k Q x_k + u' R u + \hat{J}(f(x_k, u))] = x'_k Q x_k + u'_k R u_k + \hat{J}(x_{k+1}) = \hat{J}(x_k), \quad (6.54)$$

so using Eq. (6.53), we obtain

$$x'_k Q x_k + u'_k R u_k + \hat{J}(x_{k+1}) \leq \hat{J}(x_k), \quad k = 0, 1, \dots$$

Adding this equation for all  $k$  in a range  $[0, K]$ , where  $K = 0, 1, \dots$ , we obtain

$$\hat{J}(x_{K+1}) + \sum_{k=0}^K (x'_k Q x_k + u'_k R u_k) \leq \hat{J}(x_0).$$

Since  $\hat{J}(x_{K+1}) \geq 0$ , it follows that

$$\sum_{k=0}^K (x'_k Q x_k + u'_k R u_k) \leq \hat{J}(x_0), \quad K = 0, 1, \dots, \quad (6.55)$$

and taking the limit as  $K \rightarrow \infty$ ,

$$\sum_{k=0}^{\infty} (x'_k Q x_k + u'_k R u_k) \leq \hat{J}(x_0) < \infty.$$

This shows the stability condition (6.52).

We note that the one-step lookahead function  $\hat{J}$  implicitly used by MPC [cf. Eq. (6.54)] is the cost-to-go function of a certain policy. This is the policy that drives to 0 the state after  $m - 1$  stages and keeps the state at 0 thereafter, while observing the state and control constraints  $x_k \in \bar{X}$  and  $u_k \in \bar{U}(x_k)$ , and minimizing the quadratic cost. Thus, we can also view MPC as a rollout algorithm with the policy just described viewed as the base heuristic. In fact the stability property of MPC is a special case of the cost improvement property of rollout algorithms, which in the

case of a quadratic cost, implies that if the base heuristic results in a stable closed-loop system, the same is true for the corresponding rollout algorithm.

Regarding the choice of the horizon length  $m$  for the MPC calculations, note that if the constrained controllability assumption is satisfied for some value of  $m$ , it is also satisfied for all larger values of  $m$ . Furthermore, it can be seen that the  $m$ -stage cost  $\hat{J}(x)$ , which by Eq. (6.55), is an upper bound to the cost of MPC, cannot increase with  $m$ . This argues for a larger value of  $m$ . On the other hand, the optimal control problem solved at each stage by the MPC becomes larger and hence more difficult as  $m$  increases. Thus, the horizon length is usually chosen on the basis of some experimentation: first use target tube reachability methods (cf. Section 4.6.2) to ensure that  $m$  is large enough for the constrained controllability assumption to hold with a target tube that is sufficiently large for the practical problem at hand, and then by further experimentation to ensure overall satisfactory performance.

The MPC scheme that we have described is just the starting point for a broad methodology with many variations, which often relate to the suboptimal control methods that we have discussed so far in this chapter. For example, in the problem solved by MPC at each stage, instead of the requirement of driving the system state to 0 in  $m$  steps, one may use a large penalty for the state being nonzero after  $m$  steps. Then, the preceding analysis goes through, as long as the terminal penalty is chosen so that Eq. (6.53) is satisfied. In another variant one may use a nonquadratic cost function, which is everywhere positive except at  $(x, u) = (0, 0)$ . In still another variant, instead of aiming to drive the state to 0 after  $m$  steps, one aims to reach a sufficiently small neighborhood of the origin, within which a stabilizing controller, designed by other methods, may be used. This variant is also well-suited for taking into account disturbances described by set membership, as we now proceed to explain.

### MPC with Set-Membership Disturbances

To extend the MPC methodology to the case where there are disturbances  $w_k$  in the system equation

$$x_{k+1} = f(x_k, u_k, w_k),$$

we must first modify the stability objective. The reason is that in the presence of disturbances, the stability condition (6.52) is impossible to meet. A reasonable alternative is to introduce a set-membership constraint  $w_k \in W(x_k, u_k)$  for the disturbance and a target set  $T$  for the state, and to require that the controller specified by MPC drives the state to  $T$  with finite quadratic cost.

To formulate the MPC, we assume that  $T \subset X$ , and that once the system state enters  $T$ , we will use some control law  $\bar{\mu}$  that keeps the state

within  $T$  for all possible values of the disturbances, i.e.,

$$f(x, \bar{\mu}(x), w) \in T, \quad \text{for all } x \in T, w \in W(x, \bar{\mu}(x)). \quad (6.56)$$

The detailed methodology by which such a target set  $T$  and control law  $\bar{\mu}$  are obtained is outside our scope. We refer to the discussion of reachability of target tubes in Section 4.6.2 for orientation into this problem and references; see also Exercise 4.31, and Exercises 3.21 and 3.22 of Vol. II. We view  $T$  essentially as a cost-free and absorbing state, similar to our view of the origin in the earlier deterministic context. Consistent with this interpretation, we introduce the stage cost function

$$g(x, u) = \begin{cases} x'Qx + u'Ru & \text{if } x \notin T, \\ 0 & \text{if } x \in T. \end{cases}$$

The MPC is now defined as follows: At each stage  $k$  and state  $x_k \in X$  with  $x_k \notin T$ , it solves the  $m$ -stage minimax control problem of finding a policy  $\hat{\mu}_k, \hat{\mu}_{k+1}, \dots, \hat{\mu}_{k+m-1}$  that minimizes

$$\max_{\substack{w_i \in W(x_i, \hat{\mu}(x_i)), \\ i=k, k+1, \dots, k+m-1}} \sum_{i=k}^{k+m-1} g(x_i, \hat{\mu}(x_i)),$$

subject to the system equation constraints

$$x_{i+1} = f(x_i, u_i, w_i), \quad i = k, k+1, \dots, k+m-1,$$

the control and state constraints

$$x_i \in X, \quad u_i \in U(x_i), \quad i = k, k+1, \dots, k+m-1,$$

and the terminal state constraint

$$x_i \in T, \quad \text{for some } i \in [k+1, k+m].$$

These constraints must be satisfied for all disturbance sequences satisfying

$$w_i \in W(x_i, \hat{\mu}(x_i)), \quad i = k, k+1, \dots, k+m-1.$$

The MPC applies at stage  $k$  the first component of the policy  $\hat{\mu}_k, \hat{\mu}_{k+1}, \dots, \hat{\mu}_{k+m-1}$  thus obtained,

$$\bar{\mu}(x_k) = \hat{\mu}_k(x_k),$$

and discards the remaining components. For states  $x$  within the target set  $T$ , the MPC applies the control  $\bar{\mu}(x)$  that keeps the state within  $T$ , as per Eq. (6.56), at no further cost [ $\bar{\mu}(x) = \bar{\mu}(x)$  for  $x \in T$ ].

We make a constrained controllability assumption, namely that the problem solved at each stage by MPC has a feasible solution for all  $x_k \in X$  with  $x_k \notin T$  (this assumption can be checked using the target tube reachability methods of Section 4.6.2). Note that this problem is a potentially difficult minimax control problem, which generally must be solved by DP (cf. the algorithm of Section 1.6).

### Example 6.5.4

This example is a version of the preceding one, modified to account for the presence of disturbances. We consider the scalar linear system

$$x_{k+1} = x_k + u_k + w_k,$$

and the state and control constraints

$$x_k \in X = \{x \mid |x| \leq 1.5\}, \quad u_k \in U(x_k) = \{u \mid |u| \leq 1\},$$

and assume that the disturbances satisfy

$$w_k \in W(x_k, u_k) = \{w \mid |w| \leq 0.2\}.$$

We select  $m = 2$ , and it can be verified that for the target set

$$T = \{x \mid |x| \leq 0.2\},$$

the constrained controllability assumption is satisfied, and the condition (6.56) is also satisfied using some control law  $\bar{\mu}$ , namely  $\bar{\mu}(x) = -x$ .

The associated 2-stage minimax control problem to be solved at each stage by MPC requires a DP solution. At the last stage, assuming  $x \notin T$ , the DP algorithm calculates

$$\tilde{J}(x) = \min_{\substack{|u| \leq 1, \\ |x+u+w| \leq 0.2 \text{ for all } |w| \leq 0.2}} \left[ \max_{|w| \leq 0.2} (x^2 + u^2) \right].$$

This is a straightforward minimization. It is feasible if and only if  $|x| \leq 1$ , and it yields a minimizing policy for the last stage:

$$\hat{\mu}_1(x) = -x, \quad \text{for all } x \notin T \text{ with } |x| \leq 1,$$

and a cost-to-go

$$\tilde{J}(x) = 2x^2, \quad \text{for all } x \notin T \text{ with } |x| \leq 1.$$

At the first stage, the DP algorithm calculates

$$\min_{\substack{|u| \leq 1, \\ |x+u-w| \leq 1 \text{ for all } |w| \leq 0.2}} \left[ \max_{|w| \leq 0.2} (x^2 + u^2 + \tilde{J}(x + u + w)) \right],$$

or, since the maximum over  $w$  is attained for  $w = 0.2 \operatorname{sgn}(x+u)$ ,

$$\min_{\substack{|u| \leq 1, \\ |x+u+0.2 \operatorname{sgn}(x+u)| \leq 1}} \left[ x^2 + u^2 + 2(x + u + 0.2 \operatorname{sgn}(x+u))^2 \right],$$

or

$$\min_{\substack{|u| \leq 1 \\ |x+u| \leq 0.8}} [x^2 + u^2 - 2(x^2 + u^2 + 2xu + 0.4|x+u| + 0.04)].$$

This minimization is again straightforward, and yields the MPC

$$\bar{\mu}(x) = \begin{cases} -\min[x, \frac{2}{3}(x+0.2)] & \text{if } x \in (0.2, 1.5], \\ \min[-x, -\frac{2}{3}(x-0.2)] & \text{if } x \in [-1.5, -0.2]. \end{cases}$$

This piecewise linear form of the MPC should be compared with the corresponding linear form,  $\bar{\mu}(x) = -(2/3)x$ , of Example 6.5.3, in which there are no disturbances.

The stability analysis of MPC (in the modified sense of reaching the target set  $T$  with finite quadratic cost, for all possible disturbance values) is similar to the one given earlier in the absence of disturbances. It is also possible to view MPC in the presence of disturbances as a special case of a rollout algorithm, suitably modified to take account of the set-membership description of the disturbances. The details of this analysis are sketched in Exercise 6.21.

### 6.5.3 Restricted Structure Policies

We will now introduce a general unifying suboptimal control scheme that contains as special cases several of the control schemes we have discussed: OLFC, POLFC, rollout, and MPC. The idea is to simplify the problem by selectively restricting the information and/or the controls available to the controller, thereby obtaining a restricted but more tractable problem structure, which can be used conveniently in a one-step lookahead context.

An example of such a structure is one where fewer observations are obtained, or one where the control constraint set is restricted to a single or a small number of given controls at each state. Generally, a restricted structure is associated with a problem where the optimal cost achievable is less favorable than in the given problem; this will be made specific in what follows. At each stage, we compute a policy that solves an optimal control problem involving the remaining stages and the restricted problem structure. The control applied at the given stage is the first component of the restricted policy thus obtained.

An example of a suboptimal control approach that uses a restricted structure is the OLFC, where one uses the information available at a given stage as the starting point for an open-loop computation (where future observations are ignored). Another example is the rollout algorithm, where at a given stage one restricts the controls available at future stages to be those applied by some heuristic policy. Still another example is MPC, which

under some conditions may be viewed as a form of rollout algorithm, as discussed in the preceding subsection.

For a problem with  $N$  stages, implementation of the suboptimal scheme to be discussed requires the solution of a problem involving the restricted structure at each stage. The horizon of this problem starts at the current stage, call it  $k$ , and extends up to the final stage  $N$ . This solution yields a control  $u_k$  for stage  $k$  and a policy for the remaining stages  $k+1, \dots, N-1$  (which must obey the constraints of the restricted structure). The control  $u_k$  is used at the current stage, while the policy for the remaining stages  $k+1, \dots, N-1$  is discarded. The process is repeated at the next stage  $k+1$ , using the additional information obtained between stages  $k$  and  $k+1$ ; this is similar to CEC, OLFC, multistage lookahead, and MPC.

Similarly, for an infinite horizon model, implementation of the suboptimal scheme requires, at each stage  $k$ , the solution of a problem involving the restricted structure and a (rolling) horizon of fixed length. The solution yields a control  $u_k$  for stage  $k$  and a policy for each of the remaining stages. The control  $u_k$  is then used at stage  $k$ , and the policy for the remaining stages is discarded. For simplicity in what follows, we will focus attention to the finite horizon case, but the analysis applies, with minor modifications, to infinite horizon cases as well.

Our main result is that the performance of the suboptimal control scheme is no worse than the one of the restricted problem, i.e., the problem corresponding to the restricted structure. This result unifies and generalizes our analysis for open-loop-feedback control (which is known to improve the cost of the optimal open-loop policy, cf. Section 6.2), for the rollout algorithm (which is known to improve the cost of the corresponding heuristic policy, cf. Section 6.4), and for model predictive control (where under some reasonable assumptions, stability of the suboptimal closed-loop control scheme is guaranteed, cf. Section 6.5.2).

For simplicity, we focus on the imperfect state information framework for stationary finite-state Markov chains with  $N$  stages (cf. Section 5.4.2); the ideas apply to much more general problems with perfect and imperfect state information, as well problems with an infinite horizon. We assume that the system state is one of a finite number of states denoted  $1, 2, \dots, n$ . When a control  $u$  is applied, the system moves from state  $i$  to state  $j$  with probability  $p_{ij}(u)$ . The control  $u$  is chosen from a finite set  $U$ . Following a state transition, an observation is made by the controller. There is a finite number of possible observation outcomes, and the probability of each depends on the current state and the preceding control. The information available to the controller at stage  $k$  is the information vector

$$I_k = (z_1, \dots, z_k, u_0, \dots, u_{k-1}),$$

where for all  $i$ ,  $z_i$  and  $u_i$  are the observation and control at stage  $i$ , respectively. Following the observation  $z_k$ , a control  $u_k$  is chosen by the controller,

and a cost  $g(x_k, u_k)$  is incurred, where  $x_k$  is the current (hidden) state. The terminal cost for being at state  $x$  at the end of the  $N$  stages is denoted  $G(x)$ . We wish to minimize the expected value of the sum of costs incurred over the  $N$  stages.

As discussed in Section 5.4, we can reformulate the problem into a problem of perfect state information where the objective is to control the column vector of conditional probabilities

$$p_k = (p_k^1, \dots, p_k^n)',$$

with

$$p_k^j = P(x_k = j | I_k), \quad j = 1, \dots, n.$$

We refer to  $p_k$  as the *belief state*, and we note that it evolves according to an equation of the form

$$p_{k+1} = \Phi(p_k, u_k, z_{k+1}).$$

The function  $\Phi$  represents an estimator, as discussed in Section 5.4. The initial belief state  $p_0$  is given.

The corresponding DP algorithm was given in Section 5.4, and has the form

$$J_k(p_k) = \min_{u_k \in U} \left[ p_k' g(u_k) + E_{z_{k+1}} \{ J_{k+1}(\Phi(p_k, u_k, z_{k+1})) | p_k, u_k \} \right],$$

where  $g(u_k)$  is the column vector with components  $g(1, u_k), \dots, g(n, u_k)$ , and  $p_k' g(u_k)$ , the expected stage cost, is the inner product of the vectors  $p_k$  and  $g(u_k)$ . The algorithm starts at stage  $N$ , with

$$J_N(p_N) = p_N' G,$$

where  $G$  is the column vector with components  $G(1), \dots, G(n)$ , and proceeds backwards.

We will also consider another control structure, where the information vector is

$$\bar{I}_k = (\bar{z}_1, \dots, \bar{z}_k, u_0, \dots, u_{k-1}), \quad k = 0, \dots, N-1,$$

with  $\bar{z}_i$  being some observation for each  $i$  (possibly different from  $z_i$ ), and the control constraint set at each  $p_k$  is a given set  $\bar{U}(p_k)$ . The probability distribution of  $z_k$  given  $x_k$  and  $u_{k-1}$  is known, and may be different than the one of  $z_k$ . Also  $\bar{U}(p_k)$  may be different than  $U$  [in what follows, we will assume that  $\bar{U}(p_k)$  is a subset of  $U$ ].

We introduce a suboptimal policy, which at stage  $k$ , and starting with the current belief state  $p_k$ , applies a control  $\bar{\mu}_k(p_k) \in \bar{U}$ , based on the assumption that the future observations and control constraints will be

according to the restricted structure. More specifically, this policy chooses the control at the typical stage  $k$  and state  $x_k$  as follows:

**Restricted Structure Policy:** At stage  $k$  and state  $x_k$ , apply the control

$$\bar{\mu}_k(p_k) = u_k,$$

where

$$(u_k, \bar{\mu}_{k+1}(\bar{z}_{k+1}, u_k), \dots, \bar{\mu}_{N-1}(\bar{z}_{N-1}, \dots, \bar{z}_{N-1}, u_k, \dots, u_{N-2}))$$

is a policy that attains the optimal cost achievable from stage  $k$  onward with knowledge of  $p_k$  and with access to the future observations  $\bar{z}_{k+1}, \dots, \bar{z}_{N-1}$  (in addition to the future controls), and subject to the constraints

$$u_k \in U, \quad \mu_{k+1}(p_{k+1}) \in \bar{U}(p_{k+1}), \dots, \mu_{N-1}(p_{N-1}) \in \bar{U}(p_{N-1}).$$

Let  $\bar{J}_k(p_k)$  be the cost-to-go, starting at belief state  $p_k$  at stage  $k$ , of the restricted structure policy  $\{\bar{\mu}_0, \dots, \bar{\mu}_{N-1}\}$  just described. This is given by the DP algorithm

$$\bar{J}_k(p_k) = p_k' G(\bar{\mu}_k(p_k)) + E_{z_{k+1}} \left\{ \bar{J}_{k+1}(\Phi(p_k, \bar{\mu}_k(p_k), z_{k+1})) | p_k, \bar{\mu}_k(p_k) \right\} \quad (6.57)$$

for all  $p_k$  and  $k$ , with the terminal condition  $\bar{J}_N(p_N) = p_N' G$  for all  $p_N$ .

Let us also denote by  $J_k^r(p_k)$  the optimal cost-to-go of the restricted problem, i.e., the one where the observations and control constraints of the restricted structure are used exclusively. This is the optimal cost achievable, starting at belief state  $p_k$  at stage  $k$ , using the observations  $\bar{z}_i$ ,  $i = k+1, \dots, N-1$ , and subject to the constraints

$$u_k \in \bar{U}(p_k), \quad \mu_{k+1}(p_{k+1}) \in \bar{U}(p_{k+1}), \dots, \mu_{N-1}(p_{N-1}) \in \bar{U}(p_{N-1}).$$

We will show, under certain assumptions to be introduced shortly, that

$$\bar{J}_k(p_k) \leq J_k^r(p_k), \quad \forall p_k, k = 0, \dots, N-1,$$

and we will also obtain a readily computable upper bound to  $\bar{J}_k(p_k)$ . To this end, for a given belief vector  $p_k$  and control  $u_k \in U$ , we consider three optimal costs-to-go corresponding to three different patterns of availability of information and control restriction over the remaining stages  $k+1, \dots, N-1$ . We denote:

$Q_k(p_k, u_k)$ : The cost achievable from stage  $k$  onward starting with  $p_k$ , applying  $u_k$  at stage  $k$ , and optimally choosing each future control  $u_i$ ,  $i = k+1, \dots, N-1$ , with knowledge of  $p_k$ , the observations  $z_{k+1}, \dots, z_i$  and the controls  $u_k, \dots, u_{i-1}$ , and subject to the constraint  $u_i \in U$ .

$Q_k^c(p_k, u_k)$ : The cost achievable from stage  $k$  onward starting with  $p_k$ , applying  $u_k$  at stage  $k$ , and optimally choosing each future control  $u_i$ ,  $i = k+1, \dots, N-1$ , with knowledge of  $p_k$ , the observations  $\bar{z}_{k+1}, \dots, \bar{z}_i$ , and the controls  $u_k, \dots, u_{i-1}$ , and subject to the constraint  $u_i \in \bar{U}(p_k)$ . Note that this definition is equivalent to

$$Q_k^c(p_k, \bar{\mu}_k(p_k)) = \min_{u_k \in U} Q_k^c(p_k, u_k), \quad (6.58)$$

where  $\bar{\mu}_k(p_k)$  is the control applied by the restricted structure policy just described.

$\hat{Q}_k^c(p_k, u_k)$ : The cost achievable from stage  $k$  onward starting with  $p_k$ , applying  $u_k$  at stage  $k$ , optimally choosing the control  $u_{k+1}$  with knowledge of  $p_k$ , the observation  $z_{k+1}$ , and the control  $u_k$ , subject to the constraint  $u_{k+1} \in U$ , and optimally choosing each of the remaining controls  $u_i$ ,  $i = k+2, \dots, N-1$ , with knowledge of  $p_k$ , the observations  $z_{k+1}, \bar{z}_{k+2}, \dots, \bar{z}_i$ , and the controls  $u_k, \dots, u_{i-1}$ , and subject to the constraints  $u_i \in \bar{U}(p_i)$ .

Thus, the difference between  $Q_k^c(p_k, u_k)$  and  $Q_k(p_k, u_k)$  is due to the difference in the control constraint and the information available to the controller at all future stages  $k+1, \dots, N-1$  [ $\bar{U}(p_{k+1}), \dots, \bar{U}(p_{N-1})$  versus  $U$ , and  $\bar{z}_{k+1}, \dots, \bar{z}_{N-1}$  versus  $z_{k+1}, \dots, z_{N-1}$ , respectively]. The difference between  $Q_k^c(p_k, u_k)$  and  $\hat{Q}_k^c(p_k, u_k)$  is due to the difference in the control constraint and the information available to the controller at the *single* stage  $k+1$  [ $\bar{U}(p_{k+1})$  versus  $U$ , and  $\bar{z}_{k+1}$  versus  $z_{k+1}$ , respectively]. Our key assumptions are that

$$\bar{U}(p_k) \subset U, \quad \forall p_k, k = 0, \dots, N-1, \quad (6.59)$$

$$Q_k(p_k, u_k) \leq \hat{Q}_k^c(p_k, u_k) \leq Q_k^c(p_k, u_k), \quad \forall p_k, u_k \in U, k = 0, \dots, N-1. \quad (6.60)$$

Roughly, this means that the control constraint  $\bar{U}(p_k)$  is more stringent than  $U$ , and the observations  $\bar{z}_{k+1}, \dots, \bar{z}_{N-1}$  are “weaker” (no more valuable in terms of improving the cost) than the observations  $z_{k+1}, \dots, z_{N-1}$ . Consequently, if Eqs. (6.59) and (6.60) hold, we may interpret a controller that uses in part the observations  $\bar{z}_k$  and the control constraints  $\bar{U}(p_k)$ , in place of  $z_k$  and  $U$ , respectively, as “handicapped” or “restricted.”

Let us denote:

$J_k(p_k)$ : The optimal cost-to-go of the original problem, starting at belief state  $p_k$  at stage  $k$ . This is given by

$$J_k(p_k) = \min_{u_k \in U} Q_k(p_k, u_k). \quad (6.61)$$

$J_k^c(p_k)$ : The optimal cost achievable, starting at belief state  $p_k$  at stage  $k$ , using the observations  $\bar{z}_i$ ,  $i = k+1, \dots, N-1$ , and subject to the constraints

$$u_k \in U, \quad \mu_{k+1}(p_{k+1}) \in \bar{U}(p_{k+1}), \dots, \mu_{N-1}(p_{N-1}) \in \bar{U}(p_{N-1}).$$

This is given by

$$J_k^c(p_k) = \min_{u_k \in U} Q_k^c(p_k, u_k), \quad (6.62)$$

and it is the cost that is computed when solving the optimization problem of stage  $k$  in the restricted structure policy scheme. Note that we have for all  $p_k$ ,

$$J_k^r(p_k) = \min_{u_k \in \bar{U}(p_k)} Q_k^c(p_k, u_k) \geq \min_{u_k \in U} Q_k^c(p_k, u_k) = J_k^c(p_k), \quad (6.63)$$

where the inequality holds in view of the assumption  $\bar{U}(p_k) \subset U$ .

Our main result is the following:

**Proposition 6.5.1:** Under the assumptions (6.59) and (6.60), there holds

$$J_N(p_N) \leq \bar{J}_k(p_k) \leq J_k^c(p_k) \leq J_k^r(p_k), \quad \forall p_k, k = 0, \dots, N-1.$$

**Proof:** The inequality  $J_k(p_k) \leq \bar{J}_k(p_k)$  is evident, since  $J_k(p_k)$  is the optimal cost-to-go over a class of policies that includes the restricted structure policy  $\{\bar{\mu}_0, \dots, \bar{\mu}_{N-1}\}$ . Also the inequality  $J_k^c(p_k) \leq J_k^r(p_k)$  follows from the definitions; see Eq. (6.63). We prove the remaining inequality  $\bar{J}_k(p_k) \leq J_k^c(p_k)$  by induction on  $k$ .

We have  $\bar{J}_N(p_N) = J_N^c(p_N) = 0$  for all  $p_N$ . Assume that for all  $p_{k+1}$ , we have

$$\bar{J}_{k+1}(p_{k+1}) \leq J_{k+1}^c(p_{k+1}).$$

Then, for all  $p_k$ ,

$$\begin{aligned} \bar{J}_k(p_k) &= p'_k g(\bar{\mu}_k(p_k)) + E_{z_{k+1}} \left\{ \bar{J}_{k+1}(\Phi(p_k, \bar{\mu}_k(p_k), z_{k+1})) \mid p_k, \bar{\mu}_k(p_k) \right\} \\ &\leq p'_k g(\bar{\mu}_k(p_k)) + E_{z_{k+1}} \left\{ J_{k+1}^c(\Phi(p_k, \bar{\mu}_k(p_k), z_{k+1})) \mid p_k, \bar{\mu}_k(p_k) \right\} \\ &= p'_k g(\bar{\mu}_k(p_k)) \\ &\quad + E_{z_{k+1}} \left\{ \min_{u_{k+1} \in U} Q_{k+1}^c(\Phi(p_k, \bar{\mu}_k(p_k), z_{k+1}), u_{k+1}) \mid p_k, \bar{\mu}_k(p_k) \right\} \\ &= \hat{Q}_k^c(p_k, \bar{\mu}_k(p_k)) \\ &\leq Q_k^c(p_k, \bar{\mu}_k(p_k)) \\ &= J_k^c(p_k), \end{aligned}$$

where the first equality holds by Eq. (6.57), the first inequality holds by the induction hypothesis, the second equality holds by Eq. (6.62), the third equality holds by the definition of  $\hat{Q}_k^c$ , the second inequality holds by the assumption (6.60), and the last equality holds from the definition (6.58) of the restricted structure policy. The induction is complete. **Q.E.D.**

The main conclusion from the proposition is that the performance of the restricted structure policy  $\{\bar{\mu}_0, \dots, \bar{\mu}_{N-1}\}$  is no worse than the performance associated with the restricted control structure. Furthermore, at each stage  $k$ , the value  $J_k^c(p_k)$ , which is obtained as a byproduct of the online computation of the control  $\bar{\mu}_k(p_k)$ , is an upper bound to the cost-to-go  $\bar{J}_k(p_k)$  of the suboptimal policy. This is consistent with Prop. 6.2.1, which shows the cost improvement property of the OLFC, and Prop. 6.3.1, which is the basis for the cost improvement property of the rollout algorithm and the stability property of MPC.

## 6.6 ADDITIONAL TOPICS IN APPROXIMATE DP

We close this chapter with a brief discussion of a few additional topics relating to approximate DP. We first address some of the discretization issues that arise when continuous state and control spaces are approximated by discrete spaces for DP computation purposes. We then describe some alternative suboptimal control approaches.

### 6.6.1 Discretization

An important practical issue is how to deal computationally with problems involving nondiscrete state and control spaces. In particular, problems with continuous state, control, or disturbance spaces must be discretized in order to execute the DP algorithm. Here each of the continuous spaces of the problem is replaced by a space with a finite number of elements, and the system equation is appropriately modified. Thus the resulting approximating problem involves a finite number of states, and a set of transition probabilities between these states. Once the discretization is done, the DP algorithm is executed to yield the optimal cost-to-go function and an optimal policy for the discrete approximating problem. The optimal cost function and/or the optimal policy for the discrete problem may then be extended to an approximate cost function or a suboptimal policy for the original continuous problem through some form of interpolation. We have already seen an example of such a process in the context of aggregation (cf. Example 6.3.13).

A prerequisite for success of this type of discretization is *consistency*. By this we mean that the optimal cost of the original problem should be

achieved in the limit as the discretization becomes finer and finer. Consistency is typically guaranteed if there is a “sufficient amount of continuity” in the problem; for example, if the cost-to-go functions and the optimal policy of the original problem are continuous functions of the state. This in turn can be guaranteed through appropriate continuity assumptions on the original problem data (see the references given in Section 6.7).

Continuity of the cost-to-go functions may be sufficient to guarantee consistency, even if the optimal policy is discontinuous in the state. What may happen here is that for some states there may be a large discrepancy between the optimal policy of the continuous problem and the optimal policy of its discretized version, but this discrepancy may occur over a portion of the state space that diminishes as the discretization becomes finer. As an example consider the inventory control problem of Section 4.2 with nonzero fixed cost. We obtained an optimal policy of the  $(s, S)$  type

$$\mu_k^*(x_k) = \begin{cases} S_k - x_k & \text{if } x_k < s_k, \\ 0 & \text{if } x_k \geq s_k, \end{cases}$$

which is discontinuous at the lower threshold  $s_k$ . The optimal policy obtained from the discretized problem may not approximate well the optimal around the point of discontinuity  $s_k$ , but it is intuitively clear that the discrepancy has a diminishing effect on the optimal cost as the discretization becomes finer.

If the original problem is defined in continuous time, then the time must also be discretized, to obtain a discrete-time approximating problem. The issue of consistency becomes now considerably more complex, because the time discretization affects not only the system equation but also the control constraint set. In particular, the control constraint set may change considerably as we pass to the appropriate discrete-time approximation. As an example, consider the two-dimensional system

$$\dot{x}_1(t) = u_1(t), \quad \dot{x}_2(t) = u_2(t),$$

with the control constraint

$$u_1(t) \in \{-1, 1\}, \quad u_2(t) \in \{-1, 1\}.$$

It can be seen then that the state at time  $t + \Delta t$  can be anywhere within the square centered at  $x(t)$  with side of length  $2\Delta t$  (note that the effect of any control in the interval  $[-1, 1]$  can be obtained in the continuous-time system by “chattering” between the +1 and -1 controls). Thus, given  $\Delta t$ , the appropriate discrete-time approximation of the control constraint set should involve a discretized version of the entire unit square, the *convex hull* of the control constraint set of the continuous-time problem. An example that illustrates some of the pitfalls associated with the discretization process is given in Exercise 6.10.

A general method to address the discretization issues of continuous-time/space optimal control is the aggregation/discretization approach described in Example 6.3.13. The idea is to discretize, in addition to time, the state space using some finite grid, and then to approximate the cost-to-go of nongrid states by linear interpolation of the cost-to-go values of the nearby grid states. Thus, the grid states  $x^1, \dots, x^M$  are suitably selected within the state space, and each nongrid state  $x$  is expressed as

$$x = \sum_{m=1}^M w^m(x) x^m,$$

for some nonnegative weights  $w^m(x)$ , which add to 1. When this is worked out (cf. Example 6.3.13), one ends up with a *stochastic* optimal control problem having as states the finite number of grid states, and transition probabilities that are determined from the weights  $w^m(x)$  above. If the original continuous-time optimal control problem has fixed terminal time, the resulting stochastic control approximation has finite horizon. If the terminal time of the original problem is free and subject to optimization, the stochastic control approximation is of the stochastic shortest path type to be discussed in Section 7.2. Finally, once the costs-to-go  $\tilde{J}_k(x^m)$  of the grid states in the stochastic approximating problem are computed, the cost-to-go of each nongrid state  $x$  at stage  $k$  is approximated by

$$\tilde{J}_k(x) = \sum_{m=1}^M w^m(x) \tilde{J}_k(x^m).$$

We refer to the papers by Gonzalez and Rofman [GoR85], and by Falcone [Fal87] for an account of this approach, and to the survey paper by Kushner [Kus90], and the monograph by Kushner and Dupuis [KuD92] for a detailed analysis of the associated consistency issues.

An important special case is the continuous-space shortest path problem, described in Exercise 6.10. For the corresponding stochastic shortest path problem, a finitely terminating adaptation of the Dijkstra shortest path algorithm has been developed by Tsitsiklis [Tsi95]; see Exercises 2.10 and 2.11 in Chapter 2 of Vol. II. Other related works are the papers by Bertsekas, Guerriero, and Musmanno [BGM95], and Polymenakos, Bertsekas, and Tsitsiklis [PBT98], which develop continuous space versions of label correcting algorithms, such as the Small-Label-First algorithm discussed in Section 2.3.1.

### 6.6.2 Other Approximation Approaches

We mention briefly three additional approaches for using approximations. In the first approach, the optimal cost-to-go functions  $J_k(x_k)$  are approximated with functions  $\tilde{J}_k(x_k, r_k)$ , where  $r_0, r_1, \dots, r_{N-1}$  are unknown parameter vectors, which are chosen to minimize some form of error in the

DP equations; for example by solving the problem

$$\begin{aligned} \min_{r_0, \dots, r_{N-1}} \sum_{(x_k, k) \in \tilde{S}} & \left| \tilde{J}_k(x_k, r_k) \right| \\ & - \left. \min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k), r_{k+1}) \right\} \right|^2 \end{aligned} \quad (6.64)$$

where  $\tilde{S}$  is a suitably chosen subset of “representative” state-time pairs. The above minimization can be attempted using some type of gradient method. Note that there is some difficulty in doing so because the cost function of Eq. (6.64) may be nondifferentiable for some values of  $r$ . However, there are adaptations of gradient methods that work with nondifferentiable cost functions, and for which we refer to the specialized literature. One possibility is to replace the nondifferentiable term

$$\min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k), r_{k+1}) \right\}$$

by a smooth approximation (see Bertsekas [Ber82b], Ch. 3). The approach of approximating cost-to-go functions by minimizing the error in the DP equations will also be discussed in more detail within an infinite horizon context (see Vol. II, Section 2.3).

In the second approach, optimal policies are directly approximated. In particular, suppose that the control space is a Euclidean space, and that we obtain, for a finite number of states  $x^i$ ,  $i = 1, \dots, m$ , the minimizing controls

$$\hat{\mu}_k(x^i) = \arg \min_{u_k \in U_k(x^i)} E \left\{ g_k(x^i, u_k, w_k) + \tilde{J}_{k+1}(f_k(x^i, u_k, w_k), r_{k+1}) \right\}.$$

We can then approximate the optimal policy  $\mu_k(x_k)$  by a function of some given form

$$\tilde{\mu}_k(x_k, s_k),$$

where  $s_k$  is a vector of parameters obtained by solving the problem

$$\min_{s_k} \sum_{i=1}^m \left\| \hat{\mu}_k(x^i) - \tilde{\mu}_k(x^i, s_k) \right\|^2. \quad (6.65)$$

In the case of deterministic optimal control problems, we can take advantage of the equivalence between open-loop and feedback control to carry out the approximation process more efficiently. In particular, for such problems we may select a representative finite subset of initial states, and generate an optimal open-loop trajectory starting from each of these states. (Gradient-based methods can often be used for this purpose.) Each

of these trajectories yields a sequence of pairs  $(x_k, J_k(x_k))$  and a sequence of pairs  $(x_k, \mu_k(x_k))$ , which can be used in the least-squares approximation procedures discussed above. In particular, we can use the exact values  $J_k(x^i)$  and  $\mu_k(x^i)$  obtained from the optimal open-loop trajectories in place of  $\tilde{J}_k(x^i)$  and  $\tilde{\mu}_k(x^i)$ , respectively, in the least-squares problems of Eqs. (6.51) and (6.65).

In the third approach, sometimes called *optimization in policy space*, we parameterize the set of policies by a vector  $s = (s_0, s_1, \dots, s_{N-1})$  and we optimize the corresponding cost over this vector. In particular, we consider policies of the form

$$\pi(s) = \{\bar{\mu}_0(x_0, s_0), \dots, \bar{\mu}_{N-1}(x_{N-1}, s_{N-1})\},$$

where the  $\bar{\mu}_k(\cdot, \cdot)$  are functions of a given form. We then minimize over  $s$  the expected cost

$$E\{J_{\pi(s)}(x_0)\},$$

where  $J_{\pi(s)}(x_0)$  is the cost of the policy  $\pi(s)$  starting from the initial state  $x_0$ , and the expected value is taken with respect to a suitable probability distribution of  $x_0$ . One of the difficulties associated with this approach is that the optimization of  $E\{J_{\pi(s)}(x_0)\}$  over  $s$  may be time-consuming, because it may require some brute force search, local search, or random search method. Sometimes, it is possible to use a gradient-based approach for optimizing  $E\{J_{\pi(s)}(x_0)\}$  over  $s$ , but this can be time-consuming as well.

In an important special case of this approach, the parameterization of the policies is indirect through a parameterization of an approximate cost-to-go function. In particular, for a given parameter vector  $s = (s_0, \dots, s_{N-1})$ , we define

$$\hat{\mu}_k(x_k, s_k) = \arg \min_{u_k \in U_k(x_k)} E\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k), s_k)\},$$

where  $\tilde{J}_{k+1}(\cdot, \cdot)$  is a function of a given form. For example,  $\tilde{J}_{k+1}$  may represent a linear feature-based architecture, where  $s_k$  is a vector of adjustable scalar weights multiplying corresponding features of states  $x_{k+1}$  (cf. Section 6.3.5). Note that the policies

$$\pi(s) = \{\hat{\mu}_0(x_0, s_0), \dots, \hat{\mu}_{N-1}(x_{N-1}, s_{N-1})\}$$

form a class of one-step lookahead policies parametrized by  $s$ . By optimizing over  $s$  the corresponding expected cost  $E\{J_{\pi(s)}(x_0)\}$ , we end up with a one-step lookahead policy that is optimal within this class.

## 6.7 NOTES, SOURCES, AND EXERCISES

Many schemes for suboptimal control have been discussed in this chapter, and it may be helpful to summarize them here. Most of these schemes are

based on one-step lookahead, whereby we apply at stage  $k$  and state  $x_k$  the control  $\bar{\mu}_k(x_k)$  that minimizes over  $u_k \in U_k(x_k)$

$$E\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\},$$

where  $\tilde{J}_{k+1}$  is a suitable cost-to-go approximating function; in some cases, the control constraint set and/or the expected cost per stage are also approximated. The principal distinction between alternative approaches is the method for calculating  $\tilde{J}_{k+1}$ . There are several possibilities (and variations thereto), the principal of which are:

- (a) *Explicit cost-to-go approximation*. Here  $\tilde{J}_{k+1}$  is computed *off-line* in one of a number of ways.
  - (1) By solving a related problem, obtained for example by *aggregation* or *enforced decomposition*, and by deriving  $\tilde{J}_{k+1}$  from the optimal cost-to-go of that problem.
  - (2) By introducing a parametric approximation architecture, possibly using features. The parameters of the architecture are tuned by some form of heuristic or systematic method.
- (b) *Implicit cost-to-go approximation*. Here the values of  $\tilde{J}_{k+1}$  at the states  $f_k(x_k, u_k, w_k)$  are computed *on-line* as needed, by using an open-loop computation (optimal or suboptimal/heuristic, with or without a rolling horizon). We focused on a few possibilities, all which were interpreted under the unifying framework of restricted structure policies in Section 6.5.3:
  - (1) *Open-loop-feedback control*, where an optimal open-loop computation is used, starting from the state  $x_k$  (in the case of perfect state information) or the conditional probability distribution of the state (in the case of imperfect state information).
  - (2) *Rollout*, where the cost-to-go of a suboptimal/heuristic policy is used as  $\tilde{J}_{k+1}$ . This cost is computed as necessary by on-line simulation (which in some variants may be approximate and/or use a rolling horizon).
  - (3) *Model predictive control*, where an optimal control computation is used in conjunction with a rolling horizon. This computation is deterministic, possibly based on a simplification of the original problem via certainty equivalence, but there is also a minimax variant that implicitly involves reachability of target tube computations.

A few important variations of the preceding schemes should be mentioned. The first is the use of *multistep lookahead*, which aims to improve the performance of one-step lookahead, at the expense of increased on-line

computation. The second is the use of *certainty equivalence*, which simplifies the off-line and on-line computations by replacing the current and future unknown disturbances  $w_k, \dots, w_{N-1}$  with nominal values. A third variation, which applies to problems of imperfect state information, is to use one of the preceding schemes with the unknown state  $x_k$  replaced by some estimate.

While the idea of one-step lookahead is old, it has gained a lot of credibility recently, thanks to extensive research on approximate dynamic programming and wide acceptance of model predictive control in practical applications. With experience and research, the relative merits of different approaches have been clarified to some extent, and it is now understood that some schemes possess desirable theoretical performance guarantees, while others do not. In particular, in this chapter, we have discussed qualitative and/or quantitative performance guarantees for open-loop feedback control (cf. Prop. 6.2.1 and the discussion of Section 6.5.3), rollout (Examples 6.3.1 and 6.3.2), and model predictive control (the stability guarantee discussed in Section 6.5.2). The performance guarantee for certainty equivalent control (cf. Prop. 6.3.2 and Example 6.3.3) is weaker, and indeed for some stochastic problems, certainty equivalent control may be outperformed by open-loop control (see Exercise 6.2). For additional theoretical analysis on performance bounds, see Wittenhouse [Wit69], [Wit70]. Despite the recent progress in theory and practical experience, the methodology for performance analysis of suboptimal control schemes is not very satisfactory at present, and the validation of a suboptimal policy by simulation is often essential in practice. This is true of all approaches described in this chapter, including ones that are not based on one-step lookahead, such as approximation in policy space (cf. Section 6.6.2).

Excellent surveys of adaptive control, which contain many other references, are given by Aström [Ast83] and Kumar [Kum85]. Self-tuning regulators received wide attention following the paper by Aström and Wittenmark [AsW73]. For textbook treatments of adaptive control, see Aström and Wittenmark [AsW94], Goodwin and Sin [GoS84], Hernandez-Lerma [Her89], Ioannou and Sun [IoS96], Krstic, Kanellakopoulos, and Kokotovic [KKK95], Kumar and Varaiya [KuV86], Sastry, Bodson, and Bartram [SBB89], and Slotine and Li [SL91].

Open-loop feedback control was suggested by Dreyfus [Dre65]. Its superiority over open-loop control (cf. Prop. 6.2.1) was established by the author in the context of minimax control [Ber72b]. A generalization of this result is given by White and Harrington [WhH80]. The POLFC was proposed in Bertsekas [Ber76].

Stochastic programming problems have been discussed in detail in the literature (see the texts by Birge and Louveaux [BiL97], Kall and Wallace [KaW94], and Prekopa [Pre95]). The connections between stochastic programming and stochastic optimal control have been highlighted by Varaiya and Wets [VaW89].

There is a long history of limited lookahead approximations in specific application contexts. The performance bounds for limited lookahead policies, given in Section 6.3.1 and Exercises 6.11–6.15 are new.

The main idea of rollout algorithms, obtaining an improved policy starting from some other suboptimal policy using a one-time policy improvement, has appeared in several DP application contexts. In the context of game-playing computer programs, it has been proposed by Abramson [Abr90] and by Tesauro [TeG96]. The name “rollout” was coined by Tesauro in specific reference to rolling the dice in the game of backgammon. In Tesauro’s proposal, a given backgammon position is evaluated by “rolling out” many games starting from that position, using a simulator, and the results are averaged to provide a “score” for the position. The internet contains a lot of material on computer backgammon and the use of rollout, in some cases in conjunction with multistep lookahead and cost-to-go approximation.

The application of rollout algorithms to discrete optimization problems has its origin in the neuro-dynamic programming work of the author and J. Tsitsiklis [BeT96], and has been further formalized by Bertsekas, Tsitsiklis, and Wu [BTW97], Bertsekas [Ber97], and Bertsekas and Castanon [BeC99]. The analysis of the breakthrough problem (Example 6.4.2) is based on unpublished joint work of the author with D. Castanon and J. Tsitsiklis. An analysis of the optimal policy and some suboptimal policies for this problem is given by Pearl [Pea84]. A discussion of rollout algorithms as applied to network optimization problems may be found in the author’s network optimization book [Ber98a]. The technique for variance reduction in the calculation of  $Q$ -factor differences (Section 6.4.2) is from Bertsekas [Ber97].

For work on rollout algorithms, see Christodouleas [Chr97], Secomandi [Sec00], [Sec01], [Sec03], Bertsimas and Demir [BeD02], Ferris and Voelker [FeV02], [FeV04], McGovern, Moss, and Barto [MMB02], Savagaonkar, Givan, and Chong [SGC02], Bertsimas and Popescu [BeP03], Guerriero and Mancini [GuM03], Tu and Pattipati [TuP03], Wu, Chong, and Givan [WCG03], Chang, Givan, and Chong [CGC04], Meloni, Pacciarelli, and Pranzo [MPP04], and Yan, Diaconis, Rusmevichientong, and Van Roy [YDR05]. These works discuss a broad variety of applications and case studies, and generally report positive computational experience.

The model predictive control approach has become popular in a variety of control system design contexts, and particularly in chemical process control, where meeting explicit control and state constraints is an important practical issue. Over time, there has been increasing awareness of the connection with the problem of reachability of target tubes, set-membership descriptions of uncertainty, and minimax control (see the discussion of Section 4.6). The stability analysis given here is based on the work of Keerthi and Gilbert [KeG88]. For extensive surveys of the field, see Morari and Lee [MoL99], and Mayne et. al. [MRM00], who give many references. For

related textbooks, see Camacho and Bordons [CaB04], and Maciejowski [Mac02]. The connection with rollout algorithms and one-time policy iteration reported in Section 6.5.2 is new. The material of Section 6.5.3 on the unifying suboptimal control framework based on restricted structure policies is also new.

The computational requirements for solving stochastic optimal control problems are discussed from the point of view of computational complexity in the survey by Blondel and Tsitsiklis [BIT00], who give several additional references; see also Rust [Rus97]. For consistency analyses of various discretization and approximation procedures for discrete-time stochastic optimal control problems, see Bertsekas [Ber75], [Ber76a], Chow and Tsitsiklis [ChT89], [ChT91], Fox [Fox71], and Whitt [Whi78], [Whi79]. A discretization method that takes advantage of the special structure of finite-state imperfect state information problems was first given by Lovejoy [Lov91a]; see also the survey [Lov91b]. For more recent work, based on the aggregation/discretization approach described in Example 6.3.13, see Yu and Bertsekas [YuB04]. The discretization issues of continuous-time/space optimal control problems have been the subject of considerable research; see Gonzalez and Rofman [GoR85], Falcone [Fal87], Kushner [Kus90], and Kushner and Dupuis [KuD92], which give additional sources.

There have been important algorithmic developments for certain types of continuous space shortest path problems. A finitely terminating adaptation of the label setting (Dijkstra) method has been developed by Tsitsiklis [Tsi95]. This method was rediscovered later, under the name “fast marching method,” by Sethian [Set99a], [Set99b], who discusses several other related methods and many applications, as well as by Helmsen et al. [HPC96]. Efficient analogs of label correcting algorithms for continuous space shortest path problems were developed by Bertsekas, Guerriero, and Musmanno [BGM95], and Polymenakos, Bertsekas, and Tsitsiklis [PBT98].

## E X E R C I S E S

### 6.1

Consider a problem with perfect state information involving the  $n$ -dimensional linear system of Section 4.1:

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

### Sec. 6.7 Notes, Sources, and Exercises

and a cost function of the form

$$\underset{\substack{w_k \\ k=0,1,\dots,N-1}}{E} \left\{ g_N(c' x_N) + \sum_{k=0}^{N-1} g_k(u_k) \right\},$$

where  $c \in \mathbb{R}^n$  is a given vector. Show that the DP algorithm for this problem can be carried out over a one-dimensional state space.

### 6.2

Argue that for a one-stage problem, the optimal open-loop controller and the OLFC are both optimal. Construct an example where the CEC may be strictly suboptimal. Also work out the following two-stage example, due to [ThW66], which involves the following two-dimensional linear system with scalar control and disturbance:

$$x_{k+1} = x_k + b u_k + d w_k, \quad k = 0, 1,$$

where  $b = (1, 0)'$  and  $d = (1/2, \sqrt{2}/2)'$ . The initial state is  $x_0 = 0$ . The controls  $u_0$  and  $u_1$  are unconstrained. The disturbances  $w_0$  and  $w_1$  are independent random variables and each takes the values 1 and -1 with equal probability 1/2. Perfect state information prevails. The cost is

$$\underset{w_0, w_1}{E} \{ \|x_2\| \},$$

where  $\|\cdot\|$  denotes the usual Euclidean norm. Show that the CEC with nominal values  $\bar{w}_0 = \bar{w}_1 = 0$  has worse performance than the optimal open-loop controller. In particular, show that the optimal open-loop cost and the optimal closed-loop cost are both  $\sqrt{2}/2$ , but the cost corresponding to the CEC is 1.

### 6.3

Consider a two-stage problem with perfect state information involving the scalar system

$$x_0 = 1, \quad x_1 = x_0 + u_0 + w_0, \quad x_2 = f(x_1, u_1).$$

The control constraints are  $u_0, u_1 \in \{0, -1\}$ . The random variable  $w_0$  takes the values 1 and -1 with equal probability 1/2. The function  $f$  is defined by

$$f(1, 0) = f(1, -1) = f(-1, 0) = f(-1, -1) = 0.5,$$

$$f(2, 0) = 0, \quad f(2, -1) = 2, \quad f(0, -1) = 0.6, \quad f(0, 0) = 2.$$

The cost function is

$$\underset{w_0}{E} \{ x_2 \}.$$

(a) Show that one possible OLFC for this problem is

$$\bar{u}_0(x_0) = -1, \quad \bar{u}_1(x_1) = \begin{cases} 0 & \text{if } x_1 = \pm 1, 2, \\ -1 & \text{if } x_1 = 0, \end{cases}$$

and the resulting cost is 0.5.

- (b) Show that one possible CEC for this problem is

$$\bar{\mu}_0(x_0) = 0, \quad \bar{\mu}_1(x_1) = \begin{cases} 0 & \text{if } x_1 = \pm 1, 2, \\ -1 & \text{if } x_1 = 0, \end{cases}$$

and the resulting cost is 0.3. Show also that this CEC is an optimal feedback controller.

#### 6.4

Consider the system and cost function of Exercise 6.3 but with the difference that

$$f(0, -1) = 0.$$

- (a) Show that the controller of part (a) of Exercise 6.3 is both an OLFC and a CEC, and that the corresponding cost is 0.5.
- (b) Assume that the control constraint set for the first stage is  $\{0\}$  rather than  $\{0, -1\}$ . Show that the controller of part (b) of Exercise 6.3 is both an OLFC and a CEC, and that the corresponding cost is 0. Note: This problem illustrates a pathology that occurs generically in suboptimal control; that is, if the control constraint set is restricted, the performance of a suboptimal scheme may be improved. To see this, consider a problem and a suboptimal control scheme that is not optimal for the problem. Let  $\pi^* = \{\mu_0^*, \dots, \mu_{N-1}^*\}$  be an optimal policy. Restrict the control constraint set so that only the optimal control  $\mu_k^*(x_k)$  is allowed at state  $x_k$ . Then the cost attained by the suboptimal control scheme will be improved.

#### 6.5

Consider the ARMAX model

$$y_{k+1} + ay_k = bu_k + \epsilon_{k+1} + ce_k,$$

where the parameters  $a$ ,  $b$ , and  $c$  are unknown. The controller hypothesizes a model of the form

$$y_{k+1} + ay_k = u_k + \epsilon_{k+1}$$

and uses at each  $k$  the minimum variance/certainty equivalent control

$$u_k^* = \hat{a}_k y_k,$$

where  $\hat{a}_k$  is the least-squares estimate of  $a$  obtained as

$$\hat{a}_k = \arg \min_a \sum_{n=1}^k (y_n + ay_{n-1} - u_{n-1}^*)^2.$$

Write a computer program to test the hypothesis that the sequence  $\{\hat{a}_k\}$  converges to the optimal value, which is  $(c - a)/b$ . Experiment with values  $|a| < 1$  and  $|a| > 1$ .

#### Sec. 6.7 Notes, Sources, and Exercises

#### 6.8 (Semilinear Systems)

Consider the basic problem for semilinear systems (Exercise 1.13 in Chapter 1). Show that the OLFC, and the CEC, with nominal values of the disturbances equal to their expected values, are optimal for this problem.

#### 6.7

Consider the production control problem of Example 6.3.8 for the case where there is only one part type ( $n = 1$ ), and assume that the cost per stage is a convex function  $g$  with  $\lim_{|x| \rightarrow 1} g(x) = \infty$ .

- (a) Show that the cost-to-go function  $J_k(x_k, \alpha_k)$  is convex as a function of  $x_k$  for each value of  $\alpha_k$ .
- (b) Show that for each  $k$  and  $\alpha_k$ , there is a target value  $\bar{x}_{k+1}$  such that for each  $x_k$  it is optimal to choose the control  $u_k \in U_k(\alpha_k)$  that brings  $x_{k+1} = x_k + u_k - d_k$  as close as possible to  $\bar{x}_{k+1}$ .

#### 6.8

Provide a careful argument showing that searching a chess position with and without  $\alpha\beta$  pruning will give the same result.

#### 6.9

In a version of the game of Nim, two players start with a stack of five pennies and take turns removing one, two, or three pennies from the stack. The player who removes the last penny loses. Construct the game tree and verify that the second player to move will win with optimal play.

#### 6.10 (Continuous Space Shortest Path Problems)

Consider the two-dimensional system

$$\dot{x}_1(t) = u_1(t), \quad \dot{x}_2(t) = u_2(t),$$

with the control constraint  $\|u(t)\| = 1$ . We want to find a state trajectory that starts at a given point  $x(0)$ , ends at another given point  $x(T)$ , and minimizes

$$\int_0^T r(x(t)) dt.$$

The function  $r(\cdot)$  is nonnegative and continuous, and the final time  $T$  is subject to optimization. Suppose we discretize the plane with a mesh of size  $\Delta$  that passes through  $x(0)$  and  $x(T)$ , and we introduce a shortest path problem of going from

$x(0)$  to  $x(T)$  using moves of the following type: from each mesh point  $\bar{x} = (\bar{x}_1, \bar{x}_2)$  we can go to each of the mesh points  $(\bar{x}_1 + \Delta, \bar{x}_2)$ ,  $(\bar{x}_1 - \Delta, \bar{x}_2)$ ,  $(\bar{x}_1, \bar{x}_2 + \Delta)$ , and  $(\bar{x}_1, \bar{x}_2 - \Delta)$ , at a cost  $r(\bar{x})\Delta$ . Show by example that this is a bad discretization of the original problem in the sense that the shortest distance need not approach the optimal cost of the original problem as  $\Delta \rightarrow 0$ .

### 6.11 (Discretization of Convex Problems)

Consider a problem with state space  $S$ , where  $S$  is a convex subset of  $\mathbb{R}^n$ . Suppose that  $\hat{S} = \{y_1, \dots, y_M\}$  is a finite subset of  $S$  such that  $\hat{S}$  is the convex hull of  $\hat{S}$ , and consider a one-step lookahead policy based on approximate cost-to-go functions  $\tilde{J}_0, \tilde{J}_1, \dots, \tilde{J}_N$  defined as follows:

$$\tilde{J}_N(x) = g_N(x), \quad \forall x \in S,$$

and for  $k = 1, \dots, N-1$ ,

$$\tilde{J}_k(x) = \min \left\{ \sum_{i=1}^M \lambda_i \tilde{J}_k(y_i) \mid \sum_{i=1}^M \lambda_i y_i = x, \sum_{i=1}^M \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, M \right\},$$

where  $\tilde{J}_k(x)$  is defined by

$$\hat{J}_k(x) = \min_{u \in U_k(x)} E \left\{ g_k(x, u, w_k) + \tilde{J}_{k+1}(f_k(x, u, w_k)) \right\}.$$

Thus  $\tilde{J}_k$  is obtained from  $\hat{J}_{k-1}$  as a “grid-based” convex piecewise linear approximation to  $\hat{J}_k$  based on the  $M$  values

$$\tilde{J}_k(y_1), \dots, \tilde{J}_k(y_M).$$

Assume that the cost functions  $g_k$  and the system functions  $f_k$  are such that the function  $\tilde{J}_k$  is real-valued and convex over  $S$  whenever  $\tilde{J}_{k+1}$  is real-valued and convex over  $S$ . Use Prop. 6.3.1 to show that the cost-to-go functions  $\tilde{J}_k$  corresponding to the one-step lookahead policy satisfy for all  $x \in S$

$$\tilde{J}_k(x) \leq \hat{J}_k(x) \leq \tilde{J}_k(x), \quad k = 0, 1, \dots, N-1.$$

### 6.12 (One-Step Lookahead with Cost per Stage and Constraint Approximations)

Consider a one-step lookahead policy as in Section 6.3, where  $\tilde{J}_k(x_k)$  is chosen to be the optimal cost-to-go of a different problem where the costs-per-stage and control constraint sets are  $\tilde{g}_k(x_k, u_k, w_k)$  and  $\tilde{U}_k(x_k)$ , respectively, [rather than  $g_k(x_k, u_k, w_k)$  and  $U_k(x_k)$ ]. Assume that for all  $k, x_k, u_k, w_k$ , we have

$$g_k(x_k, u_k, w_k) \leq \tilde{g}_k(x_k, u_k, w_k), \quad \tilde{U}_k(x_k) \subset \overline{U}_k(x_k).$$

Use Prop. 6.3.1 to show that the costs-to-go  $\tilde{J}_k$  of the one-step lookahead policy satisfy

$$\tilde{J}_k(x_k) \leq \hat{J}_k(x_k),$$

for all  $x_k$  and  $k$ . Extend this result for the case where  $\tilde{g}_k$  satisfies instead

$$g_k(x_k, u_k, w_k) \leq \tilde{g}_k(x_k, u_k, w_k) + \delta_k,$$

where  $\delta_k$  are some scalars that depend only on  $k$ .

### 6.13 (One-Step Lookahead/Rollout for Shortest Paths)

Consider a graph with nodes  $1, \dots, N$ , and the problem of finding a shortest path from each of the nodes  $1, \dots, N-1$  to node  $N$  with respect to a given set of arc lengths  $a_{ij}$ . We assume that all cycles have positive length. Let  $F(i)$ ,  $i = 1, \dots, N$ , be some given scalars with  $F(N) = 0$ , and denote

$$\hat{F}(i) = \min_{j \in J_i} [a_{ij} + F(j)], \quad i = 1, \dots, N-1, \quad (6.66)$$

where for each  $i$ ,  $J_i$  is a nonempty subset of the set of neighbor nodes  $\{j \mid (i, j)$  is an arc $\}$ .

- (a) Assume that  $\hat{F}(i) \leq F(i)$  for all  $i = 1, \dots, N-1$ . Let  $j(i)$  attain the minimum in Eq. (6.66) and consider the graph consisting of the  $N-1$  arcs  $(i, j(i))$ ,  $i = 1, \dots, N-1$ . Show that this graph contains no cycles and for each  $i = 1, \dots, N-1$ , it contains a unique path  $P_i$  starting at  $i$  and ending at  $N$ . Show that the length of  $P_i$  is less or equal to  $\hat{F}(i)$ .
- (b) Would the conclusion of part (a) hold if the cycles of the original graph are assumed to have nonnegative (rather than positive) length?
- (c) Let  $F(i)$  be the length of some given path  $\bar{P}_i$  from node  $i$  to node  $N$  with  $F(N) = 0$ , and assume that for the first arc of  $\bar{P}_i$ , say  $(i, j_i)$ , we have  $j_i \in J_i$ . Assume further that

$$F(i) \geq a_{ij_i} + F(j_i)$$

[this is satisfied with equality if  $\bar{P}_i$  consists of arc  $(i, j_i)$  followed by path  $\bar{P}_{j_i}$ , which is true if the paths  $\bar{P}_i$  form a tree rooted at the destination  $N$ ; for example if the paths  $\bar{P}_i$  were obtained by solving some related shortest path problem]. Show that  $\hat{F}(i) \leq F(i)$  for all  $i = 1, \dots, N-1$ .

- (d) Assume that  $J_i = \{j \mid (i, j)$  is an arc $\}$ . Let  $P_i$  be the paths obtained as in part (a) when the scalars  $F(i)$  are generated as in part (c). Interpret  $P_i$  as the result of a rollout algorithm that uses an appropriate heuristic, and show that for each  $i$ , the length of  $P_i$  is less or equal to the length of  $\bar{P}_i$ .
- (e) Assume that  $J_i = \{j \mid (i, j)$  is an arc $\}$ . Let us view the scalars  $F(i)$  as the node labels of a label correcting method. This method starts with labels  $F(i) = \infty$  for all  $i \neq N$  and  $F(N) = 0$ , and at each step sets

$$F(i) = \min_{\{j \mid (i, j) \text{ is an arc}\}} [a_{ij} + F(j)]$$

for some node  $i \neq N$  for which the above equality is violated (the method terminates if this equality holds for all  $i \neq N$ ). Show that in the course of this method, the labels  $F(i)$  satisfy the assumptions of part (c) at all times (at or before termination) for which  $F(i) < \infty$  for all  $i$ .

### 6.14 (Performance Bounds for Two-Step Lookahead Policies)

Consider a two-step lookahead policy as in Section 6.3, and assume that for all  $x_k$  and  $k$ , we have

$$\hat{J}_k(x_k) \leq \tilde{J}_k(x_k),$$

where  $\tilde{J}_N = g_N$  and for  $k = 0, \dots, N-1$ ,

$$\hat{J}_k(x_k) = \min_{u_k \in \bar{U}_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k)) \right\}.$$

Consider the cost-to-go functions  $\bar{J}_k$  corresponding to the two-step lookahead policy that uses  $\hat{J}_k$  and  $\bar{U}_k(x_k)$ . Show that for all  $x_k$  and  $k$ , we have

$$\bar{J}_k(x_k) \leq J_k^*(x_k) \leq \hat{J}_k(x_k) \leq \tilde{J}_k(x_k),$$

where  $J_k^*$  is the function obtained by two DP iterations starting from  $\tilde{J}_{k+2}$ :

$$J_k^*(x_k) = \min_{u_k \in \bar{U}_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k)) \right\}.$$

### 6.15 (Rollout Algorithms with Errors)

Consider the graph search problem of Section 6.4.1 and let  $\mathcal{H}$  be a sequentially improving base heuristic. Suppose that we generate a path  $(i_1, \dots, i_{\bar{m}})$  according to

$$i_{m+1} = \arg \min_{j \in N(i_m)} \hat{H}(j), \quad m = 1, \dots, \bar{m}-1,$$

where  $\hat{H}(j)$  differs from the cost  $H(j)$  of the base heuristic by the error

$$e(j) = \hat{H}(j) - H(j).$$

- (a) Assuming that  $|e(j)| \leq \epsilon$  for all  $j$ , show that the cost of the generated path is less than or equal to  $H(i_1) + 2(\bar{m}-1)\epsilon$ . Hint: Use the relation

$$H(i_{m+1}) - \epsilon \leq \hat{H}(i_{m+1}) = \min_{j \in N(i_m)} \hat{H}(j) \leq \min_{j \in N(i_m)} H(j) + \epsilon \leq H(i_m) + \epsilon.$$

- (b) Modify the estimate of part (a) for the case where we have  $0 \leq e(j) \leq \epsilon$  for all  $j$ , and for the case where we have  $-\epsilon \leq e(j) \leq 0$  for all  $j$ .
- (c) Consider the case where  $\mathcal{H}$  is optimal so that  $H(j) = J^*(j)$ , and derive a bound on the difference between the cost of the generated path and the optimal cost starting from  $i_1$ .

### 6.16 (Breakthrough Problem with Random Heuristic)

Consider the breakthrough problem of Example 6.4.2 with the difference that instead of the greedy heuristic, we use the *random* heuristic, which at a given node selects one of the two outgoing arcs with equal probability. Denote by

$$D_k = p^k$$

the probability of success of the random heuristic in a graph of  $k$  stages, and by  $R_k$  the probability of success of the corresponding rollout algorithm. Show that for all  $k$

$$R_k = p(2-p)R_{k-1} + p^2 D_{k-1}(1-R_{k-1}).$$

and that

$$\frac{R_k}{D_k} = (2-p) \frac{R_{k-1}}{D_{k-1}} + p(1-R_{k-1}).$$

Conclude that  $R_k/D_k$  increases exponentially with  $k$ .

### 6.17

Consider the breakthrough problem of Example 6.4.2 with the difference that there are three outgoing arcs from each node instead of two. Each arc is free with probability  $p$ , independently of other arcs. Derive an equation for the ratio  $R_k/G_k$ , where  $G_k$  is the probability of success of the greedy heuristic for a  $k$ -stage problem, and  $R_k$  is the probability of success of the corresponding rollout algorithm. Verify that the results of Example 6.4.2 still hold in a qualitative sense, and that  $R_k/G_k$  increases linearly with  $k$ .

### 6.18 (Breakthrough Problem with a Rolling Horizon Rollout)

Consider the breakthrough problem of Example 6.4.2 and consider a rolling horizon-type of rollout algorithm that uses a greedy base heuristic with  $l$ -step lookahead. This is the same algorithm as the one described in Example 6.4.2, except that if both outgoing arcs of the current node at stage  $k$  are free, the rollout algorithm considers the two end nodes of those arcs, and from each of them it runs the greedy algorithm for  $\min\{l, N-k-1\}$  steps. Consider a Markov chain with  $l+1$  states, where states  $i = 0, \dots, l-1$  correspond to the path generated by the greedy algorithm being blocked after  $i$  arcs. State  $l$  corresponds to the path generated by the greedy algorithm being unblocked after  $l$  arcs.

- (a) Derive the transition probabilities for this Markov chain so that it models the operation of the rollout algorithm.
- (b) Use computer simulation to generate the probability of a breakthrough, and to demonstrate that for large values of  $N$ , the optimal value of  $l$  is roughly constant and much smaller than  $N$  (this can also be justified analytically, by using properties of Markov chains).

### 6.19 (Rollout for Constrained DP)

Consider the deterministic constrained DP problem involving the system

$$x_{k+1} = f_k(x_k, u_k),$$

where we want to minimize the cost function

$$g_N^1(x_N) + \sum_{k=0}^{N-1} g_k^1(x_k, u_k)$$

subject to the constraints

$$g_m^m(x_N) + \sum_{k=0}^{N-1} g_k^m(x_k, u_k) \leq b^m, \quad m = 2, \dots, M;$$

cf. Section 2.3.4. We assume that each state  $x_k$  takes values in a finite set and each control  $u_k$  takes values in a finite constraint set  $U_k(x_k)$  that depends on  $x_k$ . We describe an extension of the rollout algorithm, involving some base heuristic, which is feasible in the sense that when started from the given initial state  $x_0$ , it produces a state/control trajectory that satisfies the constraints of the problem.

Consider a rollout algorithm, which at stage  $k$ , maintains a partial state/control trajectory

$$T_k = (x_0, u_0, x_1, \dots, u_{k-1}, x_k)$$

that starts at the given initial state  $x_0$ , and is such that  $x_{i+1} = f_i(x_i, u_i)$  and  $u_i \in U_i(x_i)$  for all  $i = 0, 1, \dots, k-1$ . For such a trajectory, let  $C^m(x_k)$  be the corresponding values of constraint functions

$$C^m(x_k) = \sum_{i=0}^{k-1} g_i^m(x_i, u_i), \quad m = 2, \dots, M.$$

For each  $u_k \in U_k(x_k)$ , let  $x_{k+1} = f_k(x_k, u_k)$  be the next state, and let  $\tilde{J}(x_{k+1})$  and  $\tilde{C}^m(x_{k+1})$  be the cost-to-go and values of constraint functions of the base heuristic starting from  $x_{k+1}$ .

The algorithm starts with the partial trajectory  $T_0$  that consists of just the initial state  $x_0$ . For each  $k = 0, \dots, N-1$ , and given the current trajectory  $T_k$ , it forms the subset of controls  $u_k \in U_k(x_k)$  that together with the corresponding states  $x_{k+1} = f_k(x_k, u_k)$  satisfy

$$C^m(x_k) + g_k^m(x_k, u_k) + \tilde{C}^m(x_{k+1}) \leq b^m, \quad m = 2, \dots, M.$$

The algorithm selects from this set a control  $u_k$  and corresponding state  $x_{k+1}$  such that

$$g_k^1(x_k, u_k) + \tilde{J}(x_{k+1})$$

is minimum, and then it forms the trajectory  $T_{k+1}$  by adding  $(u_k, x_{k+1})$  to  $T_k$ . Formulate analogs of the assumptions of sequential consistency and sequential improvement of Section 6.4.1, under which the algorithm is guaranteed to generate a feasible state/control trajectory that has no greater cost than the cost associated with the base heuristic. Note: For a description and analysis of a generalized version of this algorithm, see the author's report "Rollout Algorithms for Constrained Dynamic Programming," LIDS Report 2646, MIT, April 2005.

### 6.20 (Rollout for Minimax Problems)

Consider the minimax DP problem, as described in Section 1.6, and a one-step lookahead policy based on lookahead functions  $\bar{J}_1, \dots, \bar{J}_N$ , with  $\bar{J}_N = g_N$ . This is the policy obtained by minimizing at state  $x_k$  the expression

$$\max_{w_k \in W_k(x_k, u_k)} [g_k(x_k, u_k, w_k) - \bar{J}_{k+1}(f_k(x_k, u_k, w_k))]$$

over  $u_k \in U_k(x_k)$ .

- (a) State and prove analogs of Props. 6.3.1 and 6.3.2.
- (b) Consider a rollout algorithm where  $\bar{J}_k$  are the cost-to-go functions corresponding to some base heuristic. Show that the cost-to-go functions  $J_k$  of the rollout algorithm satisfy  $J_k(x_k) \leq \bar{J}_k(x_k)$  for all  $x_k$  and  $k$ .

### 6.21 (MPC with Disturbances)

Consider the MPC framework of Section 6.5.2, including disturbances with set-membership description. Let  $\bar{\mu}$  be the policy obtained from MPC.

- (a) Use the constrained controllability assumption to show that  $\bar{\mu}$  attains reachability of the target tube  $\{X, X, \dots\}$  in the sense that

$$f(x, \bar{\mu}(x), w) \in X, \quad \text{for all } x \in X \text{ and } w \in W(x, \bar{\mu}(x)).$$

- (b) Consider any sequence  $\{x_0, u_0, x_1, u_1, \dots\}$  generated by MPC [i.e.,  $x_0 \in X$ ,  $x_0 \notin T$ ,  $u_k = \bar{\mu}(x_k)$ ,  $x_{k+1} = f(x_k, u_k, w_k)$ , and  $w_k \in W(x_k, u_k)$ ]. Show that

$$\sum_{k=0}^{K_T-1} (x'_k Q x_k + u'_k R u_k) \leq \bar{J}(x_0) < \infty,$$

where  $K_T$  is the smallest integer  $k$  such that  $x_k \in T$  (with  $K_T = \infty$  if  $x_k \notin T$  for all  $k$ ), and  $\bar{J}(x)$  is the optimal cost starting at state  $x \in X$  of the  $m$ -stage minimax control problem solved by MPC. Hint: Argue as in the case where there are no disturbances. Consider an optimal control problem that is similar to the one solved at each stage by MPC, but has one stage less. In particular, given  $x \in X$  with  $x \notin T$ , consider the minimax control problem of finding a policy  $\hat{\mu}_0, \hat{\mu}_1, \dots, \hat{\mu}_{m-2}$  that minimizes

$$\max_{w_i \in W(x_i, \hat{\mu}(x_i)), i=0,1,\dots,m-2} \sum_{i=0}^{m-2} g(x_i, \hat{\mu}_i(x_i)),$$

subject to the system equation constraints

$$x_{i+1} = f(x_i, \hat{\mu}_i(x_i), w_i), \quad i = 0, 1, \dots, m-2,$$

the control and state constraints

$$x_i \in X, \quad \hat{\mu}_i(x_i) \in U(x_i), \quad i = 0, 1, \dots, m-2,$$

and the terminal state constraint

$$x_i \in T, \quad \text{for some } i \in [1, m-1].$$

These constraints must be satisfied for all disturbance sequences with

$$w_i \in W(x_i, \hat{\mu}_i(x_i)), \quad i = 0, 1, \dots, m-2.$$

Let  $\tilde{J}(x_0)$  be the corresponding optimal value, and define  $\tilde{J}(x_0) = 0$  for  $x_0 \in T$ , and  $\tilde{J}(x_0) = \infty$  for all  $x_0 \notin T$  for which the problem has no feasible solution. Show that the control  $\bar{\mu}(x)$  applied by MPC at a state  $x \in X$  with  $x \notin T$ , minimizes over  $u \in U(x)$

$$\max_{u \in W(x, u)} \left[ x' Q x + u' R u + \tilde{J}(f(x, u, w)) \right],$$

and use the fact  $\tilde{J}(x) \leq \tilde{J}(x')$  to show that for all  $x \in X$  with  $x \notin T$ , we have

$$\max_{w \in W(x, u)} \left[ x' Q x + \bar{\mu}(x)' R \bar{\mu}(x) + \tilde{J}(f(x, \bar{\mu}(x), w)) \right] \leq \tilde{J}(x).$$

Conclude that for all  $k$  such that  $x_k \in X$  with  $x_k \notin T$ , we have

$$x_k' Q x_k + u_k' R u_k + \tilde{J}(x_{k+1}) \leq \tilde{J}(x_k),$$

where  $\tilde{J}(x_{k+1}) = 0$  if  $x_{k+1} \in T$ . Add over  $k = 0, 1, \dots, K_T - 1$ .

- (c) Show that under MPC, the state  $x_k$  of the system must belong to  $T$  for all sufficiently large  $k$ , provided that  $\min_{x \in X, x \notin T} x' Q x > 0$ . Use Example 6.5.3 to show the need for this assumption.
- (d) Interpret the policy  $\bar{\mu}$  produced by MPC as a rollout policy with an appropriate base heuristic. Hint: View  $\bar{\mu}$  as a one-step lookahead policy with one-step lookahead approximation function equal to  $\tilde{J}$ , defined in the hint to part (b).



## Introduction to Infinite Horizon Problems

### Contents

7.1. An Overview . . . . .	p. 402
7.2. Stochastic Shortest Path Problems . . . . .	p. 405
7.3. Discounted Problems . . . . .	p. 417
7.4. Average Cost per Stage Problems . . . . .	p. 421
7.5. Semi-Markov Problems . . . . .	p. 435
7.6. Notes, Sources, and Exercises . . . . .	p. 445

In this chapter, we provide an introduction to infinite horizon problems. These problems differ from those considered so far in two respects:

- (a) The number of stages is infinite.
- (b) The system is stationary, i.e., the system equation, the cost per stage, and the random disturbance statistics do not change from one stage to the next.

The assumption of an infinite number of stages is never satisfied in practice, but is a reasonable approximation for problems involving a finite but very large number of stages. The assumption of stationarity is often satisfied in practice, and in other cases it approximates well a situation where the system parameters vary slowly with time.

Infinite horizon problems are interesting because their analysis is elegant and insightful, and the implementation of optimal policies is often simple. For example, optimal policies are typically stationary, i.e., the optimal rule for choosing controls does not change from one stage to the next.

On the other hand, infinite horizon problems generally require more sophisticated analysis than their finite horizon counterparts, because of the need to analyze limiting behavior as the horizon tends to infinity. This analysis is often nontrivial and at times reveals surprising possibilities. Our treatment will be limited to finite-state problems. A far more detailed development, together with applications from a variety of fields can be found in Vol. II of this work.

## 7.1 AN OVERVIEW

There are four principal classes of infinite horizon problems. In the first three classes, we try to minimize the *total cost over an infinite number of stages*, given by

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} E_{w_k \sim w_k(x_0, \dots)} \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}.$$

Here,  $J_\pi(x_0)$  denotes the cost associated with an initial state  $x_0$  and a policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , and  $\alpha$  is a positive scalar with  $0 < \alpha \leq 1$ , called the *discount factor*. The meaning of  $\alpha < 1$  is that future costs matter to us less than the same costs incurred at the present time. As an example, think of  $k$ th period dollars depreciated to initial period dollars by a factor of  $(1+r)^{-k}$ , where  $r$  is a rate of interest; here  $\alpha = 1/(1+r)$ . An important concern in total cost problems is that the limit in the definition of  $J_\pi(x_0)$  be finite. In the first two of the following classes of problems, this is guaranteed

## Sec. 7.1 An Overview

through various assumptions on the problem structure and the discount factor. In the third class, the analysis is adjusted to deal with infinite cost for some of the policies. In the fourth class, this sum need not be finite for any policy, and for this reason, the cost is appropriately redefined.

- (a) *Stochastic shortest path problems*. Here,  $\alpha = 1$  but there is a special cost-free termination state; once the system reaches that state it remains there at no further cost. We will assume a problem structure such that termination is inevitable (this assumption will be relaxed somewhat in Chapter 2 of Vol. II). Thus the horizon is in effect finite, but its length is random and may be affected by the policy being used. These problems will be considered in the next section and their analysis will provide the foundation for the analysis of the other types of problems considered in this chapter.
- (b) *Discounted problems with bounded cost per stage*. Here,  $\alpha < 1$  and the absolute cost per stage  $|g(x, u, w)|$  is bounded from above by some constant  $M$ ; this makes the cost  $J_\pi(x_0)$  well defined because it is the infinite sum of a sequence of numbers that are bounded in absolute value by the decreasing geometric progression  $\{\alpha^k M\}$ . We will consider these problems in Section 7.3.
- (c) *Discounted and undiscounted problems with unbounded cost per stage*. Here the discount factor  $\alpha$  may or may not be less than 1, and the cost per stage may be unbounded. These problems require a complicated analysis because the possibility of infinite cost for some of the policies is explicitly dealt with. We will not consider these problems here; see Chapter 3 of Vol. II.
- (d) *Average cost per stage problems*. Minimization of the total cost  $J_\pi(x_0)$  makes sense only if  $J_\pi(x_0)$  is finite for at least some admissible policies  $\pi$  and some initial states  $x_0$ . Frequently, however, it turns out that  $J_\pi(x_0) = \infty$  for every policy  $\pi$  and initial state  $x_0$  (think of the case where  $\alpha = 1$ , and the cost for every state and control is positive). It turns out that in many such problems the *average cost per stage*, defined by

$$\lim_{N \rightarrow \infty} \frac{1}{N} E_{w_k \sim w_k(x_0, \dots)} \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), w_k) \right\},$$

is well defined and finite. We will consider some of these problems in Section 7.4.

## A Preview of Infinite Horizon Results

There are several analytical and computational issues regarding infinite horizon problems. Many of these revolve around the relation between the

optimal cost-to-go function  $J^*$  of the infinite horizon problem and the optimal cost-to-go functions of the corresponding  $N$ -stage problems. In particular, consider the case  $\alpha = 1$  and let  $J_N(x)$  denote the optimal cost of the problem involving  $N$  stages, initial state  $x$ , cost per stage  $g(x, u, w)$ , and zero terminal cost. The optimal  $N$ -stage cost is generated after  $N$  iterations of the DP algorithm

$$J_{k+1}(x) = \min_{u \in U(x)} E \left\{ g(x, u, w) + J_k(f(x, u, w)) \right\}, \quad k = 0, 1, \dots \quad (7.1)$$

starting from the initial condition  $J_0(x) = 0$  for all  $x$  (note here that we have reversed the time indexing to suit our purposes). Since the infinite horizon cost of a given policy is, by definition, the limit of the corresponding  $N$ -stage costs as  $N \rightarrow \infty$ , it is natural to speculate that:

- (1) The optimal infinite horizon cost is the limit of the corresponding  $N$ -stage optimal costs as  $N \rightarrow \infty$ ; that is,

$$J^*(x) = \lim_{N \rightarrow \infty} J_N(x) \quad (7.2)$$

for all states  $x$ . This relation is extremely valuable computationally and analytically, and, fortunately, it typically holds. In particular, it holds for the models of the next two sections [categories (a) and (b) above]. However, there are some unusual exceptions for problems in category (c) above, and this illustrates that infinite horizon problems should be approached with some care. This issue is discussed in more detail in Vol. II.

- (2) The following limiting form of the DP algorithm should hold for all states  $x$ ,

$$J^*(x) = \min_{u \in U(x)} E \left\{ g(x, u, w) + J^*(f(x, u, w)) \right\},$$

as suggested by Eqs. (7.1) and (7.2). This is not really an algorithm, but rather a system of equations (one equation per state), which has as solution the costs-to-go of all the states. It can also be viewed as a *functional equation* for the cost-to-go function  $J^*$ , and it is called *Bellman's equation*. Fortunately again, an appropriate form of this equation holds for every type of infinite horizon problem of interest.

- (3) If  $\mu(x)$  attains the minimum in the right-hand side of Bellman's equation for each  $x$ , then the policy  $\{\mu, \mu, \dots\}$  should be optimal. This is true for most infinite horizon problems of interest and in particular, for all the models discussed in this chapter.

Most of the analysis of infinite horizon problems revolves around the above three issues and also around the issue of efficient computation of  $J^*$  and an optimal policy. In the next three sections we will provide a discussion of these issues for some of the simpler infinite horizon problems, all of which involve a finite state space.

### Total Cost Problem Formulation

Throughout this chapter we assume a controlled finite-state discrete-time dynamic system whereby, at state  $i$ , the use of a control  $u$  specifies the transition probability  $p_{ij}(u)$  to the next state  $j$ . Here the state  $i$  is an element of a finite state space, and the control  $u$  is constrained to take values in a given finite constraint set  $U(i)$ , which may depend on the current state  $i$ . As discussed in Section 1.1, the underlying system equation is

$$x_{k+1} = w_k,$$

where  $w_k$  is the disturbance. We will generally suppress  $w_k$  from the cost to simplify notation. Thus we will assume a  $k$ th stage cost  $g(x_k, u_k)$  for using control  $u_k$  at state  $x_k$ . This amounts to averaging the cost per stage over all successor states in our calculations, which makes no essential difference in the subsequent analysis. Thus, if  $\tilde{g}(i, u, j)$  is the cost of using  $u$  at state  $i$  and moving to state  $j$ , we use as cost per stage the expected cost  $g(i, u)$  given by

$$g(i, u) = \sum_j p_{ij}(u) \tilde{g}(i, u, j).$$

The total expected cost associated with an initial state  $i$  and a policy  $\pi = \{\mu_0, \mu_1, \dots\}$  is

$$J_\pi(i) = \lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)) \mid x_0 = i \right\},$$

where  $\alpha$  is a discount factor with  $0 < \alpha \leq 1$ . In the following two sections, we will impose assumptions that guarantee the existence of the above limit. The optimal cost from state  $i$ , that is, the minimum of  $J_\pi(i)$  over all admissible  $\pi$ , is denoted by  $J^*(i)$ . A *stationary policy* is an admissible policy of the form  $\pi = \{\mu, \mu, \dots\}$ , and its corresponding cost function is denoted by  $J_\mu(i)$ . For brevity, we refer to  $\{\mu, \mu, \dots\}$  as the stationary policy  $\mu$ . We say that  $\mu$  is optimal if

$$J_\mu(i) = J^*(i) = \min_\pi J_\pi(i), \quad \text{for all states } i.$$

### 7.2 STOCHASTIC SHORTEST PATH PROBLEMS

Here, we assume that there is no discounting ( $\alpha = 1$ ), and to make the cost meaningful, we assume that there is a *special cost-free termination state*  $t$ . Once the system reaches that state, it remains there at no further cost, i.e.,  $p_{tt}(u) = 1$  and  $g(t, u) = 0$  for all  $u \in U(t)$ . We denote by  $1, \dots, n$  the states other than the termination state  $t$ .

We are interested in problems where reaching the termination state is inevitable, at least under an optimal policy. Thus, the essence of the problem is to reach the termination state with minimum expected cost. We call this problem the *stochastic shortest path problem*. The deterministic shortest path problem is obtained as the special case where for each state-control pair  $(i, u)$ , the transition probability  $p_{ij}(u)$  is equal to 1 for a unique state  $j$  that depends on  $(i, u)$ . The reader may also verify that the finite horizon problem of Chapter 1 can be obtained as a special case by viewing as state the pair  $(x_k, k)$  (see also Section 3.6 of Vol. II).

Certain conditions are required to guarantee that, at least under an optimal policy, termination occurs with probability 1. We will make the following assumption that guarantees eventual termination under all policies:

**Assumption 7.2.1:** There exists an integer  $m$  such that regardless of the policy used and the initial state, there is positive probability that the termination state will be reached after no more than  $m$  stages; that is, for all admissible policies  $\pi$  we have

$$\rho_\pi = \max_{i=1, \dots, n} P\{x_m \neq t \mid x_0 = i, \pi\} < 1. \quad (7.3)$$

We note, however, that the results to be presented are valid under more general circumstances.<sup>†</sup> Furthermore, it can be shown that if there exists an integer  $m$  with the property of Assumption 7.2.1, then there also exists an integer less or equal to  $n$  with this property (Exercise 7.12). Thus, we can always use  $m = n$  in Assumption 7.2.1, if no smaller value of  $m$  is

<sup>†</sup> Let us call a stationary policy  $\pi$  *proper* if the condition (7.3) is satisfied for some  $m$ , and call  $\pi$  *improper* otherwise. It can be shown that Assumption 7.2.1 is equivalent to the seemingly weaker assumption that all stationary policies are proper (see Vol. II, Exercise 2.3). However, the results of Prop. 7.2.1 can also be shown under the genuinely weaker assumption that there exists at least one proper policy, and furthermore, every improper policy results in infinite expected cost from at least one initial state (see Bertsekas and Tsitsiklis [BeT89], [BeT91], or Vol. II, Chapter 2). These assumptions, when specialized to deterministic shortest path problems, are similar to the ones we used in Chapter 2. They imply that there is at least one path to the destination from every starting node and that all cycles have positive cost. Still another set of assumptions under which the results of Prop. 7.2.1 hold is described in Exercise 7.28, where again improper policies are allowed, but the stage costs  $g(i, u)$  are assumed nonnegative, and the optimal costs  $J^*(i)$  are assumed finite.

known. Let

$$\rho = \max_\pi \rho_\pi.$$

Note that  $\rho_\pi$  depends only on the first  $m$  components of the policy  $\pi$ . Furthermore, since the number of controls available at each state is finite, the number of distinct  $m$ -stage policies is also finite. It follows that there can be only a finite number of distinct values of  $\rho_\pi$  so that

$$\rho < 1.$$

We therefore have for any  $\pi$  and any initial state  $i$

$$\begin{aligned} P\{x_{2m} \neq t \mid x_0 = i, \pi\} &= P\{x_{2m} \neq t \mid x_m \neq t, x_0 = i, \pi\} \\ &\quad \cdot P\{x_m \neq t \mid x_0 = i, \pi\} \\ &\leq \rho^2. \end{aligned}$$

More generally, for each admissible policy  $\pi$ , the probability of not reaching the termination state after  $km$  stages diminishes like  $\rho^k$  regardless of the initial state, that is,

$$P\{x_{km} \neq t \mid x_0 = i, \pi\} \leq \rho^k, \quad i = 1, \dots, n. \quad (7.4)$$

Thus the limit defining the associated total cost vector  $J_\pi$  exists and is finite, since the expected cost incurred in the  $m$  periods between  $km$  and  $(k-1)m+1$  is bounded in absolute value by

$$m\rho^k \max_{\substack{i=1, \dots, n \\ u \in U(i)}} |g(i, u)|.$$

In particular, we have

$$|J_\pi(i)| \leq \sum_{k=0}^{\infty} m\rho^k \max_{\substack{i=1, \dots, n \\ u \in U(i)}} |g(i, u)| = \frac{m}{1-\rho} \max_{\substack{i=1, \dots, n \\ u \in U(i)}} |g(i, u)|. \quad (7.5)$$

The results of the following proposition are basic and are typical of many infinite horizon problems. The key idea of the proof is that the “tail” of the cost series,

$$\sum_{k=mK}^{\infty} E\{g(x_k, \mu_k(x_k))\}$$

vanishes as  $K$  increases to  $\infty$ , since the probability that  $x_{mK} \neq t$  decreases like  $\rho^K$  [cf. Eq. (7.4)].

**Proposition 7.2.1** Under Assumption 7.2.1, the following hold for the stochastic shortest path problem:

- (a) Given any initial conditions  $J_0(1), \dots, J_0(n)$ , the sequence  $J_k(i)$  generated by the iteration

$$J_{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad i = 1, \dots, n, \quad (7.6)$$

converges to the optimal cost  $J^*(i)$  for each  $i$ . [Note that, by reversing the time index this iteration can be viewed as the DP algorithm for a finite horizon problem with terminal cost function equal to  $J_0$ . In fact,  $J_k(i)$  is the optimal cost starting from state  $i$  of a  $k$ -stage problem with cost per stage given by  $g$  and terminal cost at the end of the  $k$  stages given by  $J_0$ .]

- (b) The optimal costs  $J^*(1), \dots, J^*(n)$  satisfy Bellman's equation,

$$J^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J^*(j) \right], \quad i = 1, \dots, n, \quad (7.7)$$

and in fact they are the unique solution of this equation.

- (c) For any stationary policy  $\mu$ , the costs  $J_\mu(1), \dots, J_\mu(n)$  are the unique solution of the equation

$$J_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J_\mu(j), \quad i = 1, \dots, n.$$

Furthermore, given any initial conditions  $J_0(1), \dots, J_0(n)$ , the sequence  $J_k(i)$  generated by the DP iteration

$$J_{k+1}(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J_k(j), \quad i = 1, \dots, n,$$

converges to the cost  $J_\mu(i)$  for each  $i$ .

- (d) A stationary policy  $\mu$  is optimal if and only if for every state  $i$ ,  $\mu(i)$  attains the minimum in Bellman's equation (7.7).

**Proof:** (a) For every positive integer  $K$ , initial state  $x_0$ , and policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , we break down the cost  $J_\pi(x_0)$  into the portions incurred over

the first  $mK$  stages and over the remaining stages

$$\begin{aligned} J_\pi(x_0) &= \lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \right\} \\ &= E \left\{ \sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k)) \right\} \\ &\quad + \lim_{N \rightarrow \infty} E \left\{ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right\}. \end{aligned}$$

Let  $M$  denote the following upper bound on the cost of an  $m$ -stage cycle, assuming termination does not occur during the cycle,

$$M = m \max_{\substack{i=1, \dots, n \\ u \in U(i)}} |g(i, u)|.$$

The expected cost during the  $K$ th  $m$ -stage cycle [stages  $Km$  to  $(K+1)m-1$ ] is upper bounded by  $M\rho^K$  [cf. Eqs. (7.4) and (7.5)], so that

$$\left| \lim_{N \rightarrow \infty} E \left\{ \sum_{k=mK}^{N-1} g(x_k, \mu_k(x_k)) \right\} \right| \leq M \sum_{k=mK}^{\infty} \rho^k = \frac{\rho^K M}{1 - \rho}.$$

Also, denoting  $J_0(t) = 0$ , let us view  $J_0$  as a terminal cost function and bound its expected value under  $\pi$  after  $mK$  stages. We have

$$\begin{aligned} |E\{J_0(x_{mK})\}| &= \left| \sum_{i=1}^n P(x_{mK} = i \mid x_0, \pi) J_0(i) \right| \\ &\leq \left( \sum_{i=1}^n P(x_{mK} = i \mid x_0, \pi) \right) \max_{i=1, \dots, n} |J_0(i)| \\ &\leq \rho^K \max_{i=1, \dots, n} |J_0(i)|, \end{aligned}$$

since the probability that  $x_{mK} \neq t$  is less or equal to  $\rho^K$  for any policy. Combining the preceding relations, we obtain

$$\begin{aligned} &- \rho^K \max_{i=1, \dots, n} |J_0(i)| + J_\pi(x_0) - \frac{\rho^K M}{1 - \rho} \\ &\leq E \left\{ J_0(x_{mK}) + \sum_{k=0}^{mK-1} g(x_k, \mu_k(x_k)) \right\} \quad (7.8) \\ &\leq \rho^K \max_{i=1, \dots, n} |J_0(i)| + J_\pi(x_0) + \frac{\rho^K M}{1 - \rho}. \end{aligned}$$

Note that the expected value in the middle term of the above inequalities is the  $mK$ -stage cost of policy  $\pi$  starting from state  $x_0$ , with a terminal cost  $J_0(x_{mK})$ ; the minimum of this cost over all  $\pi$  is equal to the value  $J_{mK}(x_0)$ , which is generated by the DP recursion (7.6) after  $mK$  iterations. Thus, by taking the minimum over  $\pi$  in Eq. (7.8), we obtain for all  $x_0$  and  $K$ ,

$$\begin{aligned} -\rho^K \max_{i=1,\dots,n} |J_0(i)| + J^*(x_0) & - \frac{\rho^K M}{1-\rho} \\ & \leq J_{mK}(x_0) \\ & \leq \rho^K \max_{i=1,\dots,n} |J_0(i)| + J^*(x_0) + \frac{\rho^K M}{1-\rho}, \end{aligned} \quad (7.9)$$

and by taking the limit as  $K \rightarrow \infty$ , we obtain  $\lim_{K \rightarrow \infty} J_{mK}(x_0) = J^*(x_0)$ , for all  $x_0$ . Since

$$|J_{mK-q}(x_0) - J_{mK}(x_0)| \leq \rho^K M, \quad q = 1, \dots, m,$$

we see that  $\lim_{K \rightarrow \infty} J_{mK+q}(x_0)$  is the same for all  $q = 1, \dots, m$ , so that we have  $\lim_{K \rightarrow \infty} J_k(x_0) = J^*(x_0)$ .

(b) By taking the limit as  $k \rightarrow \infty$  in the DP iteration (7.6) and using the result of part (a), we see that  $J^*(1), \dots, J^*(n)$  satisfy Bellman's equation. To show uniqueness, observe that if  $J(1), \dots, J(n)$  satisfy Bellman's equation, then the DP iteration (7.6) starting from  $J(1), \dots, J(n)$  just replicates  $J(1), \dots, J(n)$ . It follows from the convergence result of part (a) that  $J(i) = J^*(i)$  for all  $i$ .

(c) Given the stationary policy  $\mu$ , we can consider a modified stochastic shortest path problem, which is the same as the original except that the control constraint set contains only one element for each state  $i$ , the control  $\mu(i)$ ; that is, the control constraint set is  $\tilde{U}(i) = \{\mu(i)\}$  instead of  $U(i)$ . From part (b) we then obtain that  $J_\mu(1), \dots, J_\mu(n)$  solve uniquely Bellman's equation for this modified problem, that is,

$$J_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J_\mu(j), \quad i = 1, \dots, n,$$

and from part (a) it follows that the corresponding DP iteration converges to  $J_\mu(i)$ .

(d) We have that  $\mu(i)$  attains the minimum in Eq. (7.7) if and only if we have

$$\begin{aligned} J^*(i) &= \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J^*(j) \right] \\ &= g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J^*(j), \quad i = 1, \dots, n. \end{aligned}$$

Part (c) and the above equation imply that  $J_\mu(i) = J^*(i)$  for all  $i$ . Conversely, if  $J_\mu(i) = J^*(i)$  for all  $i$ , parts (b) and (c) imply the above equation. Q.E.D.

### Example 7.2.1 (Minimizing Expected Time to Termination)

The case where

$$g(i, u) = 1, \quad i = 1, \dots, n, \quad u \in U(i),$$

corresponds to a problem where the objective is to terminate as fast as possible on the average, while the corresponding optimal cost  $J^*(i)$  is the minimum expected time to termination starting from state  $i$ . Under our assumptions, the costs  $J^*(i)$  uniquely solve Bellman's equation, which has the form

$$J^*(i) = \min_{u \in U(i)} \left[ 1 + \sum_{j=1}^n p_{ij}(u) J^*(j) \right], \quad i = 1, \dots, n.$$

In the special case where there is only one control at each state,  $J^*(i)$  represents the mean first passage time from  $i$  to  $t$  (see Appendix D). These times, denoted  $m_i$ , are the unique solution of the equations

$$m_i = 1 + \sum_{j=1}^n p_{ij} m_j, \quad i = 1, \dots, n.$$

### Example 7.2.2

A spider and a fly move along a straight line at times  $k = 0, 1, \dots$ . The initial positions of the fly and the spider are integer. At each time period, the fly moves one unit to the left with probability  $p$ , one unit to the right with probability  $p$ , and stays where it is with probability  $1-2p$ . The spider, knows the position of the fly at the beginning of each period, and will always move one unit towards the fly if its distance from the fly is more than one unit. If the spider is one unit away from the fly, it will either move one unit towards the fly or stay where it is. If the spider and the fly land in the same position at the end of a period, then the spider captures the fly and the process terminates. The spider's objective is to capture the fly in minimum expected time.

We view as state the distance between spider and fly. Then the problem can be formulated as a stochastic shortest path problem with states  $0, 1, \dots, n$ , where  $n$  is the initial distance. State 0 is the termination state where the spider captures the fly. Let us denote  $p_{1j}(M)$  and  $p_{1j}(\bar{M})$  the transition probabilities from state 1 to state  $j$  if the spider moves and does not move, respectively, and let us denote by  $p_{ij}$  the transition probabilities from a state  $i \geq 2$ . We have

$$p_{ii} = p, \quad p_{i(i-1)} = 1 - 2p, \quad p_{i(i-2)} = p, \quad i \geq 2,$$

$$\begin{aligned} p_{11}(M) &= 2p, \quad p_{10}(M) = 1 - 2p, \\ p_{12}(\overline{M}) &= p, \quad p_{21}(\overline{M}) = 1 - 2p, \quad p_{10}(\overline{M}) = p, \end{aligned}$$

with all other transition probabilities being 0.

For states  $i \geq 2$ , Bellman's equation is written as

$$J^*(i) = 1 + pJ^*(i) + (1 - 2p)J^*(i-1) + pJ^*(i-2), \quad i \geq 2, \quad (7.10)$$

where  $J^*(0) = 0$  by definition. The only state where the spider has a choice is when it is one unit away from the fly, and for that state Bellman's equation is given by

$$J^*(1) = 1 + \min[2pJ^*(1), pJ^*(2) + (1 - 2p)J^*(1)], \quad (7.11)$$

where the first and the second expression within the bracket above are associated with the spider moving and not moving, respectively. By writing Eq. (7.10) for  $i = 2$ , we obtain

$$J^*(2) = 1 + pJ^*(2) + (1 - 2p)J^*(1),$$

from which

$$J^*(2) = \frac{1}{1-p} + \frac{(1-2p)J^*(1)}{1-p}. \quad (7.12)$$

Substituting this expression in Eq. (7.11), we obtain

$$J^*(1) = 1 + \min \left[ 2pJ^*(1), \frac{p}{1-p} + \frac{p(1-2p)J^*(1)}{1-p} + (1-2p)J^*(1) \right],$$

or equivalently,

$$J^*(1) = 1 + \min \left[ 2pJ^*(1), \frac{p}{1-p} + \frac{(1-2p)J^*(1)}{1-p} \right].$$

To solve the above equation, we consider the two cases where the first expression within the bracket is larger and is smaller than the second expression. Thus we solve for  $J^*(1)$  in the two cases where

$$J^*(1) = 1 + 2pJ^*(1), \quad (7.13)$$

$$2pJ^*(1) \leq \frac{p}{1-p} + \frac{(1-2p)J^*(1)}{1-p}, \quad (7.14)$$

and

$$J^*(1) = 1 + \frac{p}{1-p} + \frac{(1-2p)J^*(1)}{1-p}, \quad (7.15)$$

$$2pJ^*(1) \geq \frac{p}{1-p} + \frac{(1-2p)J^*(1)}{1-p}.$$

The solution of Eq. (7.13) is seen to be  $J^*(1) = 1/(1-2p)$ , and by substitution in Eq. (7.14), we find that this solution is valid when

$$\frac{2p}{1-2p} \leq \frac{p}{1-p} - \frac{1}{1-p},$$

or equivalently (after some calculation),  $p \leq 1/3$ . Thus for  $p \leq 1/3$ , it is optimal for the spider to move when it is one unit away from the fly.

Similarly, the solution of Eq. (7.15) is seen to be  $J^*(1) = 1/p$ , and by substitution in Eq. (7.14), we find that this solution is valid when

$$2 \geq \frac{p}{1-p} + \frac{1-2p}{p(1-p)},$$

or equivalently (after some calculation),  $p \geq 1/3$ . Thus, for  $p \geq 1/3$  it is optimal for the spider not to move when it is one unit away from the fly.

The minimal expected number of steps for capture when the spider is one unit away from the fly was calculated earlier to be

$$J^*(1) = \begin{cases} 1/(1-2p) & \text{if } p \leq 1/3, \\ 1/p & \text{if } p \geq 1/3. \end{cases}$$

Given the value of  $J^*(1)$ , we can calculate from Eq. (7.12) the minimal expected number of steps for capture when two units away,  $J^*(2)$ , and we can then obtain the remaining values  $J^*(i)$ ,  $i = 3, \dots, n$ , from Eq. (7.10).

### Value Iteration and Error Bounds

The DP iteration

$$J_{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad i = 1, \dots, n, \quad (7.16)$$

is called *value iteration* and is a principal method for calculating the optimal cost function  $J^*$ . Generally, value iteration requires an infinite number of iterations, although there are important special cases where it terminates finitely (see Vol. II, Section 2.2). Note that from Eq. (7.9) we obtain that the error

$$|J_{mK}(i) - J^*(i)|$$

is bounded by a constant multiple of  $\rho^K$ .

The value iteration algorithm can sometimes be strengthened with the use of some error bounds. In particular, it can be shown (see Exercise 7.13) that for all  $k$  and  $j$ , we have

$$J_{k+1}(j) + (N^*(j) - 1) \bar{c}_k \leq J^*(j) \leq J_{\mu^k}(j) \leq J_{k+1}(j) + (N^k(j) - 1) \bar{c}_k, \quad (7.17)$$

where  $\mu^k$  is such that  $\mu^k(i)$  attains the minimum in the  $k$ th value iteration (7.16) for all  $i$ , and

$N^*(j)$ : The average number of stages to reach  $t$  starting from  $j$  and using some optimal stationary policy,

$N^k(j)$ : The average number of stages to reach  $t$  starting from  $j$  and using the stationary policy  $\mu^k$ ,

$$\underline{c}_k = \min_{i=1,\dots,n} [J_{k+1}(i) - J_k(i)], \quad \bar{c}_k = \max_{i=1,\dots,n} [J_{k+1}(i) - J_k(i)].$$

Unfortunately, the values  $N^*(j)$  and  $N^k(j)$  are easily computed or approximated only in the presence of special problem structure (see for example the next section). Despite this fact, the bounds (7.17) often provide a useful guideline for stopping the value iteration algorithm while being assured that  $J_k$  approximates  $J^*$  with sufficient accuracy.

### Policy Iteration

There is an alternative to value iteration, which always terminates finitely. This algorithm is called *policy iteration* and operates as follows: we start with a stationary policy  $\mu^0$ , and we generate a sequence of new policies  $\mu^1, \mu^2, \dots$ . Given the policy  $\mu^k$ , we perform a *policy evaluation step*, that computes  $J_{\mu^k}(i)$ ,  $i = 1, \dots, n$ , as the solution of the (linear) system of equations

$$J(i) = g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i)) J(j), \quad i = 1, \dots, n, \quad (7.18)$$

in the  $n$  unknowns  $J(1), \dots, J(n)$  [cf. Prop. 7.2.1(c)]. We then perform a *policy improvement step*, which computes a new policy  $\mu^{k+1}$  as

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_{\mu^k}(j) \right], \quad i = 1, \dots, n. \quad (7.19)$$

The process is repeated with  $\mu^{k+1}$  used in place of  $\mu^k$ , unless we have  $J_{\mu^{k+1}}(i) = J_{\mu^k}(i)$  for all  $i$ , in which case the algorithm terminates with the policy  $\mu^k$ . The following proposition establishes the validity of policy iteration.

**Proposition 7.2.2** Under Assumption 7.2.1, the policy iteration algorithm for the stochastic shortest path problem generates an improving sequence of policies [that is,  $J_{\mu^{k+1}}(i) \leq J_{\mu^k}(i)$  for all  $i$  and  $k$ ] and terminates with an optimal policy.

**Proof:** For any  $k$ , consider the sequence generated by the recursion

$$J_{N+1}(i) = g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i)) J_N(j), \quad i = 1, \dots, n,$$

where  $N = 0, 1, \dots$ , and

$$J_0(i) = J_{\mu^k}(i), \quad i = 1, \dots, n.$$

From Eqs. (7.18) and (7.19), we have

$$\begin{aligned} J_0(i) &= g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i)) J_0(j) \\ &\geq g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i)) J_0(j) \\ &= J_1(i), \end{aligned}$$

for all  $i$ . By using the above inequality we obtain (compare with the monotonicity property of DP, Exercice 1.23 in Chapter 1)

$$\begin{aligned} J_1(i) &= g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i)) J_0(j) \\ &\geq g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i)) J_1(j) \\ &= J_2(i), \end{aligned}$$

for all  $i$ , and by continuing similarly we have

$$J_0(i) \geq J_1(i) \geq \dots \geq J_N(i) \geq J_{N+1}(i) \geq \dots, \quad i = 1, \dots, n. \quad (7.20)$$

Since by Prop. 7.2.1(c),  $J_N(i) \rightarrow J_{\mu^{k+1}}(i)$ , we obtain  $J_0(i) \geq J_{\mu^{k+1}}(i)$  or

$$J_{\mu^k}(i) \geq J_{\mu^{k+1}}(i), \quad i = 1, \dots, n, \quad k = 0, 1, \dots$$

Thus the sequence of generated policies is improving, and since the number of stationary policies is finite, we must after a finite number of iterations, say  $k+1$ , obtain  $J_{\mu^k}(i) = J_{\mu^{k+1}}(i)$  for all  $i$ . Then we will have equality holding throughout in Eq. (7.20), which means that

$$J_{\mu^k}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_{\mu^k}(j) \right], \quad i = 1, \dots, n.$$

Thus the costs  $J_{\mu^k}(1), \dots, J_{\mu^k}(n)$  solve Bellman's equation, and by Prop. 7.2.1(b), it follows that  $J_{\mu^k}(i) = J^*(i)$  and that  $\mu^k$  is optimal. Q.E.D.

The linear system of equations (7.18) of the policy evaluation step can be solved by standard methods such as Gaussian elimination, but when the number of states is large, this is cumbersome and time-consuming. A typically more efficient alternative is to approximate the policy evaluation step with a few value iterations aimed at solving the corresponding system (7.18). One can show that the policy iteration method that uses such approximate policy evaluation yields in the limit the optimal costs and an optimal stationary policy, even if we evaluate each policy using an arbitrary positive number of value iterations (see Vol. II, Section 1.3).

Another possibility for approximating the policy evaluation step is to use simulation, and this is a key idea in the rollout algorithm, discussed in Section 6.4. Simulation also plays an important role in the neuro-dynamic programming methodology, discussed in Ch. 2 of Vol. II. In particular, when the number of states is large, one can try to approximate the cost-to-go function  $J_{\mu^k}$  by simulating a large number of trajectories under the policy  $\mu^k$ , and perform some form of least squares fit of  $J_{\mu^k}$  using an approximation architecture (cf. Section 6.3.4). These are a number of variations of this idea, which are discussed in more detail in Vol. II and in the research monograph by Bertsekas and Tsitsiklis [BeT96].

### Linear Programming

Suppose that we use value iteration to generate a sequence of vectors  $J_k = (J_k(1), \dots, J_k(n))$  starting with an initial condition vector  $J_0 = (J_0(1), \dots, J_0(n))$  such that

$$J_0(i) \leq \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_0(j) \right], \quad i = 1, \dots, n.$$

Then we will have  $J_k(i) \leq J_{k+1}(i)$  for all  $k$  and  $i$  (the monotonicity property of DP; Exercise 1.23 in Chapter 1). It follows from Prop. 7.2.1(a) that we will also have  $J_0(i) \leq J^*(i)$  for all  $i$ . Thus  $J^*$  is the "largest"  $J$  that satisfies the constraint

$$J(i) \leq g(i, u) + \sum_{j=1}^n p_{ij}(u) J(j), \quad \text{for all } i = 1, \dots, n \text{ and } u \in U(i).$$

In particular,  $J^*(1), \dots, J^*(n)$  solve the linear program of maximizing  $\sum_{i=1}^n J(i)$  subject to the above constraint (see Fig. 7.2.1). Unfortunately, for large  $n$  the dimension of this program can be very large and its solution can be impractical, particularly in the absence of special structure.

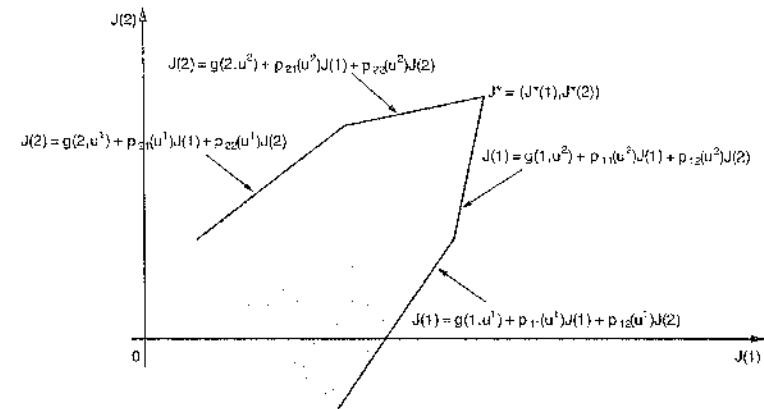
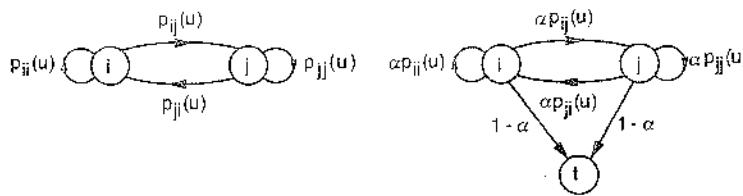


Figure 7.2.1 Linear program associated with a two-state stochastic shortest path problem. The constraint set is shaded, and the objective to maximize is  $J(i) + J(2)$ . Note that because we have  $J(i) \leq J^*(i)$  for all  $i$  and vectors  $J$  in the constraint set, the vector  $J^*$  maximizes any linear cost function of the form  $\sum_{i=1}^n \beta_i J(i)$  where  $\beta_i \geq 0$  for all  $i$ . If  $\beta_i > 0$  for all  $i$ , then  $J^*$  is the unique optimal solution of the corresponding linear program.

## 7.3 DISCOUNTED PROBLEMS

We now consider a discounted problem, where there is a discount factor  $\alpha < 1$ . We will show that this problem can be converted to a stochastic shortest path problem for which the analysis of the preceding section holds. To see this, let  $i = 1, \dots, n$  be the states, and consider an associated stochastic shortest path problem involving the states  $1, \dots, n$  plus an extra termination state  $t$ , with state transitions and costs obtained as follows: From a state  $i \neq t$ , when control  $u$  is applied, a cost  $g(i, u)$  is incurred, and the next state is  $j$  with probability  $\alpha p_{ij}(u)$  and  $t$  with probability  $1 - \alpha$ ; see Fig. 7.3.1. Note that Assumption 7.2.1 of the preceding section is satisfied for the associated stochastic shortest path problem.

Suppose now that we use the same policy in the discounted problem and in the associated stochastic shortest path problem. Then, as long as termination has not occurred, the state evolution in the two problems is governed by the same transition probabilities. Furthermore, the expected cost of the  $k$ th stage of the associated shortest path problem is  $g(x_k, \mu_k(x_k))$  multiplied by the probability that state  $t$  has not yet been reached, which is  $\alpha^k$ . This is also the expected cost of the  $k$ th stage for the discounted problem. We conclude that the cost of any policy starting from a given state, is the same for the original discounted problem and for the associated stochastic shortest path problem. Furthermore, value iteration produces identical iterates for the two problems. We can thus apply



**Figure 7.3.1** Transition probabilities for an  $\alpha$ -discounted problem and its associated stochastic shortest path problem. In the latter problem, the probability that the state is not  $t$  after  $k$  stages is  $\alpha^k$ . The expected cost at each state  $i = 1, \dots, n$  is  $g(i, u)$  for both problems, but it must be multiplied by  $\alpha^k$  because of discounting (in the discounted case) or because it is incurred with probability  $\alpha^k$  when termination has not yet been reached (in the stochastic shortest path case).

the results of the preceding section to this latter problem and obtain the following:

**Proposition 7.3.1** The following hold for the discounted problem:

(a) The value iteration algorithm

$$J_{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad i = 1, \dots, n, \quad (7.21)$$

converges to the optimal costs  $J^*(i)$ ,  $i = 1, \dots, n$ , starting from arbitrary initial conditions  $J_0(1), \dots, J_0(n)$ .

(b) The optimal costs  $J^*(1), \dots, J^*(n)$  of the discounted problem satisfy Bellman's equation,

$$J^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J^*(j) \right], \quad i = 1, \dots, n, \quad (7.22)$$

and in fact they are the unique solution of this equation.

(c) For any stationary policy  $\mu$ , the costs  $J_\mu(1), \dots, J_\mu(n)$  are the unique solution of the equation

$$J_\mu(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^n p_{ij}(\mu(i)) J_\mu(j), \quad i = 1, \dots, n.$$

Furthermore, given any initial conditions  $J_0(1), \dots, J_0(n)$ , the sequence  $J_k(i)$  generated by the DP iteration

$$J_{k+1}(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^n p_{ij}(\mu(i)) J_k(j), \quad i = 1, \dots, n,$$

converges to the cost  $J_\mu(i)$  for each  $i$ .

- (d) A stationary policy  $\mu$  is optimal if and only if for every state  $i$ ,  $\mu(i)$  attains the minimum in Bellman's equation (7.22).
- (e) The policy iteration algorithm given by

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J_{\mu^k}(j) \right], \quad i = 1, \dots, n,$$

generates an improving sequence of policies and terminates with an optimal policy.

**Proof:** Parts (a)-(d) and part (e) are proved by applying parts (a)-(d) of Prop. 7.2.1, and Prop. 7.2.2, respectively, to the associated stochastic shortest path problem described above. **Q.E.D.**

Bellman's equation (7.22) has a familiar DP interpretation. At state  $i$ , the optimal cost  $J^*(i)$  is the minimum over all controls of the sum of the expected current stage cost and the expected optimal cost of all future stages. The former cost is  $g(i, u)$ . The latter cost is  $J^*(j)$ , but since this cost starts accumulating after one stage, it is discounted by multiplication with  $\alpha$ .

As in the case of stochastic shortest path problems [see Eq. (7.9) and the discussion following the proof of Prop. 7.2.1], we can show that the error

$$|J_k(i) - J^*(i)|$$

is bounded by a constant times  $\alpha^k$ . Furthermore, the error bounds (7.17) become

$$J_{k+1}(j) + \frac{\alpha}{1-\alpha} c_k \leq J^*(j) \leq J_{\mu^k}(j) \leq J_{k+1}(j) + \frac{\alpha}{1-\alpha} \bar{c}_k, \quad (7.23)$$

where  $\mu^k$  is such that  $\mu^k(i)$  attains the minimum in the  $k$ th value iteration (7.21) for all  $i$ , and

$$\underline{c}_k = \min_{i=1, \dots, n} [J_{k+1}(i) - J_k(i)], \quad \bar{c}_k = \max_{i=1, \dots, n} [J_{k+1}(i) - J_k(i)],$$

since for the associated stochastic shortest path problem it can be shown that for every policy and starting state, the expected number of stages to reach the termination state  $t$  is  $1/(1 - \alpha)$ , so that the terms  $N^*(j) - 1$  and  $N^k(j) - 1$  appearing in Eq. (7.17) are equal to  $\alpha/(1 - \alpha)$ . We note also that there are a number of additional enhancements to the value iteration algorithm for the discounted problem (see Section 1.3 of Vol. II). There are also discounted cost variants of the approximate policy iteration and linear programming approaches discussed for stochastic shortest path problems.

### Example 7.3.1 (Asset Selling)

Consider an infinite horizon version of the asset selling example of Section 4.4, assuming the set of possible offers is finite. Here, if accepted, the amount  $x_k$  offered in period  $k$ , will be invested at a rate of interest  $r$ . By depreciating the sale amount to period 0 dollars, we view  $(1 + r)^{-k}x_k$  as the reward for selling the asset in period  $k$  at a price  $x_k$ , where  $r > 0$  is the rate of interest. Then we have a total discounted reward problem with discount factor  $\alpha = 1/(1 + r)$ . The analysis of the present section is applicable, and the optimal value function  $J^*$  is the unique solution of Bellman's equation

$$J^*(x) = \max \left[ x, \frac{E\{J^*(w)\}}{1+r} \right],$$

(see Section 4.4). The optimal reward function is characterized by the critical number

$$\bar{\alpha} = \frac{E\{J^*(w)\}}{1+r},$$

which can be calculated as in Section 4.4. An optimal policy is to sell if and only if the current offer  $x_k$  is greater than or equal to  $\bar{\alpha}$ .

### Example 7.3.2

A manufacturer at each time period receives an order for her product with probability  $p$  and receives no order with probability  $1 - p$ . At any period she has a choice of processing all the unfilled orders in a batch, or process no order at all. The cost per unfilled order at each time period is  $c > 0$ , and the setup cost to process the unfilled orders is  $K > 0$ . The manufacturer wants to find a processing policy that minimizes the total expected cost, assuming the discount factor is  $\alpha < 1$  and the maximum number of orders that can remain unfilled is  $n$ .

Here the state is the number of unfilled orders at the beginning of each period, and Bellman's equation takes the form

$$J^*(i) = \min [K + \alpha(1 - p)J^*(0) + \alpha p J^*(1), ci + \alpha(1 - p)J^*(i) + \alpha p J^*(i + 1)], \quad (7.24)$$

for the states  $i = 0, 1, \dots, n - 1$ , and takes the form

$$J^*(n) = K + \alpha(1 - p)J^*(0) + \alpha p J^*(1) \quad (7.25)$$

for state  $n$ . The first expression within brackets in Eq. (7.24) corresponds to processing the  $i$  unfilled orders, while the second expression corresponds to leaving the orders unfilled for one more period. When the maximum  $n$  of unfilled orders is reached, the orders must necessarily be processed, as indicated by Eq. (7.25).

To solve the problem, we observe that the optimal cost  $J^*(i)$  is monotonically nondecreasing in  $i$ . This is intuitively clear, and can be rigorously proved by using the value iteration method. In particular, we can show by using the (finite horizon) DP algorithm that the  $k$ -stage optimal cost functions  $J_k(i)$  are monotonically nondecreasing in  $i$  for all  $k$  (Exercise 7.7), and then argue that the optimal infinite horizon cost function  $J^*(i)$  is also monotonically nondecreasing in  $i$ , since

$$J^*(i) = \lim_{k \rightarrow \infty} J_k(i)$$

by Prop. 7.3.1(a). Given that  $J^*(i)$  is monotonically nondecreasing in  $i$ , from Eq. (7.24) we have that if processing a batch of  $m$  orders is optimal, that is,

$$K + \alpha(1 - p)J^*(0) + \alpha p J^*(1) \leq cm + \alpha(1 - p)J^*(m) + \alpha p J^*(m + 1),$$

then processing a batch of  $m + 1$  orders is also optimal. Therefore a *threshold policy*, that is, a policy that processes the orders if their number exceeds some threshold integer  $m^*$ , is optimal.

We leave it as Exercise 7.8 for the reader to verify that if we start the policy iteration algorithm with a threshold policy, every subsequently generated policy will be a threshold policy. Since there are  $n + 1$  distinct threshold policies, and the sequence of generated policies is improving, it follows that the policy iteration algorithm will yield an optimal policy after at most  $n$  iterations.

## 7.4 AVERAGE COST PER STAGE PROBLEMS

The methodology of the last two sections applies mainly to problems where the optimal total expected cost is finite either because of discounting or because of a cost-free termination state that the system eventually enters. In many situations, however, discounting is inappropriate and there is no natural cost-free termination state. In such situations it is often meaningful to optimize the average cost per stage starting from a state  $i$ , which is defined by

$$J_\pi(i) = \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \mid x_0 = i \right\}.$$

Let us first argue heuristically that for most problems of interest the average cost per stage of a policy and the optimal average cost per stage are independent of the initial state.

To this end we note that the average cost per stage of a policy primarily expresses cost incurred in the long term. Costs incurred in the early stages do not matter since their contribution to the average cost per stage is reduced to zero as  $N \rightarrow \infty$ ; that is,

$$\lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^K g(x_k, \mu_k(x_k)) \right\} = 0, \quad (7.26)$$

for any fixed  $K$ . Consider now a stationary policy  $\mu$  and two states  $i$  and  $j$  such that the system will, under  $\mu$ , eventually reach  $j$  with probability 1 starting from  $i$ . Then intuitively, it is clear that the average costs per stage starting from  $i$  and from  $j$  cannot be different, since the costs incurred in the process of reaching  $j$  from  $i$  do not contribute essentially to the average cost per stage. More precisely, let  $K_{ij}(\mu)$  be the first passage time from  $i$  to  $j$  under  $\mu$ , that is, the first index  $k$  for which  $x_k = j$  starting from  $x_0 = i$  under  $\mu$  (see Appendix D). Then the average cost per stage corresponding to initial condition  $x_0 = i$  can be expressed as

$$J_\mu(i) = \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^{K_{ij}(\mu)-1} g(x_k, \mu(x_k)) \right\} + \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=K_{ij}(\mu)}^N g(x_k, \mu(x_k)) \right\}.$$

If  $E\{K_{ij}(\mu)\} < \infty$  (which is equivalent to assuming that the system eventually reaches  $j$  starting from  $i$  with probability 1; see Appendix D), then it can be seen that the first limit is zero [cf. Eq. (7.26)], while the second limit is equal to  $J_\mu(j)$ . Therefore,

$$J_\mu(i) = J_\mu(j), \quad \text{for all } i, j \text{ with } E\{K_{ij}(\mu)\} < \infty.$$

The preceding argument suggests that the optimal cost  $J^*(i)$  should also be independent of the initial state  $i$  under normal circumstances. To see this, assume that for any two states  $i$  and  $j$ , there exists a stationary policy  $\mu$  (dependent on  $i$  and  $j$ ) such that  $j$  can be reached from  $i$  with probability 1 under  $\mu$ . Then it is impossible that

$$J^*(j) < J^*(i),$$

since when starting from  $i$  we can adopt the policy  $\mu$  up to the time when  $j$  is first reached and then switch to a policy that is optimal when starting from  $j$ , thereby achieving an average cost starting from  $i$  that is equal to  $J^*(j)$ . Indeed, it can be shown that

$$J^*(i) = J^*(j), \quad \text{for all } i, j = 1, \dots, n,$$

under the preceding assumption (see Vol. II, Section 4.2).

### The Associated Stochastic Shortest Path Problem

The results of this section can be proved under a variety of different assumptions (see Chapter 4 in Vol. II). Here, we will make the following assumption, which will allow us to use the stochastic shortest path analysis of Section 7.2.

**Assumption 7.4.1:** One of the states, by convention state  $n$ , is such that for some integer  $m > 0$ , and for all initial states and all policies,  $n$  is visited with positive probability at least once within the first  $m$  stages.

Assumption 7.4.1 can be shown to be equivalent to the assumption that the special state  $n$  is recurrent in the Markov chain corresponding to each stationary policy (see Appendix D for the definition of a recurrent state, and Exercise 2.3 of Chapter 2 in Vol. II for a proof of this equivalence).

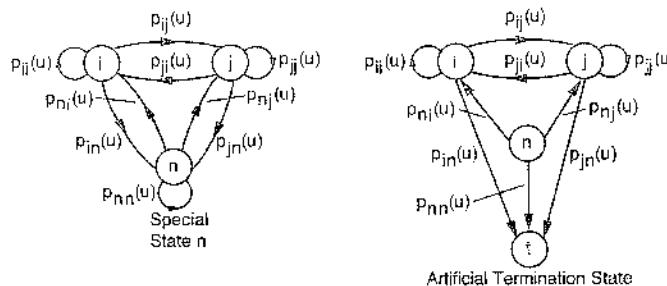
Under Assumption 7.4.1 we will make an important connection of the average cost problem with an associated stochastic shortest path problem. To motivate this connection, consider a sequence of generated states, and divide it into "independent" cycles marked by successive visits to the state  $n$ . The first cycle includes the transitions from the initial state to the first visit to state  $n$ , and the  $k$ th cycle,  $k = 2, 3, \dots$ , includes the transitions from the  $(k-1)$ th to the  $k$ th visit to state  $n$ . Each of the cycles can be viewed as a state trajectory of a corresponding stochastic shortest path problem with the termination state being essentially  $n$ .

More precisely, this problem is obtained by leaving unchanged all transition probabilities  $p_{ij}(u)$  for  $j \neq n$ , by setting all transition probabilities  $p_{in}(u)$  to 0, and by introducing an artificial termination state  $t$  to which we move from each state  $i$  with probability  $p_{in}(u)$ ; see Fig. 7.4.1. Note that Assumption 7.4.1 is equivalent to the Assumption 7.2.1 of Section 7.2 under which the results of Section 7.2 on stochastic shortest path problems were shown.

We have specified the probabilistic structure of the stochastic shortest path problem so that its state trajectories replicate the state trajectories of a single cycle of the average cost problem. We will next argue that if we fix the expected stage cost incurred at state  $i$  to be

$$g(i, u) - \lambda^*,$$

where  $\lambda^*$  is the optimal average cost per stage starting from the special state  $n$ , then the associated stochastic shortest path problem becomes essentially equivalent to the original average cost per stage problem. Furthermore, Bellman's equation for the associated stochastic shortest path problem can



**Figure 7.4.1** Transition probabilities for an average cost problem and its associated stochastic shortest path problem. The latter problem is obtained by introducing, in addition to  $1, \dots, n$ , an artificial termination state  $t$ . The corresponding transition probabilities are obtained from the transition probabilities of the original average cost problem as follows: the probabilities of transition from the states  $i \neq t$  to state  $t$  are set equal to  $p_{in}(u)$ , the probabilities of transition from all states to state  $n$  are set to 0, and all other transition probabilities are left unchanged.

be viewed as Bellman's equation for the original average cost per stage problem.

For a heuristic argument of why this is so, note that under all stationary policies there will be an infinite number of cycles marked by successive visits to state  $n$ . From this, it can be conjectured (and it can also be shown, as will be seen later) that the average cost problem is equivalent to a *minimum cycle cost problem*. This is the problem of finding a stationary policy  $\mu$  that minimizes the average cycle cost

$$\frac{C_{nn}(\mu)}{N_{nn}(\mu)},$$

where for a fixed  $\mu$ ,

$C_{nn}(\mu)$ : expected cost starting from  $n$  up to the first return to  $n$ ,

$N_{nn}(\mu)$ : expected number of stages to return to  $n$  starting from  $n$ .

The intuitive idea here is that the ratio  $C_{nn}(\mu)/N_{nn}(\mu)$  is equal to the average cost of  $\mu$ ,† so the optimal average cost  $\lambda^*$  is equal to the optimal cycle cost. Therefore, we have

$$C_{nn}(\mu) - N_{nn}(\mu)\lambda^* \geq 0, \quad \text{for all } \mu, \quad (7.27)$$

† For a heuristic argument, let  $\lambda_\mu$  be the average cost per stage corresponding to a stationary policy  $\mu$ , and consider a trajectory of the system under  $\mu$ , starting from state  $n$ . If  $C_1, C_2, \dots, C_m$  are the costs incurred in the first  $m$  cycles, and

with equality holding if  $\mu$  is optimal. Thus, to attain an optimal  $\mu$ , we must minimize over  $\mu$  the expression  $C_{nn}(\mu) - N_{nn}(\mu)\lambda^*$ , which is the expected cost of  $\mu$  starting from  $n$  in the associated stochastic shortest path problem with stage costs

$$g(i, u) - \lambda^*, \quad i = 1, \dots, n.$$

Let us denote by  $h^*(i)$  the optimal cost of this stochastic shortest path problem when starting at the nontermination states  $i = 1, \dots, n$ . Then by Prop. 7.2.1(b),  $h^*(1), \dots, h^*(n)$  solve uniquely the corresponding Bellman's equation, which has the form

$$h^*(i) = \min_{u \in U(i)} \left[ g(i, u) - \lambda^* + \sum_{j=1}^{n-1} p_{ij}(u)h^*(j) \right], \quad i = 1, \dots, n, \quad (7.28)$$

since in the stochastic shortest path problem, the transition probability from  $i$  to  $j \neq n$  is  $p_{ij}(u)$  and the transition probability from  $i$  to  $n$  is zero under all  $u$ . If  $\mu^*$  is a stationary policy that minimizes the cycle cost, then this policy must satisfy,

$$C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0,$$

and from Eq. (7.27), this policy must also be optimal for the associated stochastic shortest path problem. Thus, we must have

$$h^*(n) = C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0.$$

By using this equation, we can now write Bellman's equation (7.28) as

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left[ g(i, u) - \sum_{j=1}^n p_{ij}(u)h^*(j) \right], \quad i = 1, \dots, n. \quad (7.29)$$

Equation (7.29), which is really Bellman's equation for the associated stochastic shortest path problem, will be viewed as Bellman's equation for the average cost per stage problem. The preceding argument indicates that this equation has a unique solution as long as we impose the constraint  $h^*(n) = 0$ . Furthermore, by minimization of its right-hand side we should obtain an optimal stationary policy. We will now prove these facts formally.

$N_1, N_2, \dots, N_m$  are the corresponding numbers of stages of these cycles, we have

$$\lambda_\mu = \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m C_k}{\sum_{k=1}^m N_k} = \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m C_k}{m} \cdot \lim_{m \rightarrow \infty} \frac{m}{\sum_{k=1}^m N_k} = C_{nn}(\mu) \cdot \frac{1}{N_{nn}(\mu)}$$

(with probability one).

## Bellman's Equation

The following proposition provides the main results regarding Bellman's equation.

**Proposition 7.4.1** Under Assumption 7.4.1 the following hold for the average cost per stage problem:

- (a) The optimal average cost  $\lambda^*$  is the same for all initial states and together with some vector  $h^* = \{h^*(1), \dots, h^*(n)\}$  satisfies Bellman's equation

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h^*(j) \right], \quad i = 1, \dots, n. \quad (7.30)$$

Furthermore, if  $\mu(i)$  attains the minimum in the above equation for all  $i$ , the stationary policy  $\mu$  is optimal. In addition, out of all vectors  $h^*$  satisfying this equation, there is a unique vector for which  $h^*(n) = 0$ .

- (b) If a scalar  $\lambda$  and a vector  $h = \{h(1), \dots, h(n)\}$  satisfy Bellman's equation, then  $\lambda$  is the average optimal cost per stage for each initial state.
- (c) Given a stationary policy  $\mu$  with corresponding average cost per stage  $\lambda_\mu$ , there is a unique vector  $h_\mu = \{h_\mu(1), \dots, h_\mu(n)\}$  such that  $h_\mu(n) = 0$  and

$$\lambda_\mu + h_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h_\mu(j), \quad i = 1, \dots, n.$$

**Proof:** (a) Let us denote

$$\tilde{\lambda} = \min_{\mu} \frac{C_{nn}(\mu)}{N_{nn}(\mu)}, \quad (7.31)$$

where  $C_{nn}(\mu)$  and  $N_{nn}(\mu)$  have been defined earlier, and the minimum is taken over the finite set of all stationary policies. Note that  $C_{nn}(\mu)$  and  $N_{nn}(\mu)$  are finite in view of Assumption 7.4.1 and the results of Section 7.2. Then we have

$$C_{nn}(\mu) - N_{nn}(\mu)\tilde{\lambda} \geq 0,$$

with equality holding for all  $\mu$  that attain the minimum in Eq. (7.31). Consider the associated stochastic shortest path problem when the expected

stage cost incurred at state  $i$  is

$$g(i, u) - \tilde{\lambda}.$$

Then by Prop. 7.2.1(b), the costs  $h^*(1), \dots, h^*(n)$  solve uniquely the corresponding Bellman's equation

$$h^*(i) = \min_{u \in U(i)} \left[ g(i, u) - \tilde{\lambda} + \sum_{j=1}^{n-1} p_{ij}(u)h^*(j) \right], \quad (7.32)$$

since the transition probability from  $i$  to  $n$  is zero in the associated stochastic shortest path problem. An optimal stationary policy must minimize the cost  $C_{nn}(\mu) - N_{nn}(\mu)\tilde{\lambda}$  and reduce it to zero [in view of Eq. (7.31)], so we see that

$$h^*(n) = 0. \quad (7.33)$$

Thus, Eq. (7.32) is written as

$$\tilde{\lambda} + h^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h^*(j) \right], \quad i = 1, \dots, n. \quad (7.34)$$

We will show that this relation implies that  $\tilde{\lambda} = \lambda^*$ .

Indeed, let  $\pi = \{\mu_0, \mu_1, \dots\}$  be any admissible policy, let  $N$  be a positive integer, and for all  $k = 0, \dots, N-1$ , define  $J_k(i)$  using the following recursion

$$\begin{aligned} J_k(i) &= h^*(i), & i = 1, \dots, n, \\ J_{k+1}(i) &= g(i, \mu_{N-k-1}(i)) + \sum_{j=1}^n p_{ij}(\mu_{N-k-1}(i))J_k(j), & i = 1, \dots, n. \end{aligned} \quad (7.35)$$

Note that  $J_N(i)$  is the  $N$ -stage cost of  $\pi$  when the starting state is  $i$  and the terminal cost function is  $h^*$ . From Eq. (7.34), we have

$$\tilde{\lambda} + h^*(i) \leq g(i, \mu_{N-1}(i)) + \sum_{j=1}^n p_{ij}(\mu_{N-1}(i))h^*(j), \quad i = 1, \dots, n,$$

or equivalently, using Eq. (7.35) for  $k = 0$  and the definition of  $J_0$ ,

$$\tilde{\lambda} + J_0(i) \leq J_1(i), \quad i = 1, \dots, n.$$

Using this relation, we have

$$\begin{aligned} g(i, \mu_{N-2}(i)) + \tilde{\lambda} + \sum_{j=1}^n p_{ij}(\mu_{N-2}(i))J_0(j) \\ \leq g(i, \mu_{N-2}(i)) + \sum_{j=1}^n p_{ij}(\mu_{N-2}(i))J_1(j), \quad i = 1, \dots, n. \end{aligned}$$

By Eq. (7.34) and the definition  $J_0(j) = h^*(j)$ , the left-hand side of the above inequality is no less than  $2\tilde{\lambda} + h^*(i)$ , while by Eq. (7.35), the right-hand side is equal to  $J_2(i)$ . We thus obtain

$$2\tilde{\lambda} + h^*(i) \leq J_2(i), \quad i = 1, \dots, n.$$

By repeating this argument several times, we obtain

$$k\tilde{\lambda} + h^*(i) \leq J_k(i), \quad k = 0, \dots, N, \quad i = 1, \dots, n,$$

and in particular, for  $k = N$ ,

$$\tilde{\lambda} + \frac{h^*(i)}{N} \leq \frac{1}{N} J_N(i), \quad i = 1, \dots, n. \quad (7.36)$$

Furthermore, equality holds in the above relation if  $\mu_k(i)$  attains the minimum in Eq. (7.34) for all  $i$  and  $k$ .

Let us now take the limit as  $N \rightarrow \infty$  in Eq. (7.36). The left-hand side tends to  $\tilde{\lambda}$ . We claim that the right-hand side tends to  $J_\pi(i)$ , the average cost per stage of  $\pi$  starting at state  $i$ . The reason is that from the definition (7.35),  $J_N(i)$  is the  $N$ -stage cost of  $\pi$  starting at  $i$ , when the terminal cost function is  $h^*$ ; when we take the limit of  $(1/N)J_N(i)$ , the dependence on the terminal cost function  $h^*$  disappears. Thus, by taking the limit as  $N \rightarrow \infty$  in Eq. (7.36), we obtain

$$\tilde{\lambda} \leq J_\pi(i), \quad i = 1, \dots, n,$$

for all admissible  $\pi$ , with equality if  $\pi$  is a stationary policy  $\mu$  such that  $\mu(i)$  attains the minimum in Eq. (7.34) for all  $i$  and  $k$ . It follows that

$$\tilde{\lambda} = \min_{\pi} J_\pi(i) = \lambda^*, \quad i = 1, \dots, n,$$

and from Eq. (7.34), we obtain the desired Eq. (7.30).

Finally, Eqs. (7.33) and (7.34) are equivalent to Bellman's equation (7.32) for the associated stochastic shortest path problem. Since the solution to the latter equation is unique, the same is true for the solution of Eqs. (7.33) and (7.34).

(b) The proof of this part is obtained by using the argument of the proof of part (a) following Eq. (7.34).

(c) The proof of this part is obtained by specializing part (a) to the case where the constraint set at each state  $i$  is  $U(i) = \{\mu(i)\}$ . Q.E.D.

An examination of the preceding proof shows that the unique vector  $h^*$  in Bellman's equation (7.30) for which  $h^*(n) = 0$  is the optimal cost vector for the associated stochastic shortest path problem when the expected stage cost at state  $i$  is

$$g(i, u) - \lambda^*,$$

[cf. Eq. (7.32)]. Consequently,  $h^*(i)$  has the interpretation of a *relative or differential cost*; it is the minimum of the difference between the expected cost to reach  $n$  from  $i$  for the first time and the cost that would be incurred if the cost per stage was the average  $\lambda^*$ . We note that the relation between optimal policies of the stochastic shortest path and the average cost problems is clarified in Exercise 7.15.

We finally mention that Prop. 7.4.1 can be shown under considerably weaker conditions (see Section 4.2 of Vol. II). In particular, Prop. 7.4.1 can be shown assuming that all stationary policies have a single recurrent class, even if their corresponding recurrent classes do not have state  $n$  in common. The proof, however, requires a more sophisticated use of the connection with an associated stochastic shortest path problem. Proposition 7.4.1 can also be shown assuming that for every pair of states  $i$  and  $j$ , there exists a stationary policy under which there is positive probability of reaching  $j$  starting from  $i$ . In this case, however, an associated stochastic shortest path problem cannot be defined and the corresponding connection with the average cost per stage problem cannot be made. The analysis of Chapter 4 of Vol. II relies on another connection that exists between the average cost per stage problem and the discounted cost problem, but to establish this connection and to fully explore its ramifications, a much more sophisticated analysis is required.

#### Example 7.4.1

Consider the average cost version of the manufacturer's problem of Example 7.3.2. Here, state 0 plays the role of the special state  $n$  in Assumption 7.4.1. Bellman's equation takes the form

$$\lambda^* + h^*(i) = \min [K + (1-p)h^*(0) + ph^*(1), ci + (1-p)h^*(i) + ph^*(i+1)], \quad (7.37)$$

for the states  $i = 0, 1, \dots, n-1$ , and takes the form

$$\lambda^* + h^*(n) = K + (1-p)h^*(0) + ph^*(1)$$

for state  $n$ . The first expression within brackets in Eq. (7.37) corresponds to processing the  $i$  unfilled orders, while the second expression corresponds to leaving the orders unfilled for one more period. The optimal policy is to process  $i$  unfilled orders if

$$K + (1-p)h^*(0) + ph^*(1) \leq ci + (1-p)h^*(i) + ph^*(i+1).$$

If we view  $h^*(i)$ ,  $i = 1, \dots, n$ , as differential costs associated with an optimal policy, it is intuitively clear that  $h^*(i)$  is monotonically nondecreasing with  $i$  [this can also be proved by interpreting  $h^*(i)$  as optimal costs-to-go for the associate stochastic shortest path problem, or by using analysis based on the theory presented in Vol. II, Section 4.2]. As in Example 7.3.2, the monotonicity property of  $h^*(i)$  implies that a threshold policy is optimal.

## Value Iteration

The most natural version of the value iteration method for the average cost problem is simply to select arbitrarily a terminal cost function, say  $J_0$ , and to generate successively the corresponding optimal  $k$ -stage costs  $J_k(i)$ ,  $k = 1, 2, \dots$ . This can be done by executing the DP algorithm starting with  $J_0$ , that is, by using the recursion

$$J_{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad i = 1, \dots, n. \quad (7.38)$$

It is natural to expect that the ratios  $J_k(i)/k$  should converge to the optimal average cost per stage  $\lambda^*$  as  $k \rightarrow \infty$ , i.e.,

$$\lim_{k \rightarrow \infty} \frac{J_k(i)}{k} = \lambda^*.$$

To show this, let us define the recursion

$$J_{k+1}^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k^*(j) \right], \quad i = 1, \dots, n,$$

with the initial condition

$$J_0^*(i) = h^*(i), \quad i = 1, \dots, n,$$

where  $h^*$  is a differential cost vector satisfying Bellman's equation

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h^*(j) \right], \quad i = 1, \dots, n. \quad (7.39)$$

Using this equation, it can be shown by induction that for all  $k$  we have

$$J_k^*(i) = k\lambda^* + h^*(i), \quad i = 1, \dots, n.$$

On the other hand, it can be seen that for all  $k$ ,

$$|J_k(i) - J_k^*(i)| \leq \max_{j=1, \dots, n} |J_0(j) - h^*(j)|, \quad i = 1, \dots, n.$$

The reason is that  $J_k(i)$  and  $J_k^*(i)$  are optimal costs for two  $k$ -stage problems that differ only in the corresponding terminal cost functions, which are  $J_0$  and  $h^*$ , respectively. From the preceding two equations, we see that for all  $k$ ,

$$|J_k(i) - k\lambda^*| \leq \max_{j=1, \dots, n} |J_0(j) - h^*(j)| + \max_{j=1, \dots, n} |h^*(j)|, \quad i = 1, \dots, n.$$

so that  $J_k(i)/k$  converges to  $\lambda^*$  at the rate of a constant divided by  $k$ . Note that the above proof shows that  $J_k(i)/k$  converges to  $\lambda^*$  under any conditions that guarantee that Bellman's equation (7.39) holds for some vector  $h^*$ .

The value iteration method just described is simple and straightforward, but has two drawbacks. First, since typically some of the components of  $J_k$  diverge to  $\infty$  or  $-\infty$ , direct calculation of  $\lim_{k \rightarrow \infty} J_k(i)/k$  is numerically cumbersome. Second, this method will not provide us with a corresponding differential cost vector  $h^*$ . We can bypass both difficulties by subtracting the same constant from all components of the vector  $J_k$ , so that the difference, call it  $h_k$ , remains bounded. In particular, we can consider the algorithm

$$h_k(i) = J_k(i) - J_k(s), \quad i = 1, \dots, n. \quad (7.40)$$

where  $s$  is some fixed state. By using Eq. (7.38), we then obtain

$$\begin{aligned} h_{k+1}(i) &= J_{k+1}(i) - J_{k+1}(s) \\ &= \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right] \\ &\quad - \min_{u \in U(s)} \left[ g(s, u) + \sum_{j=1}^n p_{sj}(u) J_k(j) \right], \end{aligned}$$

from which in view of the relation  $h_k(j) = J_k(j) - J_k(s)$ , we have

$$\begin{aligned} h_{k+1}(i) &= \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h_k(j) \right] \\ &\quad - \min_{u \in U(s)} \left[ g(s, u) + \sum_{j=1}^n p_{sj}(u) h_k(j) \right], \quad i = 1, \dots, n. \end{aligned} \quad (7.41)$$

The above algorithm, known as *relative value iteration*, is mathematically equivalent to the value iteration method (7.38) that generates  $J_k(i)$ . The iterates generated by the two methods merely differ by a constant [cf. Eq. (7.40)], and the minimization problems involved in the corresponding iterations of the two methods are mathematically equivalent. However, under Assumption 7.4.1, it can be shown that the iterates  $h_k(i)$  generated by the relative value iteration method are bounded, while this is typically not true for the value iteration method.

It can be seen that if the relative value iteration (7.41) converges to some vector  $h$ , then we have

$$\lambda + h(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h(j) \right],$$

where

$$\lambda = \min_{u \in U(s)} \left[ g(s, u) + \sum_{j=1}^n p_{sj}(u) h(j) \right].$$

By Prop. 7.4.1(b), this implies that  $\lambda$  is the optimal average cost per stage for all initial states, and  $h$  is an associated differential cost vector. Unfortunately, the convergence of the relative value iteration is not guaranteed under Assumption 7.4.1 (see Exercise 7.14 for a counterexample). A stronger assumption is required. It turns out, however, that there is a simple variant of the relative value iteration for which convergence is guaranteed under Assumption 7.4.1. This variant is given by

$$\begin{aligned} h_{k+1}(i) &= (1 - \tau)h_k(i) + \min_{u \in U(i)} \left[ g(i, u) + \tau \sum_{j=1}^n p_{ij}(u) h_k(j) \right] \\ &= \min_{u \in U(s)} \left[ g(s, u) + \tau \sum_{j=1}^n p_{sj}(u) h_k(j) \right], \quad i = 1, \dots, n, \end{aligned} \quad (7.42)$$

where  $\tau$  is a scalar such that  $0 < \tau < 1$ . Note that for  $\tau = 1$ , we obtain the relative value iteration (7.41). The convergence proof of this algorithm is somewhat complicated. It can be found in Section 4.3 of Vol. II.

### Policy Iteration

It is possible to use a policy iteration algorithm for the average cost problem. This algorithm operates similar to the policy iteration algorithms of the preceding sections: given a stationary policy, we obtain an improved policy by means of a minimization process, and continue until no further improvement is possible. In particular, at the typical step of the algorithm, we have a stationary policy  $\mu^k$ . We then perform a *policy evaluation* step; that is, we obtain corresponding average and differential costs  $\lambda^k$  and  $h^k(i)$  satisfying

$$\begin{aligned} \lambda^k + h^k(i) &= g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i)) h^k(j), \quad i = 1, \dots, n, \\ h^k(n) &= 0. \end{aligned}$$

We subsequently perform a *policy improvement* step; that is, we find a stationary policy  $\mu^{k+1}$ , where for all  $i$ ,  $\mu^{k+1}(i)$  is such that

$$\begin{aligned} g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i)) h^k(j) \\ = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h^k(j) \right]. \end{aligned}$$

If  $\lambda^{k+1} = \lambda^k$  and  $h^{k+1}(i) = h^k(i)$  for all  $i$ , the algorithm terminates; otherwise, the process is repeated with  $\mu^{k+1}$  replacing  $\mu^k$ .

To prove that the policy iteration algorithm terminates, it is sufficient that each iteration makes some irreversible progress towards optimality, since there are finitely many stationary policies. The type of irreversible progress that we can demonstrate is described in the following proposition, which also shows that an optimal policy is obtained upon termination.

**Proposition 7.4.2** Under Assumption 7.4.1, in the policy iteration algorithm, for each  $k$  we either have

$$\lambda^{k+1} < \lambda^k$$

or else we have

$$\lambda^{k+1} = \lambda^k, \quad h^{k+1}(i) \leq h^k(i), \quad i = 1, \dots, n.$$

Furthermore, the algorithm terminates, and the policies  $\mu^k$  and  $\mu^{k+1}$  obtained upon termination are optimal.

**Proof:** To simplify notation, denote  $\mu^k = \mu$ ,  $\mu^{k+1} = \bar{\mu}$ ,  $\lambda^k = \lambda$ ,  $\lambda^{k+1} = \bar{\lambda}$ ,  $h^k(i) = h(i)$ ,  $h^{k+1}(i) = \bar{h}(i)$ . Define for  $N = 1, 2, \dots$

$$h_N(i) = g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i)) h_{N-1}(j), \quad i = 1, \dots, n,$$

where

$$h_0(i) = h(i), \quad i = 1, \dots, n.$$

Note that  $h_N(i)$  is the  $N$ -stage cost of policy  $\bar{\mu}$  starting from  $i$  when the terminal cost function is  $h$ . Thus we have

$$\bar{\lambda} = J_{\bar{\mu}}(i) = \lim_{N \rightarrow \infty} \frac{1}{N} h_N(i), \quad i = 1, \dots, n, \quad (7.43)$$

since the contribution of the terminal cost to  $(1/N)h_N(i)$  vanishes when  $N \rightarrow \infty$ . By the definition of  $\bar{\mu}$  and Prop. 7.4.1(c), we have for all  $i$

$$\begin{aligned} h_1(i) &= g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i)) h_0(j) \\ &\leq g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) h_0(j) \\ &= \lambda + h_0(i). \end{aligned}$$

From the above equation, we also obtain

$$\begin{aligned} h_2(i) &= g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i))h_1(j) \\ &\leq g(i, \bar{\mu}(i)) - \sum_{j=1}^n p_{ij}(\bar{\mu}(i))(\lambda + h_0(j)) \\ &= \lambda + g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i))h_0(j) \\ &\leq \lambda + g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h_0(j) \\ &= 2\lambda + h_0(i), \end{aligned}$$

and by proceeding similarly, we see that for all  $i$  and  $N$  we have

$$h_N(i) \leq N\lambda + h_0(i).$$

Thus,

$$\frac{1}{N}h_N(i) \leq \lambda + \frac{1}{N}h_0(i),$$

and by taking the limit as  $N \rightarrow \infty$  and using Eq. (7.43), we obtain  $\bar{\lambda} \leq \lambda$ .

If  $\bar{\lambda} = \lambda$ , then it is seen that the iteration that produces  $\mu^{k+1}$  is a policy improvement step for the associated stochastic shortest path problem with cost per stage

$$g(i, u) - \lambda.$$

Furthermore,  $h(i)$  and  $\bar{h}(i)$  are the optimal costs starting from  $i$  and corresponding to  $\mu$  and  $\bar{\mu}$ , respectively, in this associated stochastic shortest path problem. Thus, by Prop. 7.2.2, we must have  $\bar{h}(i) \leq h(i)$  for all  $i$ .

In view of the improvement properties just shown, no policy can be repeated without termination of the algorithm. Since there are only a finite number of policies, it follows that the algorithm will terminate. Let us now show that when the algorithm terminates with  $\bar{\lambda} = \lambda$  and  $\bar{h}(i) = h(i)$  for all  $i$ , the policies  $\bar{\mu}$  and  $\mu$  are optimal. Indeed, upon termination we have for all  $i$

$$\begin{aligned} \lambda + h(i) &= \bar{\lambda} + \bar{h}(i) \\ &= g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i))\bar{h}(j) \\ &= g(i, \bar{\mu}(i)) + \sum_{j=1}^n p_{ij}(\bar{\mu}(i))h(j) \\ &= \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h(j) \right]. \end{aligned}$$

Thus  $\lambda$  and  $h$  satisfy Bellman's equation, and by Prop. 7.4.1(b),  $\lambda$  must be equal to the optimal average cost. Furthermore,  $\bar{\mu}(i)$  attains the minimum in the right-hand side of Bellman's equation, so by Prop. 7.4.1(a),  $\bar{\mu}$  is optimal. Since we also have for all  $i$

$$\lambda + h(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h(j),$$

the same is true for  $\mu$ . Q.E.D.

We note that policy iteration can be shown to terminate with an optimal stationary policy under less restrictive conditions than Assumption 7.4.1 (see Vol. II, Section 4.3).

## 7.5 SEMI-MARKOV PROBLEMS

We have considered so far problems where the cost per stage does not depend on the time required for transition from one state to the next. Such problems have a natural discrete-time representation. On the other hand, there are situations where controls are applied at discrete times but cost is continuously accumulated. Furthermore, the time between successive control choices is variable; it may be random or it may depend on the current state and the choice of control. For example, in queuing systems state transitions correspond to arrivals or departures of customers, and the corresponding times of transition are random. In this section, we discuss continuous-time, infinite horizon problems with a finite number of states. We will provide a fairly straightforward extension of our earlier infinite horizon analysis for discrete-time problems.

We assume that there are  $n$  states, denoted by  $1, \dots, n$ , and that state transitions and control selections take place at discrete times, but the length of the time interval from one transition to the next is random. The state and control at any time  $t$  are denoted by  $x(t)$  and  $u(t)$ , respectively, and stay constant between transitions. We use the following notation:

$t_k$ : The time of occurrence of the  $k$ th transition. By convention, we denote  $t_0 = 0$ .

$x_k = x(t_k)$ : We have  $x(t) = x_k$  for  $t_k \leq t < t_{k+1}$ .

$u_k = u(t_k)$ : We have  $u(t) = u_k$  for  $t_k \leq t < t_{k+1}$ .

In place of transition probabilities, we have *transition distributions*  $Q_{ij}(\tau, u)$ , which for a given pair  $(i, u)$ , specify the joint distribution of the transition interval and the next state:

$$Q_{ij}(\tau, u) = P\{t_{k+1} - t_k \leq \tau, x_{k+1} = j \mid x_k = i, u_k = u\}.$$

Note that the transition distributions specify the ordinary transition probabilities via

$$p_{ij}(u) = P\{x_{k+1} = j \mid x_k = i, u_k = u\} = \lim_{\tau \rightarrow \infty} Q_{ij}(\tau, u).$$

Note also that the conditional cumulative distribution function (CDF) of  $\tau$  given  $i, j, u$  is

$$F_{\tau|t_k+1}^{(i,j,u)} - t_k \leq \tau \mid x_k = i, x_{k+1} = j, u_k = u\} = \frac{Q_{ij}(\tau, u)}{p_{ij}(u)} \quad (7.44)$$

[assuming that  $p_{ij}(u) > 0$ ]. Thus,  $Q_{ij}(\tau, u)$  can be viewed as a "scaled CDF", i.e., a CDF multiplied by  $p_{ij}(u)$  (see Fig. 7.4.2).

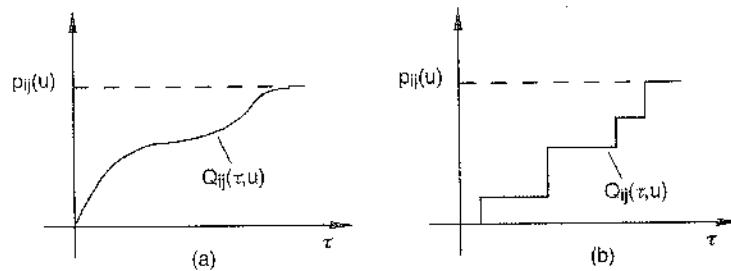


Figure 7.4.2 Illustration of the transition distributions  $Q_{ij}(\tau, u)$  and the conditional CDF of  $\tau$ . Figures (a) and (b) correspond to the cases where  $\tau$  is a continuous and a discrete random variable, respectively.

An advantage of working with transition distributions  $Q_{ij}(\tau, u)$  is that they can be used to model discrete, continuous, and mixed distributions for the transition time  $\tau$ . Generally, expected values of functions of  $\tau$  can be written as integrals involving the differential of  $Q_{ij}$  with respect to  $\tau$ , denoted  $dQ_{ij}(\tau, u)$ . For example, the conditional expected value of  $\tau$  given  $i, j$ , and  $u$  is written using the conditional CDF (7.44) as

$$E\{\tau \mid i, j, u\} = \int_0^\infty \tau \frac{dQ_{ij}(\tau, u)}{p_{ij}(u)}. \quad (7.45)$$

If  $Q_{ij}(\tau, u)$  is continuous and piecewise differentiable with respect to  $\tau$ , its partial derivative

$$q_{ij}(\tau, u) = \frac{dQ_{ij}(\tau, u)}{d\tau}$$

can be viewed as a "scaled" density function for  $\tau$ . Then,  $dQ_{ij}(\tau, u)$  may be replaced by  $q_{ij}(\tau, u)d\tau$ , and expected values of functions of  $\tau$  can be written in terms of  $q_{ij}(\tau, u)$ . For example, Eq. (7.45) is written as

$$E\{\tau \mid i, j, u\} = \int_0^\infty \tau \frac{q_{ij}(\tau, u)}{p_{ij}(u)} d\tau.$$

If  $Q_{ij}(\tau, u)$  is discontinuous and "staircase-like," then  $\tau$  is a discrete random variable, and expected values of functions of  $\tau$  can be written as summations.

We will assume that for each state  $i$  and control  $u \in U(i)$ , the expected transition time, denoted  $\bar{\tau}_i(u)$ , is nonzero and finite:

$$0 < \bar{\tau}_i(u) < \infty. \quad (7.46)$$

In view of Eq. (7.45), this expected transition time is given by

$$\bar{\tau}_i(u) = \sum_{j=1}^n p_{ij}(u) E\{\tau \mid i, j, u\} = \sum_{j=1}^n \int_0^\infty \tau dQ_{ij}(\tau, u).$$

Optimal control problems involving continuous-time Markov chains of the type described above are called *semi-Markov problems*. The reason is that, for a given policy, while at a transition time  $t_k$  the future of the system probabilistically depends only on the current state, at other times it may depend in addition on the time elapsed since the preceding transition. In fact, if we were to allow the control to depend continuously on the time  $t$  (rather than restricting the choice of control to just the transition times  $t_k$ ), we would obtain a problem where there is genuine benefit to the controller for knowing the time elapsed since the preceding transition. We would then have to include this elapsed time as part of the state, and we would obtain a difficult (infinite state space) problem. This type of complication is avoided in our formulation by restricting the control to change only at the transition times  $t_k$ .

We note, however, that there is a special case where the future of the system depends only on its current state at all times, and there is no benefit in allowing the control to depend continuously on the time elapsed since the preceding transition. This is the case where the transition distributions are exponential, of the form

$$Q_{ij}(\tau, u) = p_{ij}(u)(1 - e^{-\nu_i(u)\tau}),$$

where  $p_{ij}(u)$  are transition probabilities, and  $\nu_i(u)$  are given positive scalars, called the *transition rates* at the corresponding states  $i$ . In this case, if the system is in state  $i$  and control  $u$  is applied, the next state will be  $j$  with probability  $p_{ij}(u)$ , and the time interval between the transition to state  $i$  and the transition to the next state is exponentially distributed with parameter  $\nu_i(u)$ ; that is,

$$P\{\text{transition time interval} > \tau \mid i, u\} = e^{-\nu_i(u)\tau}.$$

The exponential distribution has the so called *memoryless property*, which in our context implies that, for any time  $t$  between the transition times  $t_k$

and  $t_{k+1}$ , the additional time  $t_{k+1} - t$  needed to effect the next transition is independent of the time  $t - t_k$  that the system has been in the current state. To see this, use the following generic calculation

$$\begin{aligned} P\{\tau > r_1 + r_2 \mid \tau > r_1\} &= \frac{P\{\tau > r_1 + r_2\}}{P\{\tau > r_1\}} \\ &= \frac{e^{-\nu(r_1+r_2)}}{e^{-\nu r_1}} \\ &= e^{-\nu r_2} \\ &\vdash P\{\tau > r_2\}, \end{aligned}$$

where  $r_1 = t - t_k$ ,  $r_2 = t_{k+1} - t$ , and  $\nu$  is the transition rate. Thus, when the transition distributions are exponential, the state evolves in continuous time as a Markov process, but this need not be true for a more general distribution.

We assume that for given state  $i$  and control  $u \in U(i)$ , the cost that is incurred in a small time interval  $dt$  is  $g(i, u)dt$ . Thus, we may view  $g(i, u)$  as *cost per unit time*. Based on this generic cost structure, we will consider analogs of the discounted and average cost per stage problems of the preceding sections.

### Discounted Problems

Here the cost function has the form

$$\lim_{T \rightarrow \infty} E \left\{ \int_0^T e^{-\beta t} g(x(t), u(t)) dt \right\},$$

where  $\beta$  is a given positive discount parameter. Since the cost per unit time,  $g(x(t), u(t))$ , remains constant between transitions, the expected cost of a single transition from state  $i$  under control  $u$  is given by

$$\begin{aligned} G(i, u) &= E \left\{ \int_0^\tau e^{-\beta t} g(i, u) dt \right\} \\ &= g(i, u) E \left\{ \int_0^\tau e^{-\beta t} dt \right\} \\ &= g(i, u) E_j \left\{ E_\tau \left\{ \int_0^\tau e^{-\beta t} dt \mid j \right\} \right\} \\ &= g(i, u) \sum_{j=1}^n p_{ij}(u) \int_0^\infty \left( \int_0^\tau e^{-\beta t} dt \right) \frac{dQ_{ij}(\tau, u)}{p_{ij}(u)} \end{aligned}$$

or equivalently, since  $\int_0^\tau e^{-\beta t} dt = (1 - e^{-\beta\tau})/\beta$ ,

$$G(i, u) = g(i, u) \sum_{j=1}^n \int_0^\infty \frac{1 - e^{-\beta\tau}}{\beta} dQ_{ij}(\tau, u). \quad (7.47)$$

The cost of an admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$  starting from state  $i$  is given by

$$J_\pi(i) = \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} g(x_k, \mu_k(x_k)) dt \mid x_0 = i \right\}.$$

This cost can be broken down into the sum of the expected cost of the first transition, which is  $G(i, \mu_0(i))$ , plus the expected cost-to-go starting from the next state, discounted by the factor  $e^{-\beta\tau}$ , where  $\tau$  is the (random) time when the first transition occurs:

$$J_\pi(i) = G(i, \mu_0(i)) + E\{e^{-\beta\tau} J_{\pi_1}(j) \mid x_0 = i, u_0 = \mu_0(i)\}. \quad (7.48)$$

The last term in the above equation can be calculated as

$$\begin{aligned} E\{e^{-\beta\tau} J_{\pi_1}(j) \mid x_0 = i, u_0 = \mu_0(i)\} &= E\{E\{e^{-\beta\tau} \mid j\} J_{\pi_1}(j) \mid x_0 = i, u_0 = \mu_0(i)\} \\ &= \sum_{j=1}^n p_{ij}(\mu_0(i)) \left( \int_0^\infty e^{-\beta\tau} \frac{dQ_{ij}(\tau, \mu_0(i))}{p_{ij}(\mu_0(i))} \right) J_{\pi_1}(j) \\ &= \sum_{j=1}^n m_{ij}(\mu_0(i)) J_{\pi_1}(j), \end{aligned}$$

where for any  $u \in U(i)$ ,  $m_{ij}(u)$  is given by

$$m_{ij}(u) = \int_0^\infty e^{-\beta\tau} dQ_{ij}(\tau, u). \quad (7.49)$$

Thus, combining Eqs. (7.47)-(7.49), we see that  $J_\pi(i)$  can be written as

$$J_\pi(i) = G(i, \mu_0(i)) + \sum_{j=1}^n m_{ij}(\mu_0(i)) J_{\pi_1}(j), \quad (7.50)$$

which is similar to the corresponding equation for discounted discrete-time problems [ $m_{ij}(\mu_0(i))$  replaces  $\alpha p_{ij}(\mu_0(i))$ ].

In analogy with the discrete-time case, we may associate Eq. (7.50) with a stochastic shortest path problem involving an artificial termination state  $t$ . Under control  $u$ , from state  $i$  the system moves to state  $j$  with probability  $m_{ij}(u)$  and to the termination state  $t$  with probability

$$1 - \sum_{j=1}^n m_{ij}(u).$$

The assumption of a positive expected transition time [cf. Eq. (7.46)] implies that

$$\sum_{j=1}^n m_{ij}(u) < 1, \quad \text{for all } i, u \in U(i),$$

so that the Assumption 7.2.1, which is required for the validity of our stochastic shortest path analysis of Section 7.2, is satisfied. By using an essentially identical approach to the one of Section 7.3, we can derive analogs of all the discounted cost results of Prop. 7.3.1. In particular, the optimal cost function  $J^*$  is the unique solution of Bellman's equation

$$J^*(i) = \min_{u \in U(i)} \left[ G(i, u) + \sum_{j=1}^n m_{ij}(u) J^*(j) \right].$$

In addition, there are analogs of the computational methods of Section 7.3, including value iteration, policy iteration, and linear programming. What is happening here is that essentially we have the equivalent of a discrete-time discounted problem where the discount factor depends on  $i$  and  $u$ .

We finally note that in some problems, in addition to the cost per unit time  $g$ , there is an extra (instantaneous) one-stage cost  $\hat{g}(i, u)$  that is incurred at the time the control  $u$  is chosen at state  $i$ , and is independent of the length of the transition interval. In this case, Bellman's equation takes the form

$$J^*(i) = \min_{u \in U(i)} \left[ \hat{g}(i, u) + G(i, u) + \sum_{j=1}^n m_{ij}(u) J^*(j) \right], \quad (7.51)$$

and the various computational methods are appropriately adjusted. Another problem variation arises when the cost  $g$  depends on the next state  $j$ . Here, once the system goes into state  $i$ , a control  $u \in U(i)$  is selected, the next state is determined to be  $j$  with probability  $p_{ij}(u)$ , and the cost incurred is  $g(i, u, j)$ . In this case,  $G(i, u)$  should be defined by

$$G(i, u) = \sum_{j=1}^n \int_0^\infty g(i, u, j) \frac{1 - e^{-\beta\tau}}{\beta} dQ_{ij}(\tau, u),$$

[cf. Eq. (7.47)] and the preceding development goes through without modification.

### Example 7.5.1

Consider the manufacturer's problem of Example 7.3.2, with the only difference that the times between the arrivals of successive orders are uniformly distributed in a given interval  $[0, \tau_{\max}]$ , and  $c$  is the cost per unit time of an

unfilled order. Let  $F$  and  $NF$  denote the choices of filling and not filling the orders, respectively. The transition distributions are

$$Q_{ij}(\tau, F) = \begin{cases} \min \left[ 1, \frac{\tau}{\tau_{\max}} \right] & \text{if } j = 1, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$Q_{ij}(\tau, NF) = \begin{cases} \min \left[ 1, \frac{\tau}{\tau_{\max}} \right] & \text{if } j = i + 1, \\ 0 & \text{otherwise.} \end{cases}$$

The one-stage expected cost  $G$  of Eq. (7.47) is given by

$$G(i, F) = 0, \quad G(i, NF) = \gamma ci,$$

where

$$\gamma = \int_0^{\tau_{\max}} \frac{1 - e^{-\beta\tau}}{\beta\tau_{\max}} d\tau.$$

The scalars  $m_{ij}$  of Eq. (7.49) that are nonzero are

$$m_{i1}(F) = m_{i(i+1)}(NF) = \alpha,$$

where

$$\alpha = \int_0^{\tau_{\max}} \frac{e^{-\beta\tau}}{\tau_{\max}} d\tau = \frac{1 - e^{-\beta\tau_{\max}}}{\beta\tau_{\max}}.$$

Bellman's equation has the form [cf. Eq. (7.51)]

$$J(i) = \min \left[ K + \alpha J(1), \gamma ci + \alpha J(i+1) \right], \quad i = 1, 2, \dots$$

As in Example 7.3.2, we can conclude that there exists a threshold  $i^*$  such that it is optimal to fill the orders if and only if their number  $i$  exceeds  $i^*$ .

### Average Cost Problems

A natural cost function for the continuous-time average cost problem would be

$$\lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_0^T g(x(t), u(t)) dt \right\}. \quad (7.52)$$

However, we will use instead the cost function

$$\lim_{N \rightarrow \infty} \frac{1}{E\{t_N\}} E \left\{ \int_0^{t_N} g(x(t), u(t)) dt \right\}, \quad (7.53)$$

where  $t_N$  is the completion time of the  $N$ th transition. This cost function is also reasonable and turns out to be analytically convenient. We note, however, that the cost functions (7.52) and (7.53) are equivalent under the conditions of the subsequent analysis, although a rigorous justification of

this is beyond our scope (see Ross [Ros70], p. 52 and p. 160 for related discussion).

For each pair  $(i, u)$ , we denote by  $G(i, u)$  the one-stage expected cost corresponding to state  $i$  and control  $u$ . We have

$$G(i, u) = g(i, u)\bar{\tau}_i(u),$$

where  $\bar{\tau}_i(u)$  is the expected value of the transition time corresponding to  $(i, u)$ . [If the cost per unit time  $g$  depends on the next state  $j$ , the expected transition cost  $G(i, u)$  should be defined by

$$G(i, u) = \sum_{j=1}^n \int_0^\infty g(i, u, j) \tau dQ_{ij}(\tau, u),$$

and the following analysis and results go through without modification.] The cost function of an admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$  is given by

$$J_\pi(i) = \lim_{N \rightarrow \infty} \frac{1}{E\{t_N \mid x_0 = i, \pi\}} E \left\{ \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} g(x_k, \mu_k(x_k)) dt \mid x_0 = i \right\}.$$

We will see that the character of the solution of the problem is determined by the structure of the *embedded Markov chain*, which is the controlled discrete-time Markov chain whose transition probabilities are

$$p_{ij}(u) = \lim_{\tau \rightarrow \infty} Q_{ij}(\tau, u).$$

In particular, assuming that the embedded Markov chain satisfies Assumption 7.4.1 of Section 7.4, we can show that the costs  $J^*(i)$  are independent of  $i$ .

It turns out that Bellman's equation for average cost semi-Markov problems takes the form

$$h(i) = \min_{u \in U(i)} \left[ G(i, u) - \lambda \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h(j) \right].$$

As a special case, when  $\bar{\tau}_i(u) = 1$  for all  $(i, u)$ , we obtain the corresponding Bellman's equation for discrete-time problems, given in Section 7.4. We motivate the above form of Bellman's equation with the stochastic shortest path argument that we used in Section 7.4. We consider a sequence of generated states, and divide it into cycles marked by successive visits to the special state  $n$ . Each of the cycles can be viewed as a state trajectory of a corresponding stochastic shortest path problem with the termination state being essentially  $n$ , as in Section 7.4.

We next conjecture that the average cost problem is equivalent to the *minimum cycle cost problem* of finding a stationary policy  $\mu$  that minimizes the average cycle cost

$$\frac{C_{nn}(\mu)}{T_{nn}(\mu)},$$

where for a fixed  $\mu$ ,

$C_{nn}(\mu)$  : expected cost starting from  $n$  up to the first return to  $n$ ,

$T_{nn}(\mu)$  : expected time to return to  $n$  starting from  $n$ .

An intuitive conjecture is that the optimal average cost  $\lambda^*$  is equal to the optimal cycle cost, so it satisfies

$$C_{nn}(\mu) - N_{nn}(\mu)\lambda^* \geq 0, \quad \text{for all } \mu, \quad (7.54)$$

with equality holding if  $\mu$  is optimal. Thus, to attain an optimal  $\mu$ , we must minimize over  $\mu$  the expression  $C_{nn}(\mu) - T_{nn}(\mu)\lambda^*$ , which is the expected cost of  $\mu$  starting from  $n$  in the associated stochastic shortest path problem with stage costs

$$G(i, u) - \lambda^* \bar{\tau}_i(\mu(i)), \quad i = 1, \dots, n.$$

Let us denote by  $h^*(i)$  the optimal cost of this stochastic shortest path problem when starting at state  $i$ . Then  $h^*(1), \dots, h^*(n)$  solve uniquely the corresponding Bellman's equation, which has the form

$$h^*(i) = \min_{u \in U(i)} \left[ G(i, u) - \lambda^* \bar{\tau}_i(u) + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) \right], \quad i = 1, \dots, n. \quad (7.55)$$

If  $\mu^*$  is an optimal stationary policy, then this policy must satisfy

$$C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0,$$

and from Eq. (7.54), this policy must also be optimal for the associated stochastic shortest path problem. Thus, we must have

$$h^*(n) = C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0.$$

By using this equation, we can now write Bellman's equation (7.55) as

$$h^*(i) = \min_{u \in U(i)} \left[ G(i, u) - \lambda^* \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h^*(j) \right], \quad i = 1, \dots, n. \quad (7.56)$$

If there is an "instantaneous" one-stage cost  $\hat{g}(i, u)$ , the term  $G(i, u)$  should be replaced by  $\hat{g}(i, u) + G(i, u)$  in this equation.

Given the correct form of Bellman's equation and the connection with the associated stochastic shortest path problem, it is possible to essentially repeat the proof of Prop. 7.4.1 and to obtain analogous results to those for the discrete-time case.

**Example 7.5.2**

Consider the average cost version of the manufacturer's problem of Example 7.5.1. Here we have

$$\bar{\tau}_i(F) = \bar{\tau}_i(NF) = \frac{\bar{\tau}_{\max}}{2},$$

$$G(i, F) = K, \quad G(i, NF) = \frac{ci\bar{\tau}_{\max}}{2},$$

where  $F$  and  $NF$  denote the decisions to fill and not fill the orders, respectively. Bellman's equation (7.56) takes the form

$$h^*(i) = \min \left[ K - \lambda^* \frac{\bar{\tau}_{\max}}{2} + h^*(1), ci \frac{\bar{\tau}_{\max}}{2} - \lambda^* \frac{\bar{\tau}_{\max}}{2} + h^*(i+1) \right].$$

We leave it as an exercise for the reader to show that there exists a threshold  $i^*$  such that it is optimal to fill the orders if and only if  $i$  exceeds  $i^*$ .

**Example 7.5.3 [LiR71]**

Consider a person providing a certain type of service to customers. Potential customers arrive according to a Poisson process with rate  $r$ ; that is, the customer's interarrival times are independent and exponentially distributed with parameter  $r$ . Each customer offers one of  $n$  pairs  $(m_i, T_i)$ ,  $i = 1, \dots, n$ , where  $m_i$  is the amount of money offered for the service and  $T_i$  is the average amount of time that will be required to perform the service. Successive offers are independent and offer  $(m_i, T_i)$  occurs with probability  $p_i$ , where  $\sum_{i=1}^n p_i = 1$ . An offer may be rejected, in which case the customer leaves, or may be accepted in which case all offers that arrive while the customer is being served are lost. The problem is to determine the acceptance-rejection policy that maximizes the service provider's average income per unit time.

Let us denote by  $i$  the state corresponding to the offer  $(m_i, T_i)$ , and let  $A$  and  $R$  denote the accept and reject decision, respectively. We have

$$\bar{\tau}_i(A) = T_i + \frac{1}{r}, \quad \bar{\tau}_i(R) = \frac{1}{r}, \quad G(i, A) = -m_i, \quad G(i, R) = 0,$$

$$p_{ij}(A) = p_{ij}(R) = p_j.$$

Bellman's equation is given by

$$h^*(i) = \min \left[ -m_i - \lambda^* \left( T_i + \frac{1}{r} \right) + \sum_{j=1}^n p_j h^*(j), -\lambda^* \frac{1}{r} + \sum_{j=1}^n p_j h^*(j) \right].$$

It follows that an optimal policy is to

$$\text{accept offer } (i, T_i) \quad \text{if and only if} \quad \frac{m_i}{T_i} \geq -\lambda^*,$$

where  $-\lambda^*$  is the optimal average income per unit time.

**7.6 NOTES, SOURCES, AND EXERCISES**

This chapter is only an introduction to infinite horizon problems. There is an extensive theory for these problems with interesting mathematical and computational content. Volume II provides a comprehensive treatment and gives many references to the literature.

The presentation in this chapter is original in that it uses the stochastic shortest path problem as the starting point for the analysis of the other problems. This line of development not only explains intuitively the connections between the various types of problems, but also leads to new solution methods. For example, an alternative value iteration algorithm for the average cost problem, based on the connection with the stochastic shortest path problem, is given in Bertsekas [Ber98b], and in Section 4.3 of Vol. II. On the other hand, there are also important results for undiscounted and average cost problems that cannot be obtained through the connection with the stochastic shortest path problem. Some of these alternative lines of analysis are pursued in Vol. II.

Semi-Markov problems were introduced by Jewell [Jew63] and were also discussed by Ross [Ros70]. Volume II contains a broader exposition of Semi-Markov problems, and applications to queueing and related systems.

**E X E R C I S E S****7.1**

A tennis player has a Fast serve and a Slow serve, denoted  $F$  and  $S$ , respectively. The probability of  $F$  (or  $S$ ) landing in bounds is  $p_F$  (or  $p_S$ , respectively). The probability of winning the point assuming the serve landed in bounds is  $q_F$  (or  $q_S$ , respectively). We assume that  $p_F < p_S$  and  $q_F > q_S$ . The problem is to find the serve to be used at each possible scoring situation during a single game in order to maximize the probability of winning that game.

- (a) Formulate this as a stochastic shortest path problem, argue that Assumption 7.2.1 of Section 7.2 holds, and write Bellman's equation.
- (b) Computer assignment: Assume that  $q_F = 0.6$ ,  $q_S = 0.4$ , and  $p_S = 0.95$ . Use value iteration to calculate and plot (in increments of 0.05) the probability of the server winning a game with optimal serve selection as a function of  $p_F$ .

## 7.2

A quarterback can choose between running and passing the ball on any given play. The number of yards gained by running is integer and is Poisson distributed with parameter  $\lambda_r$ . A pass is incomplete with probability  $p$ , is intercepted with probability  $q$ , and is completed with probability  $1 - p - q$ . When completed, a pass gains an integer number of yards that is Poisson distributed with parameter  $\lambda_p$ . We assume that the probability of scoring a touchdown on a single play starting  $i$  yards from the goal is equal to the probability of gaining  $i$  yards greater than or equal to  $i$ . We assume also that yardage cannot be lost on any play and that there are no penalties. The ball is turned over to the other team on a fourth down or when an interception occurs.

- (a) Formulate the problem as a stochastic shortest path problem, argue that Assumption 7.2.1 of Section 7.2 holds, and write Bellman's equation.
- (b) Computer assignment: Use value iteration to compute the quarterback's play-selection policy that maximizes the probability of scoring a touchdown on any single drive for  $\lambda_r = 3$ ,  $\lambda_p = 10$ ,  $p = 0.4$ , and  $q = 0.05$ .

## 7.3

A computer manufacturer can be in one of two states. In state 1 his product sells well, while in state 2 his product sells poorly. While in state 1 he can advertise his product in which case the one-stage reward is 4 units, and the transition probabilities are  $p_{11} = 0.8$  and  $p_{12} = 0.2$ . If in state 1, he does not advertise, the reward is 6 units and the transition probabilities are  $p_{11} = p_{12} = 0.5$ . While in state 2, he can do research to improve his product, in which case the one-stage reward is -5 units, and the transition probabilities are  $p_{21} = 0.7$  and  $p_{22} = 0.3$ . If in state 2 he does not do research, the reward is -3, and the transition probabilities are  $p_{21} = 0.4$  and  $p_{22} = 0.6$ . Consider the infinite horizon, discounted version of this problem.

- (a) Show that when the discount factor  $\alpha$  is sufficiently small, the computer manufacturer should follow the "shortsighted" policy of not advertising (not doing research) while in state 1 (state 2). By contrast, when  $\alpha$  is sufficiently close to unity, he should follow the "farsighted" policy of advertising (doing research) while in state 1 (state 2).
- (b) For  $\alpha = 0.9$  calculate the optimal policy using policy iteration.
- (c) For  $\alpha = 0.99$ , use a computer to solve the problem by value iteration, with and without the error bounds (7.23).

## 7.4

An energetic salesman works every day of the week. He can work in only one of two towns  $A$  and  $B$  on each day. For each day he works in town  $A$  (or  $B$ ) his expected reward is  $r_A$  (or  $r_B$ , respectively). The cost for changing towns is  $c$ . Assume that  $c > r_A > r_B$  and that there is a discount factor  $\alpha < 1$ .

- (a) Show that for  $\alpha$  sufficiently small, the optimal policy is to stay in the town he starts in, and that for  $\alpha$  sufficiently close to 1, the optimal policy is to move to town  $A$  (if not starting there) and stay in  $A$  for all subsequent times.
- (b) Solve the problem for  $c = 3$ ,  $r_A = 2$ ,  $r_B = 1$ , and  $\alpha = 0.9$  using policy iteration.
- (c) Use a computer to solve the problem of part (b) by value iteration, with and without the error bounds (7.23).

## 7.5

A person has an umbrella that she takes from home to office and vice versa. There is a probability  $p$  of rain at the time she leaves home or office independently of earlier weather. If the umbrella is in the place where she is and it rains, she takes the umbrella to go to the other place (this involves no cost). If there is no umbrella and it rains, there is a cost  $W$  for getting wet. If the umbrella is in the place where she is but it does not rain, she may take the umbrella to go to the other place (this involves an inconvenience cost  $V$ ) or she may leave the umbrella behind (this involves no cost). Costs are discounted at a factor  $\alpha < 1$ .

- (a) Formulate this as an infinite horizon total cost discounted problem. Hint: Try to use as few states as possible.
- (b) Characterize the optimal policy as best as you can.

## 7.6

For the tennis player's problem (Exercise 7.1), show that it is optimal (regardless of score) to use  $F$  on both serves if

$$(p_F q_F)/(p_S q_S) > 1,$$

to use  $S$  on both serves if

$$(p_F q_F)/(p_S q_S) < 1 + p_F - p_S,$$

and to use  $F$  on the first serve and  $S$  on the second otherwise.

## 7.7

Consider the value iteration method for the Example 7.3.2:

$$\begin{aligned} J_{k+1}(i) &= \min \left[ K + \alpha(1-p)J_k(0) - \alpha p J_k(1), \right. \\ &\quad \left. ci + \alpha(1-p)J_k(i) + \alpha p J_k(i+1) \right], \quad i = 0, 1, \dots, n-1, \\ J_{k+1}(n) &= K + \alpha(1-p)J_k(0) + \alpha p J_k(1), \end{aligned}$$

where  $J_0(i) = 0$  for all  $i$ . Show by induction that  $J_k(i)$  is monotonically nondecreasing in  $i$ .

7.8 

Consider the policy iteration algorithm for the problem of Example 7.3.2.

- Show that if we start the algorithm with a threshold policy, every subsequently generated policy will be a threshold policy. *Note:* This requires a careful argument.
- Carry out the algorithm for the case  $c = 1$ ,  $K = 5$ ,  $n = 10$ ,  $p = 0.5$ ,  $\alpha = 0.9$ , and an initial policy that always processes the unfilled orders.

## 7.9

Solve the average cost version ( $\alpha = 1$ ) of the computer manufacturer's problem by using value iteration and by using policy iteration (Exercise 7.3).

## 7.10

An unemployed worker receives a job offer at each time period, which she may accept or reject. The offered salary takes one of  $n$  possible values  $w^1, \dots, w^n$  with given probabilities, independently of preceding offers. If she accepts the offer, she must keep the job for the rest of her life at the same salary level. If she rejects the offer, she receives unemployment compensation  $c^*$  in the current period and is eligible to accept future offers. Assume that income is discounted by a factor  $\alpha < 1$ .

- Show that there is a threshold  $\bar{w}$  such that it is optimal to accept an offer if and only if its salary is larger than  $\bar{w}$ , and characterize  $\bar{w}$ .
- Consider the variant of the problem where there is a given probability  $p_i$  that the worker will be fired from her job at any one period if her salary is  $w^i$ . Show that the result of part (a) holds in the case where  $p_i$  is the same for all  $i$ . Analyze the case where  $p_i$  depends on  $i$ .

## 7.11

Do part (b) of Exercise 7.10 for the case where income is not discounted and the worker maximizes her average income per period.

7.1.2 

Show that one can always take  $m = n$  in Assumption 7.2.1. *Hint:* For any  $\pi$  and  $i$ , let  $S_k(i)$  be the set of states that are reachable with positive probability from  $i$  under  $\pi$  in  $k$  stages or less. Show that under Assumption 7.2.1, we cannot have  $S_k(i) = S_{k+1}(i)$  while  $t \neq S_k(i)$ .

## 7.13

Show the error bounds (7.17). These bounds constitute a generalization to the stochastic shortest path problem of the bounds (7.23) for the discounted problem, which have a long history, starting with the work of McQueen [McQ66]. *Hint:* Complete the details of the following argument. Let  $\mu^k(i)$  attain the minimum in the value iteration (7.16) for all  $i$ . Then, in vector form, we have

$$J_{k+1} = g_k + P_k J_k,$$

where  $J_k$  and  $g_k$  are the vectors with components  $J_k(i)$ ,  $i = 1, \dots, n$ , and  $g_k(i, \mu^k(i))$ ,  $i = 1, \dots, n$ , respectively, and  $P_k$  is the matrix whose components are the transition probabilities  $p_{ij}(\mu^k(i))$ . Also from Bellman's equation, we have

$$J^* \leq g_k + P_k J^*,$$

where the vector inequality above is meant to hold separately for each component. Let  $e = (1, \dots, 1)'$ . Using the above two relations, we have

$$J^* - J_k \leq J^* - J_{k+1} + \bar{c}_k e \leq P_k(J^* - J_k) + \bar{c}_k e. \quad (7.57)$$

Multiplying this relation with  $P_k$  and adding  $\bar{c}_k e$ , we obtain

$$P_k(J^* - J_k) + \bar{c}_k e \leq P_k^2(J^* - J_k) + \bar{c}_k(I + P_k)e.$$

Similarly continuing, we have for all  $r \geq 1$

$$J^* - J_{k+1} + \bar{c}_k e \leq P_k^r(J^* - J_k) + \bar{c}_k(I + P_k + \dots + P_k^{r-1})e.$$

For  $s = 1, 2, \dots$ , the  $i$ th component of the vector  $P_k^s e$  is equal to the probability  $P\{\omega_s \neq t \mid x_0 = i, \mu^k\}$  that  $t$  has not been reached after  $s$  stages starting from  $i$  and using the stationary policy  $\mu^k$ . Thus, Assumption 7.2.1 implies that  $\lim_{r \rightarrow \infty} P_k^r = 0$ , while we have

$$\lim_{r \rightarrow \infty} (I + P_k + \dots + P_k^{r-1})e = N^k,$$

where  $N^k$  is the vector  $(N^k(1), \dots, N^k(n))'$ . Combining the above two relations, we obtain

$$J^* \leq J_{k+1} + \bar{c}_k(N^k - e),$$

proving the desired upper bound.

The lower bound is proved similarly, by using in place of  $\mu^k$ , an optimal stationary policy  $\mu^*$ . In particular, in place of Eq. (7.57), we can show that

$$J_k - J^* \leq J_{k+1} - J^* - \underline{c}_k e \leq P^*(J_k - J^*) - \underline{c}_k e,$$

where  $P^*$  is the matrix with elements  $p_{ij}(\mu^*(i))$ . We similarly obtain for all  $r \geq 1$

$$J_{k+1} - J^* - \underline{c}_k e \leq (P^*)^r(J_k - J^*) - \underline{c}_k(I + P^* + \dots + (P^*)^{r-1})e,$$

from which  $J_{k+1} + \underline{c}_k(N^* - e) \leq J^*$ , where  $N^*$  is the vector  $(N^*(1), \dots, N^*(n))'$ .

## 7.14

Apply the relative value iteration algorithm (7.41) for the case where there are two states and only one control per state. The transition probabilities are  $p_{11} = \epsilon$ ,  $p_{12} = 1 - \epsilon$ ,  $p_{21} = 1 - \epsilon$ , and  $p_{22} = \epsilon$ , where  $0 \leq \epsilon < 1$ . Show that if  $0 < \epsilon$  the algorithm converges, but if  $\epsilon = 0$ , the algorithm may not converge. Show also that the variation (7.42) converges when  $\epsilon = 0$ .

## 7.15

Consider the average cost problem and its associated stochastic shortest path problem when the expected cost incurred at state  $i$  is  $g(i, u) - \lambda^*$ .

- (a) Use Prop. 7.2.1(d) and Prop. 7.4.1(a) to show that if a stationary policy is optimal for the latter problem it is also optimal for the former.
- (b) Show by example that the reverse of part (a) need not be true.

## 7.16

Consider a problem of operating a machine that can be in any one of  $n$  states, denoted  $1, 2, \dots, n$ . We denote by  $g(i)$  the operating cost per period when the machine is in state  $i$ , and we assume that

$$g(1) \leq g(2) \leq \dots \leq g(n).$$

The implication here is that state  $i$  is better than state  $i + 1$ , and state 1 corresponds to a machine being in the best condition. The transition probabilities during one period of operation satisfy

$$\begin{aligned} p_{i(i+1)} &> 0 & \text{if } i < n, \\ p_{ij} &= 0 & \text{if } j \neq i, j \neq i+1. \end{aligned}$$

We assume that at the start of each period we know the state of the machine and we must choose one of the following two options:

- (1) Let the machine operate one more period in the state it currently is.
- (2) Repair the machine and bring it to the best state 1 at a cost  $R$ .

We assume that the machine, once repaired, is guaranteed to stay in state 1 for one period. In subsequent periods, it may deteriorate to states  $j > 1$ .

- (a) Assume an infinite horizon and a discount factor  $\alpha \in (0, 1)$ , and show that there is an optimal policy which is a threshold policy; that is, it takes the form

replace if and only if  $i \geq i^*$ ,

where  $i^*$  is some integer.

- (b) Show that the policy iteration method, when started with a threshold policy, generates a sequence of threshold policies.

## 7.17

Consider a person providing a certain type of service to customers. The person receives at the beginning of each time period with probability  $p_i$  a proposal by a customer of type  $i$ , where  $i = 1, 2, \dots, n$ , who offers an amount of money  $M_i$ . We assume that  $\sum_{i=1}^n p_i = 1$ . The person may reject the offer, in which case the customer leaves and the person remains idle during that period, or the person may accept the offer in which case the person spends some random amount of time with that customer. In particular, we assume that the probability that the type  $i$  customer will leave after  $k$  periods ( $k = 1, 2, \dots$ ), given that the customer has already stayed with the person for  $k - 1$  periods is a given scalar  $\beta_i \in (0, 1)$ . The problem is to determine an acceptance-rejection policy that maximizes

$$\lim_{N \rightarrow \infty} \frac{1}{N} \{\text{Expected payment over } N \text{ periods}\}.$$

- (a) Formulate the person's problem as an average cost per stage problem, and show that the optimal cost is independent of the initial state.
- (b) Show that there exists a scalar  $\lambda$  and an optimal policy that accepts the offer of a type  $i$  customer if and only if

$$\lambda \leq \frac{M_i}{T_i},$$

where  $T_i$  is the expected time spent with the type  $i$  customer.

## 7.18

A person has an asset to sell for which she receives offers that take one of  $n$  values  $s_j$ ,  $j = 1, \dots, n$ . The times between successive offers are random, identically distributed, and independent of preceding times. Let  $Q_j(\tau)$  be the probability that the time between successive offers is less or equal to  $\tau$  and the next offer is  $s_j$ . Find the offer acceptance policy that maximizes  $E\{\alpha^T s\}$ , where  $T$  is the time of sale,  $s$  is the sale price, and  $\alpha \in (0, 1)$  is a discount factor.

## 7.19

An unemployed worker receives job offers, which she may accept or reject. The times between successive offers are independent and exponentially distributed with parameter  $r$ . The offered salary (per unit time) takes one of  $n$  possible values  $w_i$ ,  $i = 1, \dots, n$ , with given probabilities  $p_i$ , independently of preceding offers. If she accepts an offer at salary  $w_i$ , she keeps the job for a random amount of time that has expected value  $t_i$ . If she rejects the offer, she receives unemployment compensation  $c$  (per unit time) and is eligible to accept future offers. Solve the problem of maximizing the worker's average income per unit time.

## 7.20

Consider a computing system where the interarrival times of the jobs are independent and exponentially distributed with parameter  $r$ . A job may be rejected, in which case the job is lost, or may be accepted in which case all jobs that arrive while the job is being processed are lost. There are  $n$  types of jobs. Each arriving job is of type  $i$  with probability  $p_i$ , independently of earlier jobs, and if processed, is worth a fixed positive benefit  $b_i$  ( $i = 1, \dots, n$ ). Jobs of type  $i$  require an average amount of time  $T_i$  to complete processing. The problem is to determine the acceptance-rejection policy that maximizes the system's average benefit per unit time.

- (a) Argue that the analysis of Example 7.5.3 applies for this problem.
- (b) Calculate the optimal average benefit per unit time  $\lambda^*$  in terms of the given quantities for the case where there are only two job types.
- (c) Suppose that the time to process a job of type  $i$  is exponentially distributed with mean  $T_i$ . Assume further that the system can process up to a given number  $m > 1$  of jobs simultaneously (rather than just one). Formulate the problem as an average benefit per unit time semi-Markov problem, and write Bellman's equation for the case where  $m = 2$ . Why do we need the exponential distribution assumption?

## 7.21

Formulate a semi-Markov version of the stochastic shortest path problem of Section 7.2. The cost function has the form

$$\lim_{T \rightarrow \infty} E \left\{ \int_0^T g(x(t), u(t)) dt \right\},$$

and there is a cost-free and absorbing state. Use the transition distributions  $Q_{ij}$  to formulate an assumption that is analogous to Assumption 7.2.1. Under this assumption, state and justify a result that parallels Prop. 7.2.1.

## 7.22

A treasure hunter has obtained a lease to search a site that contains  $n$  treasures, and wants to find a searching policy that maximizes his expected gain over an infinite number of days. At each day, knowing the current number of treasures not yet found, he may decide to continue searching for more treasures at a cost  $c$  per day, or to permanently stop searching. If he searches on a day when there are  $i$  treasures on the site, he finds  $m \in [0, i]$  treasures with given probability  $p(m | i)$ , where we assume that  $p(0 | i) < 1$  for all  $i \geq 1$ , and that the expected number of treasures found,

$$r(i) = \sum_{m=0}^i mp(m | i),$$

## Sec. 7.6 Notes, Sources, and Exercises

## 453

decreases monotonically with  $i$ . Each found treasure is worth 1 unit.

- (a) Formulate the problem as an infinite horizon DP problem.
- (b) Write Bellman's equation. How do you know that this equation holds and has a unique solution?
- (c) Start policy iteration with the policy that never searches. How many policy iterations does it take to find an optimal policy, and what is that optimal policy?

## 7.23

The latest slot machine model has three arms, labeled 1, 2, and 3. A single play with arm  $i$ , where  $i = 1, 2, 3$ , costs  $c_i$  dollars, and has two possible outcomes: a "win," which occurs with probability  $p_i$ , and a "loss," which occurs with probability  $1 - p_i$ . The slot machine pays you  $m$  dollars each time you complete a sequence of three successive "wins," with each win obtained using a different arm.

- (a) Consider the problem of finding the arm-playing order that minimizes the expected cost if you are restricted to stop at the first time the machine pays you. Formulate this problem as a stochastic shortest path problem where arm-playing orders are identified with stationary policies, and write Bellman's equation for each stationary policy.
- (b) Show that the expected cost of the arm-playing order ABC is

$$\frac{c_A + p_{ABC} + p_{APBCC} - p_{APBPCm}}{1 - p_{APBPC}}.$$

Show that it is optimal to play the arms in order of decreasing  $c_i/(1 - p_i)$ .

- (c) Consider the problem of finding the arm-playing order that minimizes the average expected cost per play, assuming you play infinitely many times. Formulate this problem as an average cost per stage problem, where arm-playing orders are identified with stationary policies, and write Bellman's equation for each stationary policy.
- (d) Show that the expected cost per play of the arm-playing order ABC is

$$\frac{c_A + p_{ACB} + p_{APBCC} - p_{APBPCm}}{1 + p_{APB}}.$$

Is it possible that the optimal playing order is different than the one of part (b)? If this is so, how do you explain it?

## 7.24

A person has a house that he rents at a fixed amount  $R$  per time period. At the beginning of each period  $k$ , the person receives an offer  $w_k$  to sell the house. The amount  $w_k$  takes one of  $m$  given values  $w^1, \dots, w^m$ , with corresponding positive probabilities  $q^1, \dots, q^m$ , independently of preceding offers. The person, at the

beginning of each period, must decide whether to accept the current offer or to decline the offer and continue to rent the house.

Upon selling the house, the sale amount, call it  $w$ , is immediately reinvested in some way so that it yields at time  $k$  a random amount  $y_k w$ , where  $y_k$  takes one of  $s$  given values  $y^1, \dots, y^s$ . The value of  $y_k$  evolves according to a Markov chain with a single recurrent class and given transition probabilities

$$p_{ij} = P(y_{k+1} = y^j | y_k = y^i), \quad i, j = 1, \dots, s.$$

- (a) Suppose that at time  $k$  the house is sold when  $y_k$  is equal to  $y^i$ . Let

$$\bar{y}(i) = \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{t=k}^{k+N-1} y_t \mid y_k = y^i \right\}$$

be the average future yield per unit time. Show that  $\bar{y}(i)$  is equal to a common value  $\bar{y}$ , independent of  $i$ , and derive a Bellman-type equation for this value.

- (b) Suppose that the person's objective is to maximize the average monetary benefit per time period. Argue that an optimal stationary policy is to wait until the maximum possible offer  $\bar{w} = \max\{w^1, \dots, w^s\}$  is received and then sell the house, assuming that  $R/\bar{y} \leq \bar{w}$ . Given this result, discuss whether the average cost formulation is satisfactory for this problem.
- (c) Suppose that the person's objective is to maximize the total discounted monetary benefit over an infinite horizon, with a discount factor  $\alpha < 1$ . Show that for each  $i = 1, \dots, s$ , there is a threshold  $t(i)$  such that it is optimal to sell the house at period  $k$  when  $y_k = y^i$  and the current offer is larger than  $t(i)$ .

### 7.25

You have just bought your first car, which raises the issue of where to park it. At the beginning of each day you may either park it in a garage, which costs  $G$  per day, or on the street for free. However, in the latter case, you run the risk of getting a parking ticket, which costs  $T$ , with probability  $p_j$ , where  $j$  is the number of consecutive days that the car has been parked on the street (e.g., on the first day you park on the street, you have probability  $p_1$  of getting a ticket, on the second successive day you park on the street, you have probability  $p_2$ , etc.). Assume that  $p_j$  is monotonically nondecreasing in  $j$ , and that you may receive at most one ticket per day when parked on the street. Assume also that there exists an integer  $m$  such that  $p_m T > G$ .

- (a) Formulate this as an infinite horizon discounted cost problem with finite state space and write the corresponding Bellman's equation.
- (b) Characterize as best as you can the optimal policy.
- (c) Let  $n$  be the total number of states. Show how to use policy iteration so that it terminates after no more than  $n$  iterations. Hint: Use threshold policies as in Problem 7.8.

- (d) Formulate the infinite horizon average cost version of this problem with finite state space and write the corresponding Bellman's equation. State an assumption under which Bellman's equation holds.

### 7.26

An engineer has invented a better mouse trap and is interested in selling it for the right price. At the beginning of each period, he receives a sale offer that takes one of the values  $s_1, \dots, s_n$  with corresponding probabilities  $p_1, \dots, p_n$ , independently of prior offers. If he accepts the offer he retires from engineering. If he refuses the offer, he may accept subsequent offers but he also runs the risk that a competitor will invent an even better mouse trap, rendering his own unsaleable; this happens with probability  $\beta > 0$  at each time period, independently of earlier time periods. While he is overtaken by the competitor, at each time period, he may choose to retire from engineering, or he may choose to invest an amount  $v \geq 0$ , in which case he has a probability  $\gamma$  to improve his mouse trap, overtake his competitor, and start receiving offers as earlier. The problem is to determine the engineer's strategy to maximize his discounted expected payoff (minus investment cost), assuming a discount factor  $\alpha < 1$ .

- (a) Formulate the problem as an infinite horizon discounted cost problem and write the corresponding Bellman's equation.
- (b) Characterize as best as you can an optimal policy.
- (c) Assume that there is no discount factor. Does the problem make sense as an average cost per stage problem?
- (d) Assume that there is no discount factor and that the investment cost  $v$  is equal to 0. Does the problem make sense as a stochastic shortest path problem, and what is then the optimal policy?

### 7.27 (Eliminating Self-Transitions)

Consider a stochastic shortest path problem (SSP) with termination state  $t$ , the nontermination states  $1, \dots, n$ , transition probabilities  $p_{ij}(u)$ , and expected costs per stage  $g(i, u)$ . Let Assumption 7.2.1 hold.

- (a) Modify the costs and transition probabilities as follows:

$$\tilde{g}(i, u) = \frac{g(i, u)}{1 - p_{ii}(u)}, \quad i = 1, \dots, n, u \in U(i),$$

$$\tilde{p}_{ij}(u) = \begin{cases} 0 & \text{if } j = i, \\ \frac{p_{ij}(u)}{1 - p_{ii}(u)} & \text{if } j \neq i, \end{cases} \quad i = 1, \dots, n, j = 1, \dots, n, t, u \in U(i),$$

to obtain another SSP without self-transitions. Show that the modified SSP is equivalent to the original in the sense that its stationary policies and optimal policies have the same cost functions. What is the interpretation of the transitions of the modified SSP in terms of transitions of the original?

- (b) Fix a policy  $\mu$ . Let  $J_k$  and  $\tilde{J}_k$  be the sequences of cost vectors generated by value iteration (for the fixed policy) in the original and the modified SSP, respectively, starting from the same initial vector  $J_0$ . Show that value iteration is faster for the modified SSP in the sense that if  $J_0 \leq J_1$ , then  $J_k \leq \tilde{J}_k \leq J^*$  for all  $k$ , and if  $J_0 \geq J_1$ , then  $J_k \geq \tilde{J}_k \geq J^*$  for all  $k$ .

### 7.28 (Total Cost Problems with Nonnegative Costs) [www](#)

This is a theoretical problem whose purpose is to provide some additional analysis for undiscounted cost problems, including an extension of the results of Section 7.2 for stochastic shortest path problems. The idea is to use the analysis of Section 7.3 for discounted problems to derive the basic results for total undiscounted cost problems under the assumption that the stage costs are nonnegative and the optimal costs are finite. These results apply, among others, to some stochastic shortest path problems where not all stationary policies are proper and Assumption 7.2.1 is violated.

Consider a controlled Markov chain with states  $i = 1, \dots, n$ , controls  $u$  chosen from a finite constraint set  $U(i)$  for each state  $i$ , and transition probabilities  $p_{ij}(u)$ . (The states may include a cost-free and absorbing termination state, but this is not relevant for the following analysis.) The cost of the  $k$ th stage at state  $i$  when control  $u$  is applied has the form

$$\alpha^k g(i, u), \quad i = 1, \dots, n, \quad u \in U(i),$$

where  $\alpha$  is a scalar from  $(0, 1]$ . Our key assumption is that

$$0 \leq g(i, u), \quad i = 1, \dots, n, \quad u \in U(i).$$

For any policy  $\pi$ , let  $J_{\pi, \alpha}$  be the cost function for the  $\alpha$ -discounted problem ( $\alpha < 1$ ), and let  $J_\pi$  be the cost function for the problem where  $\alpha = 1$ . Note that for  $\alpha = 1$ , we may have  $J_\pi(i) = \infty$  for some  $\pi$  and  $i$ . However, the limit defining  $J_\pi(i)$  exists either as a real number or  $\infty$ , thanks to the assumption  $0 \leq g(i, u)$  for all  $i$  and  $u$ . Let  $J_\alpha^*(i)$  and  $J^*(i)$  be the optimal costs starting from  $i$ , when  $\alpha < 1$  and  $\alpha = 1$ , respectively. We assume that

$$J^*(i) < \infty, \quad i = 1, \dots, n,$$

(this is true in particular for the case of a stochastic shortest path problem if there exists a proper stationary policy, i.e., a policy under which there is a positive transition probability path from every state to the termination state).

- (a) Show that for all  $\alpha < 1$ , we have

$$0 \leq J_\alpha^*(i) \leq J^*(i), \quad i = 1, \dots, n.$$

- (b) Show that for any admissible policy  $\pi$ , we have

$$\lim_{\alpha \uparrow 1} J_{\pi, \alpha}(i) = J_\pi(i), \quad i = 1, \dots, n.$$

Furthermore,

$$\lim_{\alpha \uparrow 1} J_\alpha^*(i) = J^*(i), \quad i = 1, \dots, n.$$

*Hint:* To show the first equality, note that for any  $\alpha < 1$ ,  $N$ , and  $\pi = \{\mu_0, \mu_1, \dots\}$ , we have

$$J_\pi(i) \geq J_{\pi, \alpha}(i) \geq \sum_{k=0}^{N-1} \alpha^k E\{g(i_k, \mu_k(i_k)) \mid i_0 = i, \pi\}.$$

Take the limit as  $\alpha \rightarrow 1$  and then take the limit as  $N \rightarrow \infty$ . For the second equality, consider a stationary policy  $\mu$  and a sequence  $\{a_m\} \subset (0, 1)$  with  $a_m \rightarrow 1$  such that  $J_{\mu, a_m} = J_{a_m}^*$  for all  $m$ .

- (c) Use Bellman's equation for  $\alpha < 1$  to show that  $J^*$  satisfies Bellman's equation for  $\alpha = 1$ :

$$J^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J^*(j) \right], \quad i = 1, \dots, n.$$

- (d) Let  $\bar{J}$  be such that  $0 \leq \bar{J}(i) < \infty$  for all  $i$ . Show that if

$$\bar{J}(i) \geq \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) \bar{J}(j) \right], \quad i = 1, \dots, n,$$

then  $\bar{J}(i) \geq J^*(i)$  for all  $i$ . Show also that if for some stationary policy  $\mu$ , we have

$$\bar{J}(i) \geq g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) \bar{J}(j), \quad i = 1, \dots, n,$$

then  $\bar{J}(i) \geq J_\mu(i)$  for all  $i$ . *Hint:* Argue that

$$\bar{J}(i) \geq \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) \bar{J}(j) \right], \quad i = 1, \dots, n,$$

use value iteration to show that  $\bar{J} \geq J_\alpha^*$ , and take the limit as  $\alpha \rightarrow 1$ .

- (e) For  $\alpha = 1$ , show that if

$$\mu^*(i) = \arg \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J^*(j) \right], \quad i = 1, \dots, n,$$

then  $\mu^*$  is optimal. *Hint:* Use part (d) with  $\bar{J} = J^*$ .

- (f) For  $\phi = 1$ , show that for the value iteration method, given by

$$J_{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad i = 1, \dots, n,$$

we have  $J_k(i) \rightarrow J^*(i)$ ,  $i = 1, \dots, n$ , assuming that

$$0 \leq J_0(i) \leq J^*(i), \quad i = 1, \dots, n.$$

Give examples showing what may happen when this last assumption is violated. *Hint:* Prove the result by first assuming that  $J_0$  is the zero function.

- (g) Show that the set of states  $Z = \{i \mid J^*(i) = 0\}$  is nonempty. Furthermore, under an optimal stationary policy  $\mu^*$ , the set of states  $Z$  is cost-free and absorbing, i.e.,  $g(i, \mu^*(i)) = 0$  and  $p_{ij}(\mu^*(i)) = 0$  for all  $i \in Z$  and  $j \notin Z$ . In addition,  $\mu^*$  is proper in the sense that for every state  $i \notin Z$ , under  $\mu^*$ , there is a positive probability path that starts at  $i$  and ends at a state of  $Z$ .

## APPENDIX A: Mathematical Review

The purpose of this appendix is to provide a list of mathematical definitions, notations, and results that are used frequently in the text. For detailed expositions, the reader may consult textbooks such as Hoffman and Kunze [HoK71], Royden [Roy88], Rudin [Rud76], and Strang [Str76].

### A.1 SETS

If  $x$  is a member of the set  $S$ , we write  $x \in S$ . We write  $x \notin S$  if  $x$  is not a member of  $S$ . A set  $S$  may be specified by listing its elements within braces. For example, by writing  $S = \{x_1, x_2, \dots, x_n\}$  we mean that the set  $S$  consists of the elements  $x_1, x_2, \dots, x_n$ . A set  $S$  may also be specified in the generic form

$$S = \{x \mid x \text{ satisfies } P\}$$

as the set of elements satisfying property  $P$ . For example,

$$S = \{x \mid x : \text{real}, 0 \leq x \leq 1\}$$

denotes the set of all real numbers  $x$  satisfying  $0 \leq x \leq 1$ .

The *union* of two sets  $S$  and  $T$  is denoted by  $S \cup T$  and the *intersection* of  $S$  and  $T$  is denoted by  $S \cap T$ . The union and the intersection of a sequence of sets  $S_1, S_2, \dots, S_k, \dots$  are denoted by  $\cup_{k=1}^{\infty} S_k$  and  $\cap_{k=1}^{\infty} S_k$ , respectively. If  $S$  is a subset of  $T$  (i.e., if every element of  $S$  is also an element of  $T$ ), we write  $S \subset T$  or  $T \supset S$ .

## Finite and Countable Sets

A set  $S$  is said to be *finite* if it consists of a finite number of elements. It is said to be *countable* if there exists a one-to-one function from  $S$  into the set of nonnegative integers. Thus, according to our definition, a finite set is also countable but not conversely. A countable set  $S$  that is not finite may be represented by listing its elements  $x_0, x_1, x_2, \dots$  (i.e.,  $S = \{x_0, x_1, x_2, \dots\}$ ). A countable union of countable sets is countable, that is, if  $A = \{a_0, a_1, \dots\}$  is a countable set and  $S_{a_0}, S_{a_1}, \dots$  are each countable sets, then  $\bigcup_{k=0}^{\infty} S_{a_k}$  is also a countable set.

## Sets of Real Numbers

If  $a$  and  $b$  are real numbers or  $+\infty, -\infty$ , we denote by  $[a, b]$  the set of numbers  $x$  satisfying  $a \leq x \leq b$  (including the possibility  $x = +\infty$  or  $x = -\infty$ ). A rounded, instead of square, bracket denotes strict inequality in the definition. Thus  $(a, b)$ ,  $[a, b)$ , and  $(a, b)$  denote the set of all  $x$  satisfying  $a < x \leq b$ ,  $a \leq x < b$ , and  $a < x < b$ , respectively.

If  $S$  is a set of real numbers that is bounded above, then there is a smallest real number  $y$  such that  $x \leq y$  for all  $x \in S$ . This number is called the *least upper bound* or *supremum* of  $S$  and is denoted by  $\sup\{x \mid x \in S\}$  or  $\max\{x \mid x \in S\}$ . (This is somewhat inconsistent with normal mathematical usage, where the use of max in place of sup indicates that the supremum is attained by some element of  $S$ .) Similarly, the greatest real number  $z$  such that  $z \leq x$  for all  $x \in S$  is called the *greatest lower bound* or *infimum* of  $S$  and is denoted by  $\inf\{x \mid x \in S\}$  or  $\min\{x \mid x \in S\}$ . If  $S$  is unbounded above, we write  $\sup\{x \mid x \in S\} = +\infty$ , and if it is unbounded below, we write  $\inf\{x \mid x \in S\} = -\infty$ . If  $S$  is the empty set, then by convention we write  $\inf\{x \mid x \in S\} = +\infty$  and  $\sup\{x \mid x \in S\} = -\infty$ .

## A.2 EUCLIDEAN SPACE

The set of all  $n$ -tuples  $x = (x_1, \dots, x_n)$  of real numbers constitutes the  *$n$ -dimensional Euclidean space*, denoted by  $\mathbb{R}^n$ . The elements of  $\mathbb{R}^n$  are referred to as  $n$ -dimensional vectors or simply vectors when confusion cannot arise. The one-dimensional Euclidean space  $\mathbb{R}^1$  consists of all the real numbers and is denoted by  $\mathbb{R}$ . Vectors in  $\mathbb{R}^n$  can be added by adding their corresponding components. They can be multiplied by a scalar by multiplication of each component by a scalar. The *inner product* of two vectors  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  is denoted by  $x'y$  and is equal to  $\sum_{i=1}^n x_i y_i$ . The *norm* of a vector  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$  is denoted by  $\|x\|$  and is equal to  $(x'x)^{1/2} = (\sum_{i=1}^n x_i^2)^{1/2}$ .

A set of vectors  $a_1, a_2, \dots, a_k$  is said to be *linearly dependent* if there exist scalars  $\lambda_1, \lambda_2, \dots, \lambda_k$ , not all zero, such that

$$\lambda_1 a_1 + \dots + \lambda_k a_k = 0.$$

If no such set of scalars exists, the vectors are said to be *linearly independent*.

## A.3 MATRICES

An  $m \times n$  *matrix* is a rectangular array of numbers, referred to as elements or components, which are arranged in  $m$  rows and  $n$  columns. If  $m = n$  the matrix is said to be *square*. The element in the  $i$ th row and  $j$ th column of a matrix  $A$  is denoted by a subscript  $ij$ , such as  $a_{ij}$ , in which case we write  $A = [a_{ij}]$ . The  $n \times n$  *identity matrix*, denoted by  $I$ , is the matrix with elements  $a_{ij} = 0$  for  $i \neq j$  and  $a_{ii} = 1$ , for  $i = 1, \dots, n$ . The *sum* of two  $m \times n$  matrices  $A$  and  $B$  is written as  $A + B$  and is the matrix whose elements are the sum of the corresponding elements in  $A$  and  $B$ . The *product of a matrix  $A$  and a scalar  $\lambda$* , written as  $\lambda A$  or  $A\lambda$ , is obtained by multiplying each element of  $A$  by  $\lambda$ . The *product  $AB$  of an  $m \times n$  matrix  $A$  and an  $n \times p$  matrix  $B$*  is the  $m \times p$  matrix  $C$  with elements  $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$ . If  $b$  is an  $n$ -dimensional column vector and  $A$  is an  $m \times n$  matrix, then  $Ab$  is an  $m$ -dimensional column vector.

The *transpose* of an  $m \times n$  matrix  $A$  is the  $n \times m$  matrix  $A'$  with elements  $a'_{ij} = a_{ji}$ . The elements of a given row (or column) of  $A$  constitute a vector called a row vector (or column vector, respectively) of  $A$ . A square matrix  $A$  is *symmetric* if  $A' = A$ . An  $n \times n$  matrix  $A$  is called *nonsingular* or *invertible* if there is an  $n \times n$  matrix called the *inverse* of  $A$  and denoted by  $A^{-1}$ , such that  $A^{-1}A = I = AA^{-1}$ , where  $I$  is the  $n \times n$  identity matrix. An  $n \times n$  matrix is nonsingular if and only if its  $n$  row vectors are linearly independent or, equivalently, if its  $n$  column vectors are linearly independent. Thus, an  $n \times n$  matrix  $A$  is nonsingular if and only if the relation  $Av = 0$ , where  $v \in \mathbb{R}^n$ , implies that  $v = 0$ .

## Rank of a Matrix

The *rank* of a matrix  $A$  is equal to the maximum number of linearly independent row vectors of  $A$ . It is also equal to the maximum number of linearly independent column vectors. Thus, the rank of an  $m \times n$  matrix is at most equal to the minimum of the dimensions  $m$  and  $n$ . An  $m \times n$  matrix is said to be of *full rank* if its rank is maximal, that is, if its rank is equal to the minimum of  $m$  and  $n$ . A square matrix is of full rank if and only if it is nonsingular.

## Eigenvalues

Given a square  $n \times n$  matrix  $A$ , the determinant of the matrix  $\gamma I - A$ , where  $I$  is the  $n \times n$  identity matrix and  $\gamma$  is a scalar, is an  $n$ th degree polynomial. The  $n$  roots of this polynomial are called the *eigenvalues* of  $A$ . Thus,  $\gamma$  is an eigenvalue of  $A$  if and only if the matrix  $\gamma I - A$  is singular, or equivalently, if and only if there exists a nonzero vector  $v$  such that  $Av = \gamma v$ . Such a vector  $v$  is called an *eigenvector* corresponding to  $\gamma$ . The eigenvalues and eigenvectors of  $A$  can be complex even if  $A$  is real. A matrix  $A$  is singular if and only if it has an eigenvalue that is equal to zero. If  $A$  is nonsingular, then the eigenvalues of  $A^{-1}$  are the reciprocals of the eigenvalues of  $A$ . The eigenvalues of  $A$  and  $A'$  coincide.

If  $\gamma_1, \dots, \gamma_n$  are the eigenvalues of  $A$ , then the eigenvalues of  $cI + A$ , where  $c$  is a scalar and  $I$  is the identity matrix, are  $c + \gamma_1, \dots, c + \gamma_n$ . The eigenvalues of  $A^k$ , where  $k$  is any positive integer, are equal to  $\gamma_1^k, \dots, \gamma_n^k$ . From this it follows that  $\lim_{k \rightarrow \infty} A^k = 0$  if and only if all the eigenvalues of  $A$  lie strictly within the unit circle of the complex plane. Furthermore, if the latter condition holds, the iteration

$$x_{k+1} = Ax_k + b,$$

where  $b$  is a given vector, converges to

$$\bar{x} = (I - A)^{-1}b,$$

which is the unique solution of the equation  $x = Ax + b$ .

If all the eigenvalues of  $A$  are distinct, then their number is exactly  $n$ , and there exists a set of corresponding linearly independent eigenvectors. In this case, if  $\gamma_1, \dots, \gamma_n$  are the eigenvalues and  $v_1, \dots, v_n$  are such eigenvectors, every vector  $x \in \mathbb{R}^n$  can be decomposed as

$$x = \sum_{i=1}^n \xi_i v_i,$$

where  $\xi_i$  are some unique (possibly complex) numbers. Furthermore, we have for all positive integers  $k$ ,

$$A^k x = \sum_{i=1}^n \gamma_i^k \xi_i v_i.$$

If  $A$  is a transition probability matrix, that is, all the elements of  $A$  are nonnegative and the sum of the elements of each of its rows is equal to 1, then all the eigenvalues of  $A$  lie within the unit circle of the complex plane. Furthermore, 1 is an eigenvalue of  $A$  and the unit vector  $(1, 1, \dots, 1)$  is a corresponding eigenvector.

## Positive Definite and Semidefinite Symmetric Matrices

A square symmetric  $n \times n$  matrix  $A$  is said to be *positive semidefinite* if  $x'Ax \geq 0$  for all  $x \in \mathbb{R}^n$ . It is said to be *positive definite* if  $x'Ax > 0$  for all nonzero  $x \in \mathbb{R}^n$ . The matrix  $A$  is said to be *negative semidefinite* (*definite*) if  $-A$  is *positive semidefinite* (*definite*). In this book, the notions of positive definiteness and semidefiniteness will be used only in connection with symmetric matrices.

A positive definite symmetric matrix is invertible and its inverse is also positive definite symmetric. Also, an invertible positive semidefinite symmetric matrix is positive definite. Analogous results hold for negative definite and semidefinite symmetric matrices. If  $A$  and  $B$  are  $n \times n$  positive semidefinite (definite) symmetric matrices, then the matrix  $\lambda A + \mu B$  is also positive semidefinite (definite) symmetric for all  $\lambda \geq 0$  and  $\mu \geq 0$ . If  $A$  is an  $n \times n$  positive semidefinite symmetric matrix and  $C$  is an  $m \times n$  matrix, then the matrix  $CAC'$  is positive semidefinite symmetric. If  $A$  is positive definite symmetric, and  $C$  has rank  $m$  (equivalently,  $m \leq n$  and  $C$  has full rank), then  $CAC'$  is positive definite symmetric.

An  $n \times n$  positive definite symmetric matrix  $A$  can be written as  $CC'$  where  $C$  is a square invertible matrix. If  $A$  is positive semidefinite symmetric and its rank is  $m$ , then it can be written as  $CC'$ , where  $C$  is an  $n \times m$  matrix of full rank.

A symmetric  $n \times n$  matrix  $A$  has real eigenvalues and a set of  $n$  real linearly independent eigenvectors, which are orthogonal (the inner product of any pair is 0). If  $A$  is positive semidefinite (definite) symmetric, its eigenvalues are nonnegative (respectively, positive).

## Partitioned Matrices

It is often convenient to partition a matrix into submatrices. For example, the matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

may be partitioned into

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where

$$A_{11} = (a_{11} \ a_{12}), \quad A_{12} = (a_{13} \ a_{14}),$$

$$A_{21} = (a_{21} \ a_{22}), \quad A_{22} = (a_{23} \ a_{24}).$$

We separate the components of a partitioned matrix by a space, as in  $(B \ C)$ , or by a comma, as in  $(B, C)$ . The transpose of the partitioned matrix  $A$  is

$$A' = \begin{pmatrix} A'_{11} & A'_{21} \\ A'_{12} & A'_{22} \end{pmatrix}.$$

Partitioned matrices may be multiplied just as nonpartitioned matrices, provided the dimensions involved in the partitions are compatible. Thus if

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

then

$$AB = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix},$$

provided the dimensions of the submatrices are such that the preceding products  $A_{ij}B_{jk}$ ,  $i, j, k = 1, 2$  can be formed.

### Matrix Inversion Formulas

Let  $A$  and  $B$  be square invertible matrices, and let  $C$  be a matrix of appropriate dimension. Then, if all the following inverses exist, we have

$$(A + CBC')^{-1} = A^{-1} - A^{-1}C(B^{-1} + C'A^{-1}C)^{-1}C'A^{-1}.$$

The equation can be verified by multiplying the right-hand side by

$$A + CBC'$$

and showing that the product is the identity matrix.

Consider a partitioned matrix  $M$  of the form

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

Then we have

$$M^{-1} = \begin{pmatrix} Q & -QBD^{-1} \\ -D^{-1}CQ & D^{-1} + D^{-1}CQBD^{-1} \end{pmatrix},$$

where

$$Q = (A - BD^{-1}C)^{-1},$$

provided all the inverses exist. The proof is obtained by multiplying  $M$  with the expression given for  $M^{-1}$  and verifying that the product yields the identity matrix.

## A.4 ANALYSIS

### Convergence of Sequences

A sequence of vectors  $x_0, x_1, \dots, x_k, \dots$  in  $\mathbb{R}^n$ , denoted by  $\{x_k\}$ , is said to converge to a *limit*  $x$  if  $\|x_k - x\| \rightarrow 0$  as  $k \rightarrow \infty$  (i.e., if, given any  $\epsilon > 0$ , there is an integer  $N$  such that for all  $k \geq N$  we have  $\|x_k - x\| < \epsilon$ ). If  $\{x_k\}$  converges to  $x$ , we write  $x_k \rightarrow x$  or  $\lim_{k \rightarrow \infty} x_k = x$ . We have  $Ax_k + By_k \rightarrow Ax + By$  if  $x_k \rightarrow x$ ,  $y_k \rightarrow y$ , and  $A$ ,  $B$  are matrices of appropriate dimension.

A vector  $x$  is said to be a *limit point* of a sequence  $\{x_k\}$  if there is a subsequence of  $\{x_k\}$  that converges to  $x$ , that is, if there is an infinite subset  $K$  of the nonnegative integers such that for any  $\epsilon > 0$ , there is an integer  $N$  such that for all  $k \in K$  with  $k \geq N$  we have  $\|x_k - x\| < \epsilon$ .

A sequence of real numbers  $\{r_k\}$ , which is monotonically nondecreasing (nonincreasing), that is, satisfies  $r_k \leq r_{k+1}$  for all  $k$ , must either converge to a real number or be unbounded above (below). In the latter case we write  $\lim_{k \rightarrow \infty} r_k = \infty$  ( $-\infty$ ). Given any bounded sequence of real numbers  $\{r_k\}$ , we may consider the sequence  $\{s_k\}$ , where  $s_k = \sup\{r_i \mid i \geq k\}$ . Since this sequence is monotonically nonincreasing and bounded, it must have a limit. This limit is called the *limit superior* of  $\{r_k\}$  and is denoted by  $\limsup_{k \rightarrow \infty} r_k$ . The *limit inferior* of  $\{r_k\}$  is similarly defined and is denoted by  $\liminf_{k \rightarrow \infty} r_k$ . If  $\{r_k\}$  is unbounded above, we write  $\limsup_{k \rightarrow \infty} r_k = \infty$ , and if it is unbounded below, we write  $\liminf_{k \rightarrow \infty} r_k = -\infty$ . We also use this notation if  $r_k \in [-\infty, \infty]$  for all  $k$ .

### Open, Closed, and Compact Sets

A subset  $S$  of  $\mathbb{R}^n$  is said to be *open* if for every vector  $x \in S$  one can find an  $\epsilon > 0$  such that  $\{z \mid \|z - x\| < \epsilon\} \subset S$ . A set  $S$  is *closed* if and only if every convergent sequence  $\{x_k\}$  with elements in  $S$  converges to a point that also belongs to  $S$ . A set  $S$  is said to be *compact* if and only if it is both closed and bounded (i.e., it is closed and for some  $M > 0$  we have  $\|x\| \leq M$  for all  $x \in S$ ). A set  $S$  is compact if and only if every sequence  $\{x_k\}$  with elements in  $S$  has at least one limit point that belongs to  $S$ . Another important fact is that if  $S_0, S_1, \dots, S_k, \dots$  is a sequence of nonempty compact sets in  $\mathbb{R}^n$  such that  $S_k \supset S_{k+1}$  for all  $k$ , then the intersection  $\bigcap_{k=0}^{\infty} S_k$  is a nonempty and compact set.

### Continuous Functions

A function  $f$  mapping a set  $S_1$  into a set  $S_2$  is denoted by  $f : S_1 \rightarrow S_2$ . A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is said to be *continuous* if for all  $x$ ,  $f(x_k) \rightarrow f(x)$  whenever  $x_k \rightarrow x$ . Equivalently,  $f$  is continuous if, given  $x \in \mathbb{R}^n$  and  $\epsilon > 0$ ,

there is a  $\delta > 0$  such that whenever  $\|y - x\| < \delta$ , we have  $\|f(y) - f(x)\| < \epsilon$ . The function

$$(a_1 f_1 + a_2 f_2)(\cdot) = a_1 f_1(\cdot) + a_2 f_2(\cdot)$$

is continuous for any two scalars  $a_1, a_2$  and any two continuous functions  $f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . If  $S_1, S_2, S_3$  are any sets and  $f_1 : S_1 \rightarrow S_2, f_2 : S_2 \rightarrow S_3$  are functions, the function  $f_2 \circ f_1 : S_1 \rightarrow S_3$  defined by  $(f_2 \circ f_1)(x) = f_2(f_1(x))$  is called the *composition* of  $f_1$  and  $f_2$ . If  $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $f_2 : \mathbb{R}^m \rightarrow \mathbb{R}^p$  are continuous, then  $f_2 \circ f_1$  is also continuous.

### Derivatives

Let  $f : \mathbb{R}^n \mapsto \mathbb{R}$  be some function. For a fixed  $x \in \mathbb{R}^n$ , the first partial derivative of  $f$  at the point  $x$  with respect to the  $i$ th coordinate is defined by

$$\frac{\partial f(x)}{\partial x_i} = \lim_{\alpha \rightarrow 0} \frac{f(x + \alpha e_i) - f(x)}{\alpha},$$

where  $e_i$  is the  $i$ th unit vector, and we assume that the above limit exists. If the partial derivatives with respect to all coordinates exist,  $f$  is called differentiable at  $x$  and its *gradient* at  $x$  is defined to be the column vector

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

The function  $f$  is called differentiable if it is differentiable at every  $x \in \mathbb{R}^n$ . If  $\nabla f(x)$  exists for every  $x$  and is a continuous function of  $x$ ,  $f$  is said to be *continuously differentiable*. Such a function admits, for every fixed  $x$ , the first order expansion

$$f(x + y) = f(x) + y' \nabla f(x) + o(\|y\|),$$

where  $o(\|y\|)$  is a function of  $y$  with the property  $\lim_{\|y\| \rightarrow 0} o(\|y\|)/\|y\| = 0$ .

A vector-valued function  $f : \mathbb{R}^n \mapsto \mathbb{R}^m$  is called differentiable (respectively, continuously differentiable) if each component  $f_i$  of  $f$  is differentiable (respectively, continuously differentiable). The *gradient matrix* of  $f$ , denoted by  $\nabla f(x)$ , is the  $n \times m$  matrix whose  $i$ th column is the gradient  $\nabla f_i(x)$  of  $f_i$ . Thus,

$$\nabla f(x) = [\nabla f_1(x) \cdots \nabla f_m(x)].$$

The transpose of  $\nabla f$  is the *Jacobian* of  $f$ ; it is the matrix whose  $ij$ th entry is equal to the partial derivative  $\partial f_i / \partial x_j$ .

If the gradient  $\nabla f(x)$  is itself a differentiable function, then  $f$  is said to be twice differentiable. We denote by  $\nabla^2 f(x)$  the Hessian matrix of  $f$  at  $x$ , that is, the matrix

$$\nabla^2 f(x) = \left[ \frac{\partial^2 f(x)}{\partial x^i \partial x^j} \right]$$

the elements of which are the second partial derivatives of  $f$  at  $x$ .

Let  $f : \mathbb{R}^k \mapsto \mathbb{R}^m$  and  $g : \mathbb{R}^m \mapsto \mathbb{R}^n$  be continuously differentiable functions, and let  $h(x) = g(f(x))$ . The *chain rule* for differentiation states that

$$\nabla h(x) = \nabla f(x) \nabla g(f(x)), \quad \text{for all } x \in \mathbb{R}^k.$$

For example, if  $A$  and  $B$  are given matrices, then if  $h(x) = Ax$ , we have  $\nabla h(x) = A'$  and if  $h(x) = ABx$ , we have  $\nabla h(x) = B'A'$ .

### A.5 CONVEX SETS AND FUNCTIONS

A subset  $C$  of  $\mathbb{R}^n$  is said to be *convex* if for every  $x, y \in C$  and every scalar  $\alpha$  with  $0 \leq \alpha \leq 1$ , we have  $\alpha x + (1 - \alpha)y \in C$ . In words,  $C$  is convex if the line segment connecting any two points in  $C$  belongs to  $C$ . A function  $f : C \rightarrow \mathbb{R}$ , defined over a convex subset  $C$  of  $\mathbb{R}^n$ , is said to be *convex* if for every  $x, y \in C$  and every scalar  $\alpha$  with  $0 \leq \alpha \leq 1$  we have

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

The function  $f$  is said to be *concave* if  $(-f)$  is convex, or equivalently if for every  $x, y \in C$  and every scalar  $\alpha$  with  $0 \leq \alpha \leq 1$  we have

$$f(\alpha x + (1 - \alpha)y) \geq \alpha f(x) + (1 - \alpha)f(y).$$

If  $f : C \rightarrow \mathbb{R}$  is convex, then the sets  $\Gamma_\lambda = \{x \mid x \in C, f(x) \leq \lambda\}$  are convex for every scalar  $\lambda$ . An important property is that a real-valued convex function defined over  $\mathbb{R}^n$  is continuous.

If  $f_1, f_2, \dots, f_m$  are convex functions defined over a convex subset  $C$  of  $\mathbb{R}^n$  and  $\alpha_1, \alpha_2, \dots, \alpha_m$  are nonnegative scalars, then the function  $\alpha_1 f_1 + \dots + \alpha_m f_m$  is also convex over  $C$ . If  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex,  $A$  is an  $m \times n$  matrix, and  $b$  is a vector in  $\mathbb{R}^m$ , the function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  defined by  $g(x) = f(Ax + b)$  is also convex. If  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex, then the function  $g(x) = E_w\{f(x + w)\}$ , where  $w$  is a random vector in  $\mathbb{R}^n$ , is a convex function provided the expected value is finite for every  $x \in \mathbb{R}^n$ .

For functions  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  that are differentiable, there are alternative characterizations of convexity. Thus,  $f$  is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)'(y - x), \quad \text{for all } x, y \in \mathbb{R}^n.$$

If  $f$  is twice continuously differentiable, then  $f$  is convex if and only if  $\nabla^2 f(x)$  is a positive semidefinite symmetric matrix for every  $x \in \mathbb{R}^n$ .

For accounts of convexity and its applications in optimization, see Bertsekas [BNO03] and Rockafellar [Roc70].

# APPENDIX B:

## On Optimization Theory

The purpose of this appendix is to provide a few definitions and results of deterministic optimization. For detailed expositions, which include both convex and nonconvex problems, see textbooks such as Bertsekas [Ber99], [BNO03], Luenberger [Lue84], and Rockafellar [Roc70].

### B.1 OPTIMAL SOLUTIONS

Given a set  $S$ , a real-valued function  $f : S \rightarrow \mathbb{R}$ , and a subset  $X \subseteq S$ , the optimization problem

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } x \in X, \end{aligned} \tag{B.1}$$

is to find an element  $x^* \in X$  (called a *minimizing element* or an *optimal solution*) such that

$$f(x^*) \leq f(x), \quad \text{for all } x \in X.$$

For any minimizing element  $x^*$ , we write

$$x^* = \arg \min_{x \in X} f(x).$$

Note that a minimizing element need not exist. For example, the scalar functions  $f(x) = x$  and  $f(x) = e^x$  have no minimizing elements over the set of real numbers. The first function decreases without bound to  $-\infty$  as  $x$  tends toward  $-\infty$ , while the second decreases toward 0 as  $x$  tends toward  $-\infty$  but always takes positive values. Given the range of values that  $f(x)$  takes as  $x$  ranges over  $X$ , that is, the set of real numbers

$$\{f(x) \mid x \in X\},$$

there are two possibilities:

1. The set  $\{f(x) \mid x \in X\}$  is unbounded below (i.e., contains arbitrarily small real numbers) in which case we write

$$\min\{f(x) \mid x \in X\} = -\infty \quad \text{or} \quad \min_{x \in X} f(x) = -\infty.$$

2. The set  $\{f(x) \mid x \in X\}$  is bounded below; that is, there exists a scalar  $M$  such that  $M \leq f(x)$  for all  $x \in X$ . The greatest lower bound of  $\{f(x) \mid x \in X\}$  is also denoted by

$$\min\{f(x) \mid x \in X\} \quad \text{or} \quad \min_{x \in X} f(x).$$

In either case we call  $\min_{x \in X} f(x)$  the *optimal value* of problem (B.1).

A maximization problem of the form

$$\begin{aligned} & \text{maximize } f(x) \\ & \text{subject to } x \in X \end{aligned}$$

may be converted to the minimization problem

$$\begin{aligned} & \text{minimize } -f(x) \\ & \text{subject to } x \in X, \end{aligned}$$

in the sense that both problems have the same optimal solutions, and the optimal value of one is equal to minus the optimal value of the other. The optimal value for the maximization problem is denoted by  $\max_{x \in X} f(x)$ .

Existence of at least one optimal solution in problem (B.1) is guaranteed if  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuous function and  $X$  is a compact subset of  $\mathbb{R}^n$ . This is the *Weierstrass theorem*. By a related result, existence of an optimal solution is guaranteed if  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuous function,  $X$  is closed, and  $f(x) \rightarrow \infty$  if  $\|x\| \rightarrow \infty$ .

## B.2 OPTIMALITY CONDITIONS

Optimality conditions are available when  $f$  is a differentiable function on  $\mathbb{R}^n$  and  $X$  is a convex subset of  $\mathbb{R}^n$  (possibly  $X = \mathbb{R}^n$ ). In particular, if  $x^*$  is an optimal solution of problem (B.1),  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function on  $\mathbb{R}^n$ , and  $X$  is convex, we have

$$\nabla f(x^*)'(x - x^*) \geq 0, \quad \text{for all } x \in X, \quad (\text{B.2})$$

where  $\nabla f(x^*)$  denotes the gradient of  $f$  at  $x^*$ . When  $X = \mathbb{R}^n$  (i.e., the minimization is unconstrained), the necessary condition (B.2) is equivalent to

$$\nabla f(x^*) = 0. \quad (\text{B.3})$$

When  $f$  is twice continuously differentiable and  $X = \mathbb{R}^n$ , an additional necessary condition is that the *Hessian matrix*  $\nabla^2 f(x^*)$  be positive semidefinite at  $x^*$ . An important fact is that if  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a convex function and  $X$  is convex, then Eq. (B.2) is both a necessary and a sufficient condition for optimality of  $x^*$ .

Other types of optimality conditions deal with the case where the constraint set  $X$  consists of equality and inequality constraints, i.e., problems of the form

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } h_1(x) = 0, \dots, h_m(x) = 0, g_1(x) \leq 0, \dots, g_r(x) \leq 0, \end{aligned}$$

where  $f, h_i, g_j$  are continuously differentiable functions from  $\mathbb{R}^n$  to  $\mathbb{R}$ .

We say that the vectors  $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*)$  and  $\mu^* = (\mu_1^*, \dots, \mu_r^*)$  are *Lagrange multiplier vectors* corresponding to a local minimum  $x^*$  if they satisfy the following conditions:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) + \sum_{j=1}^r \mu_j^* \nabla g_j(x^*) &= 0, \\ \mu_j^* &\geq 0, \quad \text{for all } j = 1, \dots, r, \\ \mu_j^* &= 0, \quad \text{for all } j \notin A(x^*), \end{aligned}$$

where  $A(x^*)$  is the index set of inequality constraints that are active at  $x^*$ :

$$A(x^*) = \{j \mid g_j(x^*) = 0\}.$$

Lagrange multiplier theory revolves around conditions under which Lagrange multiplier vectors are guaranteed to exist for a given local minimum  $x^*$ . Such conditions are known as *constraint qualifications*. Some of the most useful ones are the following:

CQ1: The equality constraint gradients  $\nabla h_i(x^*)$ ,  $i = 1, \dots, m$ , and the active inequality constraint gradients  $\nabla g_j(x^*)$ ,  $j \in A(x^*)$ , are linearly independent.

CQ2: The equality constraint gradients  $\nabla h_i(x^*)$ ,  $i = 1, \dots, m$ , are linearly independent, and there exists a  $y \in \mathbb{R}^n$  such that

$$\nabla h_i(x^*)'y = 0 \text{ for all } i = 1, \dots, m, \quad \nabla g_j(x^*)'y < 0 \text{ for all } j \in A(x^*).$$

CQ3: The functions  $h_i$  are linear and the functions  $g_j$  are concave.

CQ4: The functions  $h_i$  are linear, the functions  $g_j$  are convex, and there exists a  $y \in \mathbb{R}^n$  such that

$$g_j(y) < 0, \quad \text{for all } j = 1, \dots, r.$$

Each of the above constraint qualifications implies the existence of at least one Lagrange multiplier vector associated with  $x^*$  (unique in the case of CQ1); see e.g., [Ber99] for a detailed account.

## B.3 MINIMIZATION OF QUADRATIC FORMS

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a quadratic form

$$f(x) = \frac{1}{2}x'Qx + b'x,$$

where  $Q$  is a symmetric  $n \times n$  matrix and  $b \in \mathbb{R}^n$ . Its gradient is given by

$$\nabla f(x) = Qx + b.$$

The function  $f$  is convex if and only if  $Q$  is positive semidefinite. If  $Q$  is positive definite, then  $f$  is convex and  $Q$  is invertible, so by Eq. (B.3), a vector  $x^*$  minimizes  $f$  if and only if

$$\nabla f(x^*) = Qx^* + b = 0,$$

or equivalently

$$x^* = -Q^{-1}b.$$

# APPENDIX C:

## On Probability Theory

This appendix lists selectively some of the basic probabilistic notions that we will be using. Its main purpose is to familiarize the reader with some of our terminology. It is not meant to be exhaustive, and the reader should consult textbooks such as Ash [Ash70], Feller [Fel68], Papoulis [Pap65], Ross [Ros85], Stirzaker [Sti94], and Bertsekas and Tsitsiklis [BeT02] for detailed accounts. For fairly accessible treatments of measure theoretic probability, see Adams and Guillemin [AdG86], and Ash [Ash72].

### C.1 PROBABILITY SPACES

A *probability space* consists of

- (a) A set  $\Omega$ .
- (b) A collection  $\mathcal{F}$  of subsets of  $\Omega$ , called *events*, which includes  $\Omega$  and has the following properties:
  - (1) If  $A$  is an event, then the complement  $\bar{A} = \{\omega \in \Omega \mid \omega \notin A\}$  is also an event. (The complement of  $\Omega$  is the empty set and is considered to be an event.)
  - (2) If  $A_1, A_2, \dots, A_k, \dots$  are events, then  $\cup_{k=1}^{\infty} A_k$  is also an event.
  - (3) If  $A_1, A_2, \dots, A_k, \dots$  are events, then  $\cap_{k=1}^{\infty} A_k$  is also an event.

(c) A function  $P(\cdot)$  assigning to each event  $A$  a real number  $P(A)$ , called the *probability of the event*  $A$ , and satisfying:

- (1)  $P(A) \geq 0$  for every event  $A$ .
- (2)  $P(\Omega) = 1$ .
- (3)  $P(A_1 \cup A_2) = P(A_1) + P(A_2)$  for every pair of disjoint events  $A_1, A_2$ .
- (4)  $P(\cup_{k=1}^{\infty} A_k) = \sum_{k=1}^{\infty} P(A_k)$  for every sequence of mutually disjoint events  $A_1, A_2, \dots, A_k, \dots$

The function  $P$  is referred to as a *probability measure*.

### Convention for Finite and Countable Probability Spaces

The case of a probability space where the set  $\Omega$  is a countable (possibly finite) set is encountered frequently in this book. When we specify that  $\Omega$  is finite or countable, we implicitly assume that the associated collection of events is the collection of *all* subsets of  $\Omega$  (including  $\Omega$  and the empty set). Then, if  $\Omega$  is a finite set,  $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ , the probability space is specified by the probabilities  $p_1, p_2, \dots, p_n$ , where  $p_i$  denotes the probability of the event consisting of just  $\omega_i$ . Similarly, if  $\Omega = \{\omega_1, \omega_2, \dots, \omega_k, \dots\}$ , the probability space is specified by the corresponding probabilities  $p_1, p_2, \dots, p_k, \dots$ . In either case we refer to  $(p_1, p_2, \dots, p_n)$  or  $(p_1, p_2, \dots, p_k, \dots)$  as a *probability distribution over*  $\Omega$ .

### C.2 RANDOM VARIABLES

A *random variable* on a probability space  $(\Omega, \mathcal{F}, P)$  is a function  $x : \Omega \rightarrow \mathbb{R}$  such that for every scalar  $\lambda$  the set

$$\{\omega \in \Omega \mid x(\omega) \leq \lambda\}$$

is an event (i.e., belongs to the collection  $\mathcal{F}$ ). An  $n$ -dimensional *random vector*  $x = (x_1, x_2, \dots, x_n)$  is an  $n$ -tuple of random variables  $x_1, x_2, \dots, x_n$ , each defined on the same probability space.

We define the *distribution function*  $F : \mathbb{R} \rightarrow \mathbb{R}$  [or *cumulative distribution function* (CDF for short)] of a random variable  $x$  by

$$F(z) = P\left(\{\omega \in \Omega \mid x(\omega) \leq z\}\right);$$

that is,  $F(z)$  is the probability that the random variable takes a value less than or equal to  $z$ . We define the distribution function  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  of a random vector  $x = (x_1, x_2, \dots, x_n)$  by

$$F(z_1, z_2, \dots, z_n) = P\left(\{\omega \in \Omega \mid x_1(\omega) \leq z_1, x_2(\omega) \leq z_2, \dots, x_n(\omega) \leq z_n\}\right).$$

Given the distribution function of a random vector  $x = (x_1, \dots, x_n)$ , the (marginal) distribution function of each random variable  $x_i$  is obtained from

$$F_i(z_i) = \lim_{z_j \rightarrow -\infty, j \neq i} F(z_1, z_2, \dots, z_n).$$

The random variables  $x_1, \dots, x_n$  are said to be *independent* if

$$F(z_1, z_2, \dots, z_n) = F_1(z_1)F_2(z_2) \cdots F_n(z_n),$$

for all scalars  $z_1, \dots, z_n$ .

The *expected value* of a random variable  $x$  with distribution function  $F$  is defined by

$$E\{x\} = \int_{-\infty}^{\infty} zdF(z)$$

provided the integral is well-defined. The *expected value* of a random vector  $x = (x_1, \dots, x_n)$  is the vector

$$E\{x\} = (E\{x_1\}, E\{x_2\}, \dots, E\{x_n\}).$$

The *covariance matrix* of a random vector  $x = (x_1, \dots, x_n)$  with expected value  $E\{x\} = (\bar{x}_1, \dots, \bar{x}_n)$  is defined to be the  $n \times n$  positive semidefinite symmetric matrix

$$\begin{pmatrix} E\{(x_1 - \bar{x}_1)^2\} & \cdots & E\{(x_1 - \bar{x}_1)(x_n - \bar{x}_n)\} \\ \vdots & \vdots & \vdots \\ E\{(x_n - \bar{x}_n)(x_1 - \bar{x}_1)\} & \cdots & E\{(x_n - \bar{x}_n)^2\} \end{pmatrix},$$

provided the expected values are well-defined.

Two random vectors  $x$  and  $y$  are said to be *uncorrelated* if

$$E\{(x - E\{x\})(y - E\{y\})'\} = 0,$$

where  $(x - E\{x\})$  is viewed as a column vector and  $(y - E\{y\})'$  is viewed as a row vector.

The random vector  $x = (x_1, \dots, x_n)$  is said to be characterized by a *probability density function*  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  if

$$F(z_1, z_2, \dots, z_n) = \int_{-\infty}^{z_1} \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_n} f(y_1, \dots, y_n) dy_1 \cdots dy_n,$$

for every  $z_1, \dots, z_n$ .

### C.3 CONDITIONAL PROBABILITY

We restrict ourselves to the case where the underlying probability space  $\Omega$  is a countable (possibly finite) set and the set of events is the set of all subsets of  $\Omega$ .

Given two events  $A$  and  $B$ , we define the *conditional probability* of  $B$  given  $A$  by

$$P(B | A) = \begin{cases} \frac{P(A \cap B)}{P(A)} & \text{if } P(A) > 0, \\ 0 & \text{if } P(A) = 0. \end{cases}$$

We also use the notation  $P\{B | A\}$  in place of  $P(B | A)$ . If  $B_1, B_2, \dots$  are a countable (possibly finite) collection of mutually exclusive and exhaustive events (i.e., the sets  $B_i$  are disjoint and their union is  $\Omega$ ) and  $A$  is an event, then we have

$$P(A) = \sum_i P(A \cap B_i).$$

From the two preceding relations, we obtain the *total probability theorem*:

$$P(A) = \sum_i P(B_i)P(A | B_i).$$

We thus obtain for every  $k$ ,

$$P(B_k | A) = \frac{P(A \cap B_k)}{P(A)} = \frac{P(B_k)P(A | B_k)}{\sum_i P(B_i)P(A | B_i)},$$

assuming that  $P(A) > 0$ . This relation is referred to as *Bayes' rule*.

Consider now two random vectors  $x$  and  $y$  taking values in  $\mathbb{R}^n$  and  $\mathbb{R}^m$ , respectively [i.e.,  $x(\omega) \in \mathbb{R}^n$ ,  $y(\omega) \in \mathbb{R}^m$  for all  $\omega \in \Omega$ ]. Given two subsets  $X$  and  $Y$  of  $\mathbb{R}^n$  and  $\mathbb{R}^m$ , respectively, we denote

$$P(X | Y) = P\left(\{\omega | x(\omega) \in X\} \mid \{\omega | y(\omega) \in Y\}\right).$$

For a fixed vector  $v \in \mathbb{R}^n$ , we define the *conditional distribution function* of  $x$  given  $v$  by

$$F(z | v) = P\left(\{\omega | x(\omega) \leq z\} \mid \{\omega | y(\omega) = v\}\right),$$

and the *conditional expectation* of  $x$  given  $v$  by

$$E\{x | v\} = \int_{\mathbb{R}^n} zdF(z | v),$$

assuming that the integral is well-defined. Note that  $E\{x | v\}$  is a function mapping  $v$  into  $\mathbb{R}^n$ .

Finally, let us provide Bayes' rule for random vectors. If  $\omega_1, \omega_2, \dots$  are the elements of  $\Omega$ , denote

$$z_i = x(\omega_i), \quad v_i = y(\omega_i), \quad i = 1, 2, \dots$$

Also, for any vectors  $z \in \mathbb{R}^n$ ,  $v \in \mathbb{R}^m$ , let us denote

$$P(z) = P\left(\{\omega \mid x(\omega) = z\}\right), \quad P(v) = P\left(\{\omega \mid y(\omega) = v\}\right).$$

We have  $P(z) = 0$  if  $z \neq z_i$ ,  $i = 1, 2, \dots$ , and  $P(v) = 0$  if  $v \neq v_i$ ,  $i = 1, 2, \dots$ . Denote also

$$P(z \mid v) = P\left(\{\omega \mid x(\omega) = z\} \mid \{\omega \mid y(\omega) = v\}\right),$$

$$P(v \mid z) = P\left(\{\omega \mid y(\omega) = v\} \mid \{\omega \mid x(\omega) = z\}\right).$$

Then, for all  $k = 1, 2, \dots$ , Bayes' rule yields

$$P(z_k \mid v) = \begin{cases} \frac{P(z_k)P(v \mid z_k)}{\sum_i P(z_i)P(v \mid z_i)} & \text{if } P(v) > 0, \\ 0 & \text{if } P(v) = 0. \end{cases}$$

## APPENDIX D: On Finite-State Markov Chains

This appendix provides some of the basic probabilistic notions related to stationary Markov chains with a finite number of states. For detailed presentations, see Ash [Ash70], Bertsekas and Tsitsiklis [BeT02], Chung [Chu60], Gallager [Gal69], Kemeny and Snell [KeS60], and Ross [Ros85].

### D.1 STATIONARY MARKOV CHAINS

A square  $n \times n$  matrix  $[p_{ij}]$  is said to be a *stochastic* matrix if all its elements are nonnegative, that is,  $p_{ij} \geq 0$ ,  $i, j = 1, \dots, n$ , and the sum of the elements of each of its rows is equal to 1, that is,  $\sum_{j=1}^n p_{ij} = 1$  for all  $i = 1, \dots, n$ .

Suppose we are given a stochastic  $n \times n$  matrix  $P$  together with a finite set of states  $S = \{1, \dots, n\}$ . The pair  $(S, P)$  will be referred to as a *stationary finite-state Markov chain*. We associate with  $(S, P)$  a process whereby an initial state  $x_0 \in S$  is chosen in accordance with some initial probability distribution

$$r_0 = (r_0^1, r_0^2, \dots, r_0^n).$$

Subsequently, a transition is made from state  $x_0$  to a new state  $x_1 \in S$  in accordance with a probability distribution specified by  $P$  as follows. The

probability that the new state will be  $j$  is equal to  $p_{ij}$  whenever the initial state is  $i$ , i.e.,

$$P(x_1 = j \mid x_0 = i) = p_{ij}, \quad i, j = 1, \dots, n.$$

Similarly, subsequent transitions produce states  $x_2, x_3, \dots$  in accordance with

$$P(x_{k+1} = j \mid x_k = i) = p_{ij}, \quad i, j = 1, \dots, n. \quad (\text{D.1})$$

The probability that after the  $k$ th transition the state  $x_k$  will be  $j$ , given that the initial state  $x_0$  is  $i$ , is denoted by

$$p_{ij}^k = P(x_k = j \mid x_0 = i), \quad i, j = 1, \dots, n. \quad (\text{D.2})$$

A straightforward calculation shows that these probabilities are equal to the elements of the matrix  $P^k$  ( $P$  raised to the  $k$ th power), in the sense that  $p_{ij}^k$  is the element in the  $i$ th row and  $j$ th column of  $P^k$ :

$$P^k = [p_{ij}^k]. \quad (\text{D.3})$$

Given the initial probability distribution  $p_0$  of the state  $x_0$  (viewed as a row vector in  $\mathbb{R}^n$ ), the probability distribution of the state  $x_k$  after  $k$  transitions

$$r_k = (r_k^1, r_k^2, \dots, r_k^n)$$

(viewed again as a row vector) is given by

$$r_k = r_0 P^k, \quad k = 1, 2, \dots \quad (\text{D.4})$$

This relation follows from Eqs. (D.2) and (D.3) once we write

$$r_k^j = \sum_{i=1}^n P(x_k = j \mid x_0 = i) r_0^i = \sum_{i=1}^n p_{ij}^k r_0^i.$$

## D.2 CLASSIFICATION OF STATES

Given a stationary finite-state Markov chain  $(S, P)$ , we say that two states  $i$  and  $j$  communicate if there exist two positive integers  $k_1$  and  $k_2$  such that  $p_{ij}^{k_1} > 0$  and  $p_{ji}^{k_2} > 0$ . In words, states  $i$  and  $j$  communicate if one can be reached from the other with positive probability.

Let  $\tilde{S} \subset S$  be a subset of states such that:

1. All states in  $\tilde{S}$  communicate.

2. If  $i \in \tilde{S}$  and  $j \notin \tilde{S}$ , then  $p_{ij}^k = 0$  for all  $k$ .

Then we say that  $\tilde{S}$  forms a recurrent class of states.

If  $S$  forms by itself a recurrent class (i.e., all states communicate with each other), then we say that the Markov chain is irreducible. It is possible that there exist several recurrent classes. It can also be proved that at least one recurrent class must exist. A state that belongs to some recurrent class is called recurrent; otherwise it is called transient. We have

$$\lim_{k \rightarrow \infty} p_{ii}^k = 0 \quad \text{if and only if } i \text{ is transient.}$$

In other words, if the process starts at a transient state, the probability of returning to the same state after  $k$  transitions diminishes to zero as  $k$  tends to infinity.

The definitions imply that if the process starts within a recurrent class, it stays within that class. If it starts at a transient state, it eventually (with probability one) enters a recurrent class after a number of transitions, and subsequently remains there.

## D.3 LIMITING PROBABILITIES

An important property of any stochastic matrix  $P$  is that the matrix  $P^*$  defined by

$$P^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k \quad (\text{D.5})$$

exists [in the sense that the sequences of the elements of  $(1/N) \sum_{k=0}^{N-1} P^k$  converge to the corresponding elements of  $P^*$ ]. A proof of this is given in Prop. A.1 of Appendix A in Vol. II. The elements  $p_{ij}^*$  of  $P^*$  satisfy

$$p_{ij}^* \geq 0, \quad \sum_{j=1}^n p_{ij}^* = 1, \quad i, j = 1, \dots, n.$$

Thus,  $P^*$  is a stochastic matrix.

Note that the  $(i, j)$ th element of the matrix  $P^k$  is the probability that the state will be  $j$  after  $k$  transitions starting from state  $i$ . With this in mind, it can be seen from the definition (D.5) that  $p_{ij}^*$  can be interpreted as the long term fraction of time that the state is  $j$  given that the initial state is  $i$ . This suggests that for any two states  $i$  and  $i'$  in the same recurrent class we have  $p_{ij}^* = p_{i'j}^*$ , and this can indeed be proved. In particular, if a Markov chain is irreducible, the matrix  $P^*$  has identical rows. Also, if  $j$  is a transient state, we have

$$p_{ij}^* = 0, \quad \text{for all } i = 1, \dots, n,$$

so the columns of the matrix  $P^*$  corresponding to transient states consist of zeroes.

## D.4 FIRST PASSAGE TIMES

Let us denote by  $q_{ij}^k$  the probability that the state will be  $j$  for the first time after exactly  $k \geq 1$  transitions given that the initial state is  $i$ , that is,

$$q_{ij}^k = P(x_k = j, x_m \neq j, 1 \leq m < k \mid x_0 = i).$$

Denote also, for fixed  $i$  and  $j$ ,

$$K_{ij} = \min\{k \geq 1 \mid x_k = j, x_0 = i\}.$$

Then  $K_{ij}$ , called the *first passage time from  $i$  to  $j$* , may be viewed as a random variable. We have, for every  $k = 1, 2, \dots$ ,

$$P(K_{ij} = k) = q_{ij}^k,$$

and we write

$$P(K_{ij} = \infty) = P(x_k \neq j, k = 1, 2, \dots \mid x_0 = i) = 1 - \sum_{k=1}^{\infty} q_{ij}^k.$$

Note that it is possible that  $\sum_{k=1}^{\infty} q_{ij}^k < 1$ . This will occur, for example, if  $j$  cannot be reached from  $i$ , in which case  $q_{ij}^k = 0$  for all  $k = 1, 2, \dots$ . The *mean first passage time* from  $i$  to  $j$  is the expected value of  $K_{ij}$ :

$$E\{K_{ij}\} = \begin{cases} \sum_{k=1}^{\infty} k q_{ij}^k & \text{if } \sum_{k=1}^{\infty} q_{ij}^k = 1, \\ \infty & \text{if } \sum_{k=1}^{\infty} q_{ij}^k < 1. \end{cases}$$

It may be proved that if  $i$  and  $j$  belong to the same recurrent class then

$$E\{K_{ij}\} < \infty.$$

In fact if there is only one recurrent class and  $t$  is a state of that class, the mean first passage times  $E\{K_{it}\}$  are the unique solution of the following linear system of equations

$$E\{K_{it}\} = 1 + \sum_{j=1, j \neq t}^n p_{ij} E\{K_{jt}\}, \quad i = 1, \dots, n, i \neq t;$$

see Example 7.2.1. If  $i$  and  $j$  belong to two different recurrent classes, then  $E\{K_{ij}\} = E\{K_{ji}\} = \infty$ . If  $i$  belongs to a recurrent class and  $j$  is transient, we have  $E\{K_{ij}\} = \infty$ .

## APPENDIX E: *Kalman Filtering*

In this appendix we present the basic principles of least-squares estimation and their application in estimating the state of a linear discrete-time dynamic system using measurements that are linear in the state variables.

Fundamentally, the problem is the following. There are two random vectors  $x$  and  $y$ , which are related through their joint probability distribution so that the value of one provides information about the value of the other. We get to know the value of  $y$ , and we want to estimate the value of  $x$  so that the average squared error between  $x$  and its estimate is minimized. A related problem is to find the best estimate of  $x$  within the class of all estimates that are *linear* in the measured vector  $y$ . We will specialize these problems to a case where there is an underlying linear dynamic system. In particular, we will estimate the state of the system using measurements that are obtained sequentially in time. By exploiting the special structure of the problem, the computation of the state estimate can be organized conveniently in a recursive algorithm – the Kalman filter.

### E.1 LEAST-SQUARES ESTIMATION

Consider two jointly distributed random vectors  $x$  and  $y$  taking values in  $\mathbb{R}^n$  and  $\mathbb{R}^m$ , respectively. We view  $y$  as a measurement that provides some information about  $x$ . Thus, while prior to knowing  $y$  our estimate of  $x$  may

have been the expected value  $E\{x\}$ , once the value of  $y$  is known, we want to form an updated estimate  $x(y)$  of the value  $x$ . This updated estimate depends, of course, on the value of  $y$ , so we are interested in a rule that gives us the estimate for each possible value of  $y$ , i.e., we are interested in a function  $x(\cdot)$ , where  $x(y)$  is the estimate of  $x$  given  $y$ . Such a function  $x(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is called an *estimator*. We are seeking an estimator that is optimal in some sense and the criterion we shall employ is based on minimization of

$$E_{x,y}\{\|x - x(y)\|^2\}. \quad (\text{E.1})$$

Here,  $\|\cdot\|$  denotes the usual norm in  $\mathbb{R}^n$  ( $\|z\|^2 = z'z$  for  $z \in \mathbb{R}^n$ ). Furthermore, throughout the appendix, we assume that all encountered expected values are finite.

An estimator that minimizes the expected squared error above over all  $x(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is called a *least-squares estimator* and is denoted by  $x^*(\cdot)$ . Since

$$E_{x,y}\{\|x - x(y)\|^2\} = E_y\left\{E_x\{\|x - x(y)\|^2 | y\}\right\},$$

it is clear that  $x^*(\cdot)$  is a least-squares estimator if  $x^*(y)$  minimizes the conditional expectation in the right-hand side above for every  $y \in \mathbb{R}^m$ , that is,

$$E_x\{\|x - x^*(y)\|^2 | y\} = \min_{z \in \mathbb{R}^m} E_x\{\|x - z\|^2 | y\}, \quad \text{for all } y \in \mathbb{R}^m. \quad (\text{E.2})$$

By carrying out this minimization, we obtain the following proposition.

**Proposition E.1:** The least-squares estimator  $x^*(\cdot)$  is given by

$$x^*(y) = E_x\{x | y\}, \quad \text{for all } y \in \mathbb{R}^m. \quad (\text{E.3})$$

**Proof:** We have for every fixed  $z \in \mathbb{R}^n$

$$E_x\{\|x - z\|^2 | y\} = E_x\{\|x\|^2 | y\} - 2z'E_x\{x | y\} + \|z\|^2.$$

By setting to zero the derivative with respect to  $z$ , we see that the above expression is minimized by  $z = E_x\{x | y\}$ , and the result follows. **Q.E.D.**

## E.2 LINEAR LEAST-SQUARES ESTIMATION

The least-squares estimator  $E_x\{x | y\}$  may be a complicated nonlinear function of  $y$ . As a result its practical calculation may be difficult. This motivates finding optimal estimators within the restricted class of *linear* estimators, i.e., estimators of the form

$$x(y) = Ay + b, \quad (\text{E.4})$$

where  $A$  is an  $n \times m$  matrix and  $b$  is an  $n$ -dimensional vector. An estimator

$$\hat{x}(y) = \hat{A}y + \hat{b}$$

where  $\hat{A}$  and  $\hat{b}$  minimize

$$E_{x,y}\{\|x - Ay - b\|^2\}$$

over all  $n \times m$  matrices  $A$  and vectors  $b \in \mathbb{R}^n$  is called a *linear least-squares estimator*.

In the special case where  $x$  and  $y$  are jointly Gaussian random vectors it turns out that the conditional expectation  $E_x\{x | y\}$  is a linear function of  $y$  (plus a constant vector), and as a result, a linear least-squares estimator is also a least-squares estimator. This is shown in the next proposition.

**Proposition E.2:** If  $x, y$  are jointly Gaussian random vectors, then the least-squares estimate  $E_x\{x | y\}$  of  $x$  given  $y$  is linear in  $y$ .

**Proof:** Consider the random vector  $z \in \mathbb{R}^{n+m}$

$$z = \begin{pmatrix} x \\ y \end{pmatrix}$$

and assume that  $z$  is Gaussian with mean

$$\bar{z} = E\{z\} = \begin{pmatrix} E\{x\} \\ E\{y\} \end{pmatrix} = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \quad (\text{E.5})$$

and covariance matrix

$$\begin{aligned} \Sigma &= E\{(z - \bar{z})(z - \bar{z})'\} = \begin{pmatrix} E\{(x - \bar{x})(x - \bar{x})'\} & E\{(x - \bar{x})(y - \bar{y})'\} \\ E\{(y - \bar{y})(x - \bar{x})'\} & E\{(y - \bar{y})(y - \bar{y})'\} \end{pmatrix} \\ &= \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}. \end{aligned} \quad (\text{E.6})$$

To simplify our proof we assume that  $\Sigma$  is a positive definite symmetric matrix so that it possesses an inverse; the result, however, holds without this assumption. We recall that if  $z$  is Gaussian, its probability density function is of the form

$$p(z) = p(x, y) = ce^{-\frac{1}{2}(z-\bar{z})'\Sigma^{-1}(z-\bar{z})},$$

where

$$c = (2\pi)^{-(n+m)/2}(\det \Sigma)^{-1/2}$$

and  $\det \Sigma$  denotes the determinant of  $\Sigma$ . Similarly the probability density functions of  $x$  and  $y$  are of the form

$$p(x) = c_1 e^{-\frac{1}{2}(x-\bar{x})'\Sigma_{xx}^{-1}(x-\bar{x})},$$

$$p(y) = c_2 e^{-\frac{1}{2}(y-\bar{y})'\Sigma_{yy}^{-1}(y-\bar{y})},$$

where  $c_1$  and  $c_2$  are appropriate constants. By Bayes' rule the conditional probability density function of  $x$  given  $y$  is

$$p(x | y) = \frac{p(x, y)}{p(y)} = \frac{c}{c_2} e^{-\frac{1}{2}((z-\bar{z})'\Sigma^{-1}(z-\bar{z}) - (y-\bar{y})'\Sigma_{yy}^{-1}(y-\bar{y}))}. \quad (\text{E.7})$$

It can now be seen that there exist a positive definite symmetric  $n \times n$  matrix  $D$ , an  $n \times m$  matrix  $A$ , a vector  $b \in \mathbb{R}^n$ , and a scalar  $s$  such that

$$(z-\bar{z})'\Sigma^{-1}(z-\bar{z}) - (y-\bar{y})'\Sigma_{yy}^{-1}(y-\bar{y}) = (x - Ay - b)'D^{-1}(x - Ay - b) + s. \quad (\text{E.8})$$

This is because by substitution of the expressions for  $\bar{z}$  and  $\Sigma$  of Eqs. (E.5) and (E.6), the left-hand side of Eq. (E.8) becomes a quadratic form in  $x$  and  $y$ , which can be put in the form indicated in the right-hand side of Eq. (E.8). In fact, by computing the inverse of  $\Sigma$  using the partitioned matrix inversion formula (Appendix A) it can be verified that  $A$ ,  $b$ ,  $D$ , and  $s$  in Eq. (E.8) have the form

$$A = \Sigma_{xy}\Sigma_{yy}^{-1}, \quad b = \bar{x} - \Sigma_{xy}\Sigma_{yy}^{-1}\bar{y}, \quad D = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}, \quad s = 0.$$

Now it follows from Eqs. (E.8) and (E.7) that the conditional expectation  $E_x\{x | y\}$  is of the form  $Ay + b$ , where  $A$  is some  $n \times m$  matrix and  $b \in \mathbb{R}^n$ . **Q.E.D.**

We now turn to the characterization of the linear least-squares estimator.

**Proposition E.3:** Let  $x, y$  be random vectors taking values in  $\mathbb{R}^n$  and  $\mathbb{R}^m$ , respectively, with given joint probability distribution. The expected values and covariance matrices of  $x, y$  are denoted by

$$E\{x\} = \bar{x} \quad E\{y\} = \bar{y}, \quad (\text{E.9})$$

$$E\{(x - \bar{x})(x - \bar{x})'\} = \Sigma_{xx}, \quad E\{(y - \bar{y})(y - \bar{y})'\} = \Sigma_{yy}, \quad (\text{E.10})$$

$$E\{(x - \bar{x})(y - \bar{y})'\} = \Sigma_{xy}, \quad E\{(y - \bar{y})(x - \bar{x})'\} = \Sigma_{yx}, \quad (\text{E.11})$$

and we assume that  $\Sigma_{yy}$  is invertible. Then the linear least-squares estimator of  $x$  given  $y$  is

$$\hat{x}(y) = \bar{x} + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \bar{y}). \quad (\text{E.12})$$

The corresponding error covariance matrix is given by

$$E_{x,y}\{(x - \hat{x}(y))(x - \hat{x}(y))'\} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}. \quad (\text{E.13})$$

**Proof:** The linear least-squares estimator is defined as

$$\hat{x}(y) = \hat{A}y + \hat{b},$$

where  $\hat{A}, \hat{b}$  minimize the function  $f(A, b) = E_{x,y}\{\|x - Ay - b\|^2\}$  over  $A$  and  $b$ . Taking the derivatives of  $f(A, b)$  with respect to  $A$  and  $b$  and setting them to zero, we obtain the two conditions

$$0 = \frac{\partial f}{\partial A}\Big|_{\hat{A}, \hat{b}} = 2 E_{x,y}\{(\hat{b} + \hat{A}y - x)y'\}, \quad (\text{E.14})$$

$$0 = \frac{\partial f}{\partial b}\Big|_{\hat{A}, \hat{b}} = 2 E_{x,y}\{\hat{b} + \hat{A}y - x\}. \quad (\text{E.15})$$

The second condition yields

$$\hat{b} = \bar{x} - \hat{A}\bar{y}, \quad (\text{E.16})$$

and by substitution in the first, we obtain

$$E_{x,y}\{y(\hat{A}(y - \bar{y}) - (x - \bar{x}))'\} = 0. \quad (\text{E.17})$$

We have

$$E_{x,y}\{\hat{A}(y - \bar{y}) - (x - \bar{x})\}' = 0,$$

so that

$$\bar{y} \underset{x,y}{E} \{ \hat{A}(y - \bar{y}) - (x - \bar{x}) \}' = 0. \quad (\text{E.18})$$

By subtracting Eq. (E.18) from Eq. (E.17), we obtain

$$\underset{x,y}{E} \{ (y - \bar{y})(\hat{A}(y - \bar{y}) - (x - \bar{x}))' \} = 0.$$

Equivalently,

$$\Sigma_{yy} \hat{A}' - \Sigma_{yx} = 0,$$

from which

$$\hat{A} = \Sigma_{yy}' \Sigma_{yy}^{-1} = \Sigma_{xy} \Sigma_{yy}^{-1}. \quad (\text{E.19})$$

Using the expressions (E.16) and (E.19) for  $\hat{b}$  and  $\hat{A}$ , respectively, we obtain

$$\hat{x}(y) = \hat{A}y + \hat{b} = \bar{x} + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \bar{y}),$$

which was to be proved. The desired Eq. (E.13) for the error covariance follows upon substitution of the expression for  $\hat{x}(y)$  obtained above. Q.E.D.

We list some of the properties of the least-squares estimator as corollaries.

**Corollary E.3.1:** The linear least-squares estimator is unbiased, i.e.,

$$\underset{y}{E} \{ \hat{x}(y) \} = \bar{x}.$$

**Proof:** This follows from Eq. (E.12). Q.E.D.

**Corollary E.3.2:** The estimation error  $x - \hat{x}(y)$  is uncorrelated with both  $y$  and  $\hat{x}(y)$ , i.e.,

$$\underset{x,y}{E} \{ y(x - \hat{x}(y))' \} = 0,$$

$$\underset{x,y}{E} \{ \hat{x}(y)(x - \hat{x}(y))' \} = 0.$$

**Proof:** The first equality is Eq. (E.14). The second equality can be written as

$$\underset{x,y}{E} \{ (\hat{A}y + \hat{b})(x - \hat{x}(y))' \} = 0$$

and follows from the first equality and Cor. E.3.1. Q.E.D.

Corollary E.3.2 is known as the *orthogonal projection principle*. It states a property that characterizes the linear least-squares estimate and forms the basis for an alternative treatment of least-squares estimation as a problem of projection in a Hilbert space of random variables (see Luenberger [Lue69]).

**Corollary E.3.3:** Consider in addition to  $x$  and  $y$ , the random vector  $z$  defined by

$$z = Cx,$$

where  $C$  is a given  $p \times m$  matrix. Then the linear least-squares estimate of  $z$  given  $y$  is

$$\hat{z}(y) = C\hat{x}(y),$$

and the corresponding error covariance matrix is given by

$$\underset{z,y}{E} \{ (z - \hat{z}(y))(z - \hat{z}(y))' \} = C \underset{x,y}{E} \{ (x - \hat{x}(y))(x - \hat{x}(y))' \} C'.$$

**Proof:** We have  $E\{z\} = \bar{z} = C\bar{x}$  and

$$\Sigma_{zz} = \underset{z}{E} \{ (z - \bar{z})(z - \bar{z})' \} = C\Sigma_{xx}C',$$

$$\Sigma_{zy} = \underset{z,y}{E} \{ (z - \bar{z})(y - \bar{y})' \} = C\Sigma_{xy},$$

$$\Sigma_{yz} = \Sigma_{zy}' = \Sigma_{yx}C'.$$

By Prop. E.3 we have

$$\hat{z}(y) = \bar{z} + \Sigma_{zy} \Sigma_{yy}^{-1} (y - \bar{y}) = C\bar{x} + C\Sigma_{xy} \Sigma_{yy}^{-1} (y - \bar{y}) - C\hat{x}(y),$$

$$\begin{aligned} \underset{x,y}{E} \{ (z - \hat{z}(y))(z - \hat{z}(y))' \} &= \Sigma_{zz} - \Sigma_{zy} \Sigma_{yy}^{-1} \Sigma_{yz} \\ &= C(\Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx})C' \\ &= C \underset{x,y}{E} \{ (x - \hat{x}(y))(x - \hat{x}(y))' \} C'. \end{aligned}$$

Q.E.D.

**Corollary E.3.4:** Consider in addition to  $x$  and  $y$ , an additional random vector  $z$  of the form

$$z = Cy + u, \quad (\text{E.20})$$

where  $C$  is a given  $p \times m$  matrix of rank  $p$  and  $u$  is a given vector in  $\mathbb{R}^p$ . Then the linear least-squares estimate  $\hat{x}(z)$  of  $x$  given  $z$  is

$$\hat{x}(z) = \bar{x} + \Sigma_{xz} C' (C \Sigma_{yy} C')^{-1} (z - C \bar{y} - u), \quad (\text{E.21})$$

and the corresponding error covariance matrix is

$$E_{x,z} \left\{ (x - \hat{x}(z)) (x - \hat{x}(z))' \right\} = \Sigma_{xx} - \Sigma_{xy} C' (C \Sigma_{yy} C')^{-1} C \Sigma_{yx}. \quad (\text{E.22})$$

**Proof:** We have

$$\bar{z} = E\{z\} = C \bar{y} + u, \quad (\text{E.23a})$$

$$\Sigma_{zz} = E\{(z - \bar{z})(z - \bar{z})'\} = C \Sigma_{yy} C', \quad (\text{E.23b})$$

$$\Sigma_{zx} = E\{(z - \bar{z})(x - \bar{x})'\} = C \Sigma_{yx}, \quad (\text{E.23c})$$

$$\Sigma_{xz} = E\{(x - \bar{x})(z - \bar{z})'\} = \Sigma_{xy} C'. \quad (\text{E.23d})$$

From Prop. E.3 we have

$$\hat{x}(z) = \bar{x} + \Sigma_{xz} \Sigma_{zz}^{-1} (z - \bar{z}), \quad (\text{E.24a})$$

$$E_{x,z} \left\{ (x - \hat{x}(z)) (x - \hat{x}(z))' \right\} = \Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} \Sigma_{zx}, \quad (\text{E.24b})$$

where  $\Sigma_{zz} = C \Sigma_{yy} C'$  has an inverse, since  $\Sigma_{yy}$  is invertible and  $C$  has rank  $p$ . By substituting the relations (E.23) into Eqs. (E.24a) and (E.24b) the result follows. Q.E.D.

Frequently we want to estimate a vector of parameters  $x \in \mathbb{R}^n$  given a measurement vector  $z \in \mathbb{R}^m$  of the form  $z = Cx + v$ , where  $C$  is a given  $m \times n$  matrix, and  $v \in \mathbb{R}^m$  is a random measurement error vector. The following corollary gives the linear least-squares estimate  $\hat{x}(z)$  and its error covariance.

**Corollary E.3.5:** Let

$$z = Cx + v,$$

where  $C$  is a given  $m \times n$  matrix, and the random vectors  $x \in \mathbb{R}^n$  and  $v \in \mathbb{R}^m$  are uncorrelated. Denote

$$E\{x\} = \bar{x}, \quad E\{(x - \bar{x})(x - \bar{x})'\} = \Sigma_{xx},$$

$$E\{v\} = \bar{v}, \quad E\{(v - \bar{v})(v - \bar{v})'\} = \Sigma_{vv},$$

and assume further that  $\Sigma_{vv}$  is a positive definite matrix. Then

$$\hat{x}(z) = \bar{x} + \Sigma_{xx} C' (C \Sigma_{yy} C' + \Sigma_{vv})^{-1} (z - C \bar{y} - \bar{v}),$$

$$E_{x,v} \left\{ (x - \hat{x}(z)) (x - \hat{x}(z))' \right\} = \Sigma_{xx} - \Sigma_{xx} C' (C \Sigma_{yy} C' + \Sigma_{vv})^{-1} C \Sigma_{yy}. \quad (\text{E.25})$$

**Proof:** Define

$$y = (x' \quad v')', \quad \bar{y} = (\bar{x}' \quad \bar{v}')', \quad \tilde{C} = (C \quad I).$$

Then we have  $z = \tilde{C}y$ , and by Cor. E.3.3,

$$\hat{x}(z) = (I \quad 0) \hat{y}(z),$$

$$E \left\{ (x - \hat{x}(z)) (x - \hat{x}(z))' \right\} = (I \quad 0) E \left\{ (y - \hat{y}(z)) (y - \hat{y}(z))' \right\} \begin{pmatrix} I \\ 0 \end{pmatrix},$$

where  $\hat{y}(z)$  is the linear least-squares estimate of  $y$  given  $z$ . By applying Cor. E.3.4 with  $u = 0$  and  $x = y$  we obtain

$$\hat{y}(z) = \bar{y} + \Sigma_{yy} \tilde{C}' (\tilde{C} \Sigma_{yy} \tilde{C}')^{-1} (z - \tilde{C} \bar{y}),$$

$$E \left\{ (y - \hat{y}(z)) (y - \hat{y}(z))' \right\} = \Sigma_{yy} - \Sigma_{yy} \tilde{C}' (\tilde{C} \Sigma_{yy} \tilde{C}')^{-1} \tilde{C} \Sigma_{yy}.$$

By using the equations

$$\Sigma_{yy} = \begin{pmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{vv} \end{pmatrix}, \quad \tilde{C} = (C \quad I),$$

and by carrying out the straightforward calculation the result follows. Q.E.D.

The next two corollaries deal with least-squares estimates involving multiple measurement vectors that are obtained sequentially. In particular,

the corollaries show how to modify an existing least-squares estimate  $\hat{x}(y)$  to obtain  $\hat{x}(y, z)$  once an additional vector  $z$  becomes known. This is a central operation in Kalman filtering.

**Corollary E.3.6:** Consider in addition to  $x$  and  $y$ , an additional random vector  $z$  taking values in  $\Re^p$ , which is uncorrelated with  $y$ . Then the linear least-squares estimate  $\hat{x}(y, z)$  of  $x$  given  $y$  and  $z$  [i.e., given the composite vector  $(y, z)$ ] has the form

$$\hat{x}(y, z) = \hat{x}(y) + \hat{x}(z) - \bar{x}, \quad (\text{E.25})$$

where  $\hat{x}(y)$  and  $\hat{x}(z)$  are the linear least-squares estimates of  $x$  given  $y$  and given  $z$ , respectively. Furthermore,

$$E_{x,y,z} \left\{ (x - \hat{x}(y, z))(x - \hat{x}(y, z))' \right\} = \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} - \Sigma_{xz} \Sigma_{zz}^{-1} \Sigma_{zx}, \quad (\text{E.26})$$

where

$$\Sigma_{xz} = E_{x,z} \left\{ (x - \bar{x})(z - \bar{z})' \right\}, \quad \Sigma_{zx} = E_{z,x} \left\{ (z - \bar{z})(x - \bar{x})' \right\},$$

$$\Sigma_{zz} = E_z \left\{ (z - \bar{z})(z - \bar{z})' \right\}, \quad \bar{z} = E_z \{z\},$$

and it is assumed that  $\Sigma_{zz}$  is invertible.

**Proof:** Let

$$w = \begin{pmatrix} y \\ z \end{pmatrix}, \quad \bar{w} = \begin{pmatrix} \bar{y} \\ \bar{z} \end{pmatrix}.$$

By Eq. (E.12) we have

$$\hat{x}(w) = \bar{x} + \Sigma_{xw} \Sigma_{ww}^{-1} (w - \bar{w}). \quad (\text{E.27})$$

Furthermore

$$\Sigma_{xw} = [\Sigma_{xy}, \Sigma_{xz}],$$

and since  $y$  and  $z$  are uncorrelated, we have

$$\Sigma_{ww} = \begin{pmatrix} \Sigma_{yy} & 0 \\ 0 & \Sigma_{zz} \end{pmatrix}.$$

Substituting the above expressions in Eq. (E.27), we obtain

$$\hat{x}(w) = \bar{x} + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \bar{y}) + \Sigma_{xz} \Sigma_{zz}^{-1} (z - \bar{z}) = \hat{x}(y) + \hat{x}(z) - \bar{x},$$

and Eq. (E.25) is proved. The proof of Eq. (E.26) is similar by using the relations above and the covariance Eq. (E.13). **Q.E.D.**

**Corollary E.3.7:** Let  $z$  be as in the preceding corollary and assume that  $y$  and  $z$  are not necessarily uncorrelated, that is, we may have

$$\Sigma_{yz} = \Sigma_{zy} = E_{y,z} \{ (y - \bar{y})(z - \bar{z})' \} \neq 0.$$

Then

$$\hat{x}(y, z) = \hat{x}(y) + \hat{x}(z - \hat{z}(y)) - \bar{x}, \quad (\text{E.28})$$

where  $\hat{x}(z - \hat{z}(y))$  denotes the linear least-squares estimate of  $x$  given the random vector  $z - \hat{z}(y)$  and  $\hat{z}(y)$  is the linear least-squares estimate of  $z$  given  $y$ . Furthermore,

$$E_{x,y,z} \left\{ (x - \hat{x}(y, z))(x - \hat{x}(y, z))' \right\} = \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} - \hat{\Sigma}_{xz} \hat{\Sigma}_{zz}^{-1} \hat{\Sigma}_{zx}, \quad (\text{E.29})$$

where

$$\hat{\Sigma}_{xz} = E_{x,y,z} \left\{ (x - \bar{x})(z - \hat{z}(y))' \right\},$$

$$\hat{\Sigma}_{zz} = E_{y,z} \left\{ (z - \hat{z}(y))(z - \hat{z}(y))' \right\},$$

$$\hat{\Sigma}_{zx} = E_{x,y,z} \left\{ (z - \hat{z}(y))(x - \bar{x})' \right\}.$$

**Proof:** It can be seen that, since  $\hat{z}(y)$  is a linear function of  $y$ , the linear least-squares estimate of  $x$  given  $y$  and  $z$  is the same as the linear least-squares estimate of  $x$  given  $y$  and  $z - \hat{z}(y)$ . By Cor. E.3.2 the random vectors  $y$  and  $z - \hat{z}(y)$  are uncorrelated. Given this observation the result follows by application of the preceding corollary. **Q.E.D.**

### E.3 STATE ESTIMATION — THE KALMAN FILTER

Consider now a linear dynamic system of the type considered in Section 5.2 but without a control vector ( $u_k \equiv 0$ )

$$x_{k+1} = A_k x_k + w_k, \quad k = 0, 1, \dots, N-1, \quad (\text{E.30})$$

where  $x_k \in \Re^n$  and  $w_k \in \Re^n$  denote the state and random disturbance vectors, respectively, and the matrices  $A_k$  are known. Consider also the

measurement equation

$$z_k = C_k x_k + v_k, \quad k = 0, 1, \dots, N-1, \quad (\text{E.31})$$

where  $z_k \in \mathbb{R}^s$  and  $v_k \in \mathbb{R}^s$  are the observation and observation noise vectors, respectively.

We assume that  $x_0, w_0, \dots, w_{N-1}, v_0, \dots, v_{N-1}$  are independent random vectors with given probability distributions and that

$$E\{w_k\} = E\{v_k\} = 0, \quad k = 0, 1, \dots, N-1. \quad (\text{E.32})$$

We use the notation

$$S = E\{(x_0 - E\{x_0\})(x_0 - E\{x_0\})'\}, \quad M_k = E\{w_k w_k'\}, \quad N_k = E\{v_k v_k'\}, \quad (\text{E.33})$$

and we assume that  $N_k$  is positive definite for all  $k$ .

### A Nonrecursive Least-Squares Estimate

We first give a straightforward but somewhat tedious method to derive the linear least-squares estimate of  $x_{k+1}$  or  $x_k$  given the values of  $z_0, z_1, \dots, z_k$ . Let us denote

$$Z_k = (z'_0, z'_1, \dots, z'_k)', \quad r_{k-1} = (x'_0, w'_0, w'_1, \dots, w'_{k-1})'.$$

In this method, we first find the linear least-squares estimate of  $r_{k-1}$  given  $Z_k$ , and we then obtain the linear least-squares estimate of  $x_k$  given  $Z_k$  after expressing  $x_k$  as a linear function of  $r_{k-1}$ .

For each  $i$  with  $0 \leq i \leq k$  we have, by using the system equation,

$$x_{i+1} = L_i r_i,$$

where  $L_i$  is the  $n \times (n(i+1))$  matrix

$$L_i = (A_i \cdots A_0, \quad A_i \cdots A_1, \quad \dots, \quad A_i, \quad I).$$

As a result we may write

$$Z_k = \Phi_{k-1} r_{k-1} + V_k,$$

where

$$V_k = (v'_0, v'_1, \dots, v'_k)',$$

and  $\Phi_{k-1}$  is an  $s(k+1) \times (nk)$  matrix of the form

$$\Phi_{k-1} = \begin{pmatrix} C_0 & 0 \\ C_1 L_0 & 0 \\ \vdots & \vdots \\ C_{k-1} L_{k-2} & 0 \\ C_k L_{k-1} & \end{pmatrix}.$$

We can thus use Cor. E.3.5, the equations above, and the data of the problem to compute

$$\hat{r}_{k-1}(Z_k) \quad \text{and} \quad E\{(r_{k-1} - \hat{r}_{k-1}(Z_k))(r_{k-1} - \hat{r}_{k-1}(Z_k))'\}.$$

Let us denote the linear least-squares estimates of  $x_{k+1}$  and  $x_k$  given  $Z_k$  by  $\hat{x}_{k-1|k}$  and  $\hat{x}_{k|k}$ , respectively. We can now obtain  $\hat{x}_{k|k} = \hat{x}_k(Z_k)$  and the corresponding error covariance matrix by using Cor. E.3.3, that is,

$$\begin{aligned} \hat{x}_{k|k} &= L_{k-1} \hat{r}_{k-1}(Z_k), \\ E\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'\} \\ &= L_{k-1} E\{(r_{k-1} - \hat{r}_{k-1}(Z_k))(r_{k-1} - \hat{r}_{k-1}(Z_k))'\} L'_{k-1}. \end{aligned}$$

These equations may in turn be used to yield  $\hat{x}_{k+1|k}$  and the corresponding error covariance again via Cor. E.3.3.

### The Kalman Filtering Algorithm

The preceding method for obtaining the least-squares estimate of  $x_k$  is cumbersome when the number of measurements is large. Fortunately, the sequential structure of the problem can be exploited and the computations can be organized conveniently, as first proposed by Kalman [Kal60]. The main attractive feature of the Kalman filtering algorithm is that the estimate  $\hat{x}_{k-1|k}$  can be obtained by means of a simple equation that involves the previous estimate  $\hat{x}_{k|k-1}$  and the new measurement  $z_k$  but *does not involve any of the past measurements  $z_0, z_1, \dots, z_{k-1}$* .

Suppose that we have computed the estimate  $\hat{x}_{k|k-1}$  together with the covariance matrix

$$\Sigma_{k|k-1} = E\{(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})'\}. \quad (\text{E.34})$$

At time  $k$  we receive the additional measurement

$$z_k = C_k x_k + v_k.$$

We may use now Cor. E.3.7 to compute the linear least-squares estimate of  $x_k$  given  $Z_{k-1} = (z'_0, z'_1, \dots, z'_{k-1})'$  and  $z_k$ . This estimate is denoted by  $\hat{x}_{k|k}$  and, by Cor. E.3.7, it is given by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \hat{x}_k(z_k - \hat{z}_k(Z_{k-1})) - E\{x_k\}, \quad (\text{E.35})$$

where  $\hat{z}_k(Z_{k-1})$  denotes the linear least-squares estimate of  $z_k$  given  $Z_{k-1}$  and  $\hat{x}_k(z_k - \hat{z}_k(Z_{k-1}))$  denotes the linear least-squares estimate of  $x_k$  given  $(z_k - \hat{z}_k(Z_{k-1}))$ .

We now calculate the term  $\hat{x}_k(z_k - \hat{z}_k(Z_{k-1}))$  in Eq. (E.35). We have by Eqs. (E.31), (E.32), and Cor. E.3.3,

$$\hat{z}_k(Z_{k-1}) = C_k \hat{x}_{k|k-1}. \quad (\text{E.36})$$

Also we use Cor. E.3.3 to obtain

$$E\{(z_k - \hat{z}_k(Z_{k-1}))(z_k - \hat{z}_k(Z_{k-1}))'\} = C_k \Sigma_{k|k-1} C_k' + N_k, \quad (\text{E.37})$$

$$\begin{aligned} E\{x_k(z_k - \hat{z}_k(Z_{k-1}))'\} \\ = E\{x_k(C_k(x_k - \hat{x}_{k|k-1}))'\} + E\{x_k v_k'\} \\ = E\{(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})'\} C_k' + E\{\hat{x}_{k|k-1}(x_k - \hat{x}_{k|k-1})'\} C_k'. \end{aligned}$$

The last term in the right-hand side above is zero by Cor. E.3.2, so by using Eq. (E.34) we have

$$E\{x_k(z_k - \hat{z}_k(Z_{k-1}))'\} = \Sigma_{k|k-1} C_k'. \quad (\text{E.38})$$

Using Eqs. (E.36)-(E.38) in Prop. E.3, we obtain

$$\hat{x}_k(z_k - \hat{z}_k(Z_{k-1})) = E\{x_k\} + \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} (z_k - C_k \hat{x}_{k|k-1}),$$

and Eq. (E.35) is written as

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} (z_k - C_k \hat{x}_{k|k-1}). \quad (\text{E.39})$$

By using Cor. E.3.3 we also have

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k}. \quad (\text{E.40})$$

Concerning the covariance matrix  $\Sigma_{k+1|k}$ , we have from Eqs. (E.30), (E.32), (E.33), and Cor. E.3.3,

$$\Sigma_{k+1|k} = A_k \Sigma_{k|k} A_k' + M_k, \quad (\text{E.41})$$

where

$$\Sigma_{k|k} = E\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'\}.$$

The error covariance matrix  $\Sigma_{k|k}$  may be computed via Cor. E.3.7 similar to  $\hat{x}_{k|k}$  [cf. Eq. (E.35)]. Thus, we have from Eqs. (E.29), (E.37), (E.38)

$$\Sigma_{k|k} = \Sigma_{k|k-1} - \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} C_k \Sigma_{k|k-1}. \quad (\text{E.42})$$

Equations (E.39)-(E.42) with the initial conditions [cf. Eq. (E.33)]

$$\hat{x}_{0|k-1} = E\{x_0\}, \quad \Sigma_{0|k-1} = S, \quad (\text{E.43})$$

constitute the *Kalman filtering algorithm*. This algorithm recursively generates the linear least-squares estimates  $\hat{x}_{k+1|k}$  or  $\hat{x}_{k|k}$  together with the associated error covariance matrices  $\Sigma_{k+1|k}$  or  $\Sigma_{k|k}$ . In particular, given  $\Sigma_{k|k-1}$  and  $\hat{x}_{k|k-1}$ , Eqs. (E.39) and (E.42) yield  $\Sigma_{k|k}$  and  $\hat{x}_{k|k}$ , and then Eqs. (E.41) and (E.40) yield  $\Sigma_{k+1|k}$  and  $\hat{x}_{k+1|k}$ .

An alternative expression for Eq. (E.39) is

$$\hat{x}_{k|k} = A_{k-1} \hat{x}_{k-1|k-1} + \Sigma_{k|k} C_k' N_k^{-1} (z_k - C_k A_{k-1} \hat{x}_{k-1|k-1}), \quad (\text{E.44})$$

which can be obtained from Eqs. (E.39) and (E.40) by using the following equality

$$\Sigma_{k|k} C_k' N_k^{-1} = \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1}. \quad (\text{E.45})$$

This equality may be verified by using Eq. (E.42) to write

$$\begin{aligned} \Sigma_{k|k} C_k' N_k^{-1} &= (\Sigma_{k|k-1} - \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} C_k \Sigma_{k|k-1}) C_k' N_k^{-1} \\ &= \Sigma_{k|k-1} C_k' (N_k^{-1} - (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} C_k \Sigma_{k|k-1} C_k' N_k^{-1}), \end{aligned}$$

and then use in the above formula the following calculation

$$\begin{aligned} N_k^{-1} &= (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} (C_k \Sigma_{k|k-1} C_k' + N_k) N_k^{-1} \\ &= (C_k \Sigma_{k|k-1} C_k' + N_k)^{-1} (C_k \Sigma_{k|k-1} C_k' N_k^{-1} + I). \end{aligned}$$

When the system equation contains a control vector  $u_k$ ,

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k = 0, 1, \dots, N-1,$$

it is straightforward to show that Eq. (E.44) takes the form

$$\begin{aligned} \hat{x}_{k|k} &= A_{k-1} \hat{x}_{k-1|k-1} - B_{k-1} u_{k-1} \\ &\quad + \Sigma_{k|k} C_k' N_k^{-1} (z_k - C_k A_{k-1} \hat{x}_{k-1|k-1} - C_k B_{k-1} u_{k-1}), \end{aligned} \quad (\text{E.46})$$

where  $\hat{x}_{k|k}$  is the linear least-squares estimate of  $x_k$  given  $z_0, z_1, \dots, z_k$  and  $u_0, u_1, \dots, u_{k-1}$ . The equations (E.41)-(E.43) that generate  $\Sigma_{k|k}$  remain unchanged.

### Steady-State Kalman Filtering Algorithm

Finally we note that Eqs. (E.41) and (E.42) yield

$$\Sigma_{k+1|k} = A_k (\Sigma_{k|k-1} - \Sigma_{k|k-1} C'_k (C_k \Sigma_{k|k-1} C'_k + N_k)^{-1} C_k \Sigma_{k|k-1}) A'_k + M_k, \quad (E.47)$$

with the initial condition  $\Sigma_{0|-1} = S$ . This equation is a matrix Riccati equation of the type considered in Section 4.1. Thus when  $A_k$ ,  $C_k$ ,  $N_k$ , and  $M_k$  are constant matrices,

$$A_k = A, \quad C_k = C, \quad N_k = N, \quad M_k = M, \quad k = 0, 1, \dots, N-1,$$

we have by invoking the proposition proved there, that  $\Sigma_{k+1|k}$  tends to a positive definite symmetric matrix  $\Sigma$  that solves the algebraic Riccati equation

$$\Sigma = A(\Sigma - \Sigma C'(C\Sigma C' + N)^{-1} C\Sigma) A' + M,$$

assuming observability of the pair  $(A, C)$  and controllability of the pair  $(A, D)$ , where  $M = DD'$ . Under the same conditions, we have  $\Sigma_{k|k} \rightarrow \bar{\Sigma}$ , where from Eq. (E.42),

$$\bar{\Sigma} = \Sigma - \Sigma C'(C\Sigma C' + N)^{-1} C\Sigma.$$

We may then write the Kalman filter recursion [cf. Eq. (E.44)] in the asymptotic form

$$\hat{x}_{k|k} = A\hat{x}_{k-1|k-1} + \bar{\Sigma}C'N^{-1}(z_k - CA\hat{x}_{k-1|k-1}). \quad (E.48)$$

This estimator is simple and convenient for implementation.

### STABILITY ASPECTS

Let us consider now the stability properties of the steady-state form of the Kalman filter. From Eqs. (E.39) and (E.40), we have

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + A\Sigma C'(C\Sigma C' + N)^{-1}(z_k - C\hat{x}_{k|k-1}). \quad (E.49)$$

Let  $e_k$  denote the “one-step prediction” error

$$e_k = x_k - \hat{x}_{k|k-1}.$$

By using Eq. (E.49), the system equation

$$x_{k+1} = Ax_k + w_k,$$

and the measurement equation

$$z_k = Cx_k + v_k,$$

we obtain

$$e_{k+1} = (A - A\Sigma C'(C\Sigma C' + N)^{-1} C)e_k + w_k - A\Sigma C'(C\Sigma C' + N)^{-1} v_k. \quad (E.50)$$

From the practical point of view it is important that the error equation (E.50) represents a stable system, that is, the matrix

$$A - A\Sigma C'(C\Sigma C' + N)^{-1} C \quad (E.51)$$

has eigenvalues strictly within the unit circle. This, however, follows by Prop. 4.4.1 of Section 4.1 under the observability and controllability assumptions given earlier, since  $\Sigma$  is the unique positive semidefinite symmetric solution of the algebraic Riccati equation

$$\Sigma = A(\Sigma - \Sigma C'(C\Sigma C' + N)^{-1} C\Sigma) A' + M.$$

Actually this proposition yields that the transpose of the matrix (E.51) has eigenvalues strictly within the unit circle, but this is sufficient for our purposes since the eigenvalues of a matrix are the same as those of its transpose.

Let us consider also the stability properties of the equation governing the estimation error

$$\tilde{e}_k = x_k - \hat{x}_{k|k}.$$

We have by a straightforward calculation

$$\tilde{e}_k = (I - \Sigma C'(C\Sigma C' + N)^{-1} C)e_k - \Sigma C'(C\Sigma C' + N)^{-1} v_k. \quad (E.52)$$

By multiplying both sides of Eq. (E.50) by  $I - \Sigma C'(C\Sigma C' + N)^{-1} C$  and by using Eq. (E.52), we obtain

$$\begin{aligned} \tilde{e}_{k+1} &+ \Sigma C'(C\Sigma C' + N)^{-1} v_{k+1} \\ &= (A - \Sigma C'(C\Sigma C' + N)^{-1} C A)(\tilde{e}_k + \Sigma C'(C\Sigma C' + N)^{-1} v_k) \\ &\quad + (I - \Sigma C'(C\Sigma C' + N)^{-1} C)(w_k - A\Sigma C'(C\Sigma C' + N)^{-1} v_k), \end{aligned}$$

or equivalently

$$\begin{aligned} \tilde{e}_{k+1} &= (A - \Sigma C'(C\Sigma C' + N)^{-1} C A)\tilde{e}_k \\ &\quad + (I - \Sigma C'(C\Sigma C' + N)^{-1} C)w_k - \Sigma C'(C\Sigma C' + N)^{-1} v_{k+1}. \end{aligned} \quad (E.53)$$

Since the matrix (E.51) has eigenvalues strictly within the unit circle, the sequence  $\{e_k\}$  generated by Eq. (E.50) tends to zero whenever the vectors  $w_k$  and  $v_k$  are identically zero for all  $k$ . Hence, by Eq. (E.52), the same is true for the sequence  $\{\tilde{e}_k\}$ . It follows from Eq. (E.53) that the matrix

$$A = \Sigma C' (C \Sigma C' + N)^{-1} CA \quad (\text{E.54})$$

has eigenvalues strictly within the unit circle, and the estimation error sequence  $\{\tilde{e}_k\}$  is generated by a stable system.

Let us finally consider the stability properties of the  $2n$ -dimensional system of equations with state vector  $(x'_k, \hat{x}'_k)$ :

$$x_{k+1} = Ax_k + BL\hat{x}_k, \quad (\text{E.55})$$

$$\hat{x}_{k+1} = \bar{\Sigma} C' N^{-1} C A x_k + (A + BL - \bar{\Sigma} C' N^{-1} C A) \hat{x}_k. \quad (\text{E.56})$$

This is the steady-state, asymptotically optimal closed-loop system that was encountered at the end of Section 5.2.

We assume that the appropriate observability and controllability assumptions stated there are in effect. By using the equation

$$\bar{\Sigma} C' N^{-1} = \Sigma C' (C \Sigma C' + N)^{-1},$$

shown earlier, we obtain from Eqs. (E.55) and (E.56) that

$$(x_{k+1} - \hat{x}_{k+1}) = (A - \Sigma C' (C \Sigma C' + N)^{-1} CA)(x_k - \hat{x}_k).$$

Since we have proved that the matrix (E.54) has eigenvalues strictly within the unit circle, it follows that

$$\lim_{k \rightarrow \infty} (x_{k+1} - \hat{x}_{k+1}) = 0, \quad (\text{E.57})$$

for arbitrary initial states  $x_0$  and  $\hat{x}_0$ . From Eq. (E.55) we obtain

$$x_{k+1} = (A + BL)x_k + BL(\hat{x}_k - x_k). \quad (\text{E.58})$$

Since in accordance with the theory of Section 4.1 the matrix  $(A + BL)$  has eigenvalues strictly within the unit circle, it follows from Eqs. (E.57) and (E.58) that we have

$$\lim_{k \rightarrow \infty} x_k = 0 \quad (\text{E.59})$$

and hence from Eq. (E.57),

$$\lim_{k \rightarrow \infty} \hat{x}_k = 0. \quad (\text{E.60})$$

Since the equations above hold for arbitrary initial states  $x_0$  and  $\hat{x}_0$  it follows that the system defined by Eqs. (E.55) and (E.56) is stable.

## E.5 GAUSS-MARKOV ESTIMATORS

Suppose that we want to estimate a vector  $x \in \mathbb{R}^n$  given a measurement vector  $z \in \mathbb{R}^m$  that is related to  $x$  by

$$z = Cx + v, \quad (\text{E.61})$$

where  $C$  is a given  $m \times n$  matrix with rank  $m$ , and  $v$  is a random measurement error vector. Let us assume that  $v$  is uncorrelated with  $x$ , and has a known mean and a positive definite covariance matrix

$$E\{v\} = \bar{v}, \quad E\{(v - \bar{v})(v - \bar{v})'\} = \Sigma_{vv}. \quad (\text{E.62})$$

If the a priori probability distribution of  $x$  is known, we can obtain a linear least-squares estimate of  $x$  given  $z$  by using the theory of Section E.2 (cf. Cor. E.3.5). In many cases, however, the probability distribution of  $x$  is unknown. In such cases we can use the Gauss-Markov estimator, which is optimal within the class of linear estimators that satisfy certain restrictions, as described below.

Let us consider an estimator of the form

$$\hat{x}(z) = \hat{A}(z - \bar{v}),$$

where  $\hat{A}$  minimizes

$$f(A) = E_{x,z}\{\|x - A(z - \bar{v})\|^2\} \quad (\text{E.63})$$

over all  $n \times m$  matrices  $A$ . Since  $x$  and  $v$  are uncorrelated, we have using Eqs. (E.61)-(E.63)

$$\begin{aligned} f(A) &= E_{x,v}\{\|x - ACx - A(v - \bar{v})\|^2\} \\ &= E_x\{\|(I - AC)x\|^2\} + E_v\{\|A(v - \bar{v})\|^2\}, \end{aligned}$$

where  $I$  is the  $n \times n$  identity matrix. Since  $f(A)$  depends on the unknown statistics of  $x$ , we see that the optimal matrix  $\hat{A}$  also depends on these statistics. We can circumvent this difficulty by requiring that

$$AC = I.$$

Then our problem becomes

$$\begin{aligned} &\text{minimize } E_v\{\|A(v - \bar{v})\|^2\} \\ &\text{subject to } AC = I. \end{aligned} \quad (\text{E.64})$$

Note that the requirement  $AC = I$  is not only convenient analytically, but also makes sense conceptually. In particular, it is equivalent to requiring that the estimator  $\hat{x}(z) = A(z - \bar{v})$  be *unbiased* in the sense that

$$E\{\hat{x}(z)\} = E\{x\} = \bar{x}, \quad \text{for all } \bar{x} \in \mathbb{R}^n.$$

This can be seen by writing

$$E\{\hat{x}(z)\} = E\{A(Cx + v - \bar{v})\} = ACE\{x\} = AC\bar{x} = \bar{x}.$$

To derive the optimal solution  $\hat{A}$  of problem (E.64), let  $a'_i$  denote the  $i$ th row of  $A$ . We have

$$\begin{aligned} \|A(v - \bar{v})\|^2 &= (v - \bar{v})' (a_1 \quad \cdots \quad a_n) \begin{pmatrix} a'_1 \\ \vdots \\ a'_n \end{pmatrix} (v - \bar{v}) \\ &= \sum_{i=1}^n (v - \bar{v})' a_i a'_i (v - \bar{v}) \\ &= \sum_{i=1}^n a'_i (v - \bar{v})(v - \bar{v})' a_i. \end{aligned}$$

Hence, the minimization problem (E.64) can also be written as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^n a'_i \Sigma_{vv} a_i \\ \text{subject to} \quad & C' a_i = e_i, \quad i = 1, \dots, n, \end{aligned}$$

where  $e_i$  is the  $i$ th column of the identity matrix. The minimization can be carried out separately for each  $i$ , yielding

$$\hat{x}_i = \Sigma_{vv}^{-1} C (C' \Sigma_{vv} C)^{-1} e_i, \quad i = 1, \dots, n,$$

and finally

$$\hat{A} = (C' \Sigma_{vv}^{-1} C)^{-1} C' \Sigma_{vv}^{-1}.$$

Thus, the Gauss-Markov estimator is given by

$$\hat{x}(z) = (C' \Sigma_{vv}^{-1} C)^{-1} C' \Sigma_{vv}^{-1} (z - \bar{v}). \quad (\text{E.65})$$

Let us also calculate the corresponding error covariance matrix. We have

$$\begin{aligned} E\{(x - \hat{x}(z))(x - \hat{x}(z))'\} &= E\{(x - \hat{A}(z - \bar{v}))(x - \hat{A}(z - \bar{v}))'\} \\ &= E\{\hat{A}(v - \bar{v})(v - \bar{v})' \hat{A}'\} \\ &= \hat{A} \Sigma_{vv} \hat{A}' \\ &= (C' \Sigma_{vv}^{-1} C)^{-1} C' \Sigma_{vv}^{-1} \Sigma_{vv} \Sigma_{vv}^{-1} C (C' \Sigma_{vv}^{-1} C)^{-1}, \end{aligned}$$

and finally

$$E\{(x - \hat{x}(z))(x - \hat{x}(z))'\} = (C' \Sigma_{vv}^{-1} C)^{-1}. \quad (\text{E.66})$$

Finally, let us compare the Gauss-Markov estimator with the linear least-squares estimator of Cor. E.3.5. Assuming that  $\Sigma_{xx}$  is invertible, a straightforward calculation shows that the latter estimator can be written as

$$\hat{x}(z) = \bar{x} + (\Sigma_{xx}^{-1} + C' \Sigma_{vv}^{-1} C)^{-1} C' \Sigma_{vv}^{-1} (z - C \bar{x} - \bar{v}). \quad (\text{E.67})$$

By comparing Eqs. (E.65) and (E.67), we see that the Gauss-Markov estimator is obtained from the linear least-squares estimator by setting  $\bar{x} = 0$  and  $\Sigma_{xx}^{-1} = 0$ , i.e., a zero mean and infinite covariance for the unknown random variable  $x$ . Thus, the Gauss-Markov estimator may be viewed as a limiting form of the linear least-squares estimator. The error covariance matrix (E.66) of the Gauss-Markov estimator is similarly related with the error covariance matrix of the linear least-squares estimator.

## E.6 DETERMINISTIC LEAST-SQUARES ESTIMATION

Suppose again that we want to estimate a vector  $x \in \mathbb{R}^n$  given a measurement vector  $z \in \mathbb{R}^m$  that is related to  $x$  by

$$z = Cx + v,$$

where  $C$  is a known  $m \times n$  matrix of rank  $m$ . However, we know nothing about the probability distribution of  $x$  and  $v$ , and thus we can't use a statistically-based estimator. Then it is reasonable to select as our estimate the vector  $\hat{x}$  that minimizes

$$f(x) = \|z - Cx\|^2,$$

that is, the estimate that fits best the data in a least-squares sense. We denote this estimate by  $\hat{x}(z)$ .

By setting to zero the gradient of  $f$  at  $\hat{x}(z)$ , we obtain

$$\nabla f|_{\hat{x}(z)} = 2C'(C\hat{x}(z) - z) = 0,$$

from which

$$\hat{x}(z) = (C'C)^{-1} C' z. \quad (\text{E.68})$$

An interesting observation is that the estimate (E.68) is the same as the Gauss-Markov estimate given by Eq. (E.65), provided the measurement error has zero mean and covariance matrix equal to the identity, i.e.,  $\bar{v} = 0$ ,  $\Sigma_{vv} = I$ . In fact, if instead of  $\|z - Cx\|^2$ , we minimize

$$(z - \bar{v} - Cx)' \Sigma_{vv}^{-1} (z - \bar{v} - Cx),$$

then the deterministic least-squares estimate obtained is identical to the Gauss-Markov estimate. If instead of  $\|z - Cx\|^2$  we minimize

$$(x - \bar{x})'\Sigma_{xx}^{-1}(z - \bar{x}) + (z - \bar{v} - Cx)'\Sigma_{vv}^{-1}(z - \bar{v} - Cx),$$

then the estimate obtained is identical to the linear least-squares estimate given by Eq. (E.67). Thus, we arrive at the interesting conclusion that the estimators obtained earlier on the basis of a stochastic optimization framework can also be obtained by minimization of a deterministic measure of fitness of estimated parameters to the data at hand.

## APPENDIX F: *Modeling of Stochastic Linear Systems*

In this appendix we show how controlled linear time-invariant systems with stochastic inputs can be represented by the ARMAX model used in Section 5.3.

### F.1 LINEAR SYSTEMS WITH STOCHASTIC INPUTS

Consider a linear system with output  $\{y_k\}$ , control input  $\{u_k\}$ , and an additional zero-mean random input  $\{w_k\}$ . We assume that  $\{w_k\}$  is a stationary (up to second order) stochastic process. That is,  $\{w_k\}$  is a sequence of random variables satisfying, for all  $i, k = 0, \pm 1, \pm 2, \dots$ ,

$$E\{w_k\} = 0, \quad E\{w_0 w_i\} = E\{w_k w_{k+i}\} < \infty.$$

(All references to stationary processes in this section are meant in the limited sense just described.) By linearity,  $y_k$  is the sum of one sequence  $\{y_k^1\}$  due to the presence of  $\{u_k\}$  and another sequence  $\{y_k^2\}$  due to the presence of  $\{w_k\}$ :

$$y_k = y_k^1 + y_k^2. \quad (\text{F.1})$$

We assume that  $y_k^1$  and  $y_k^2$  are generated by some filters  $B_1(s)/A_1(s)$  and  $B_2(s)/A_2(s)$ , respectively:

$$A_1(s)y_k^1 = B_1(s)u_k, \quad (\text{F.2a})$$

$$A_2(s)y_k^2 = B_2(s)w_k. \quad (\text{F.2b})$$

Operating on Eqs. (F.2a) and (F.2b) with  $A_2(s)$  and  $A_1(s)$ , respectively, adding, and using Eq. (F.1), we obtain

$$\bar{A}(s)y_k = \bar{B}(s)u_k + v_k, \quad (\text{F.3})$$

where  $\bar{A}(s) = A_1(s)A_2(s)$ ,  $\bar{B}(s) = A_2(s)B_1(s)$ , and  $\{v_k\}$ , given by

$$v_k = A_1(s)B_2(s)w_k, \quad (\text{F.4})$$

is a zero-mean, generally correlated, stationary stochastic process.

We are interested in the case where  $u_k$  is a control input applied after  $y_k$  has occurred and has been observed, so that in Eq. (F.2a) we have  $B_1(0) = 0$ . Then, we may assume that the polynomials  $\bar{A}(s)$  and  $\bar{B}(s)$  have the form

$$\bar{A}(s) = 1 + \bar{a}_1s + \cdots + \bar{a}_{m_0}s^{m_0}, \quad \bar{B}(s) = \bar{b}_1s + \cdots + \bar{b}_{m_0}s^{m_0}$$

for some scalars  $\bar{a}_i$  and  $\bar{b}_i$ , and some positive integer  $m_0$ .

To summarize, we have constructed a model of the form

$$\bar{A}(s)y_k = \bar{B}(s)u_k + v_k,$$

where  $\bar{A}(s)$  and  $\bar{B}(s)$  are polynomials of the preceding form and  $\{v_k\}$  is some zero-mean, correlated, stationary stochastic process. We now need to model further the sequence  $\{v_k\}$ .

## F.2 PROCESSES WITH RATIONAL SPECTRUM

Given a zero-mean, stationary scalar process  $\{v_k\}$ , denote by  $V(k)$  the autocorrelation function

$$V(k) = E\{v_i v_{i+k}\}, \quad k = 0, \pm 1, \pm 2, \dots$$

We say that  $\{v_k\}$  has *rational spectrum* if the transform of  $\{V(k)\}$  defined by

$$S_v(\lambda) = \sum_{k=-\infty}^{\infty} V(k)e^{-jk\lambda}$$

exists for  $\lambda \in [-\pi, \pi]$  and can be expressed as

$$S_v(\lambda) = \sigma^2 \frac{|C(e^{j\lambda})|^2}{|D(e^{j\lambda})|^2}, \quad \lambda \in [-\pi, \pi], \quad (\text{F.5})$$

where  $\sigma$  is a scalar,  $C(z)$  and  $D(z)$  are some polynomials with real coefficients

$$C(z) = 1 + c_1z + \cdots + c_m z^m, \quad (\text{F.6a})$$

$$D(z) = 1 + d_1z + \cdots + d_m z^m, \quad (\text{F.6b})$$

and  $D(z)$  has no roots on the unit circle  $\{z \mid |z| = 1\}$ .

The following facts are of interest:

- (a) If  $\{v_k\}$  is an uncorrelated process with  $V(0) = \sigma^2$ ,  $V(k) = 0$  for  $k \neq 0$ , then

$$S_v(\lambda) = \sigma^2, \quad \lambda \in [-\pi, \pi],$$

and clearly  $\{v_k\}$  has rational spectrum.

- (b) If  $\{v_k\}$  has rational spectrum  $S_v$  given by Eq. (F.5), then  $S_v$  can be written as

$$S_v(\lambda) = \tilde{\sigma}^2 \frac{|\tilde{C}(e^{j\lambda})|^2}{|\tilde{D}(e^{j\lambda})|^2}, \quad \lambda \in [-\pi, \pi],$$

where  $\tilde{\sigma}$  is a scalar and  $\tilde{C}(z)$ ,  $\tilde{D}(z)$  are unique real polynomials of the form

$$\tilde{C}(z) = 1 + \tilde{c}_1z + \cdots + \tilde{c}_m z^m,$$

$$\tilde{D}(z) = 1 + \tilde{d}_1z + \cdots + \tilde{d}_m z^m,$$

such that:

- (1)  $\tilde{C}(z)$  has all its roots outside or on the unit circle, and if  $C(z)$  has no roots on the unit circle, then the same is true for  $\tilde{C}(z)$ .

- (2)  $\tilde{D}(z)$  has all roots strictly outside the unit circle.

These facts are seen by noting that if  $\rho \neq 0$  is a root of  $D(z)$ , then  $|D(e^{j\lambda})|^2 = D(e^{j\lambda})D(e^{-j\lambda})$  contains a factor

$$(1 - \rho^{-1}e^{j\lambda})(1 - \rho^{-1}e^{-j\lambda}) = \rho^{-2}(\rho - e^{j\lambda})(\rho - e^{-j\lambda}).$$

A little reflection shows that the roots of  $\tilde{D}(z)$  should be  $\rho$  or  $\rho^{-1}$  depending on whether  $\rho$  is outside or inside the unit circle. Similarly, the roots of  $\tilde{C}(z)$  are obtained from the roots of  $C(z)$ . Thus the polynomials  $\tilde{C}(z)$  and  $\tilde{D}(z)$  as well as  $\tilde{\sigma}^2$  can be uniquely determined. We may thus assume without loss of generality that  $C(z)$  and  $D(z)$  in Eq. (F.5) have no roots inside the unit circle.

There is a fundamental result here that relates to the realization of processes with rational spectrum. The proof is hard; see for example, Ash and Gardner [AsG75, pp. 75-76].

**Proposition F.1:** If  $\{v_k\}$  is a zero-mean, stationary stochastic process with rational spectrum

$$S_v(\lambda) = \sigma^2 \frac{|C(e^{j\lambda})|^2}{|D(e^{j\lambda})|^2}, \quad \lambda \in [-\pi, \pi],$$

where the polynomials  $C(s)$  and  $D(s)$  are given by

$$C(s) = 1 + c_1 s + \cdots + c_m s^m, \quad D(s) = 1 + d_1 s + \cdots + d_m s^m,$$

and are assumed (without loss of generality) to have no roots inside the unit circle, then there exists a zero-mean, uncorrelated stationary process  $\{\epsilon_k\}$  with  $E\{\epsilon_k^2\} = \sigma^2$  such that for all  $k$

$$v_k = d_1 v_{k-1} + \cdots + d_m v_{k-m} = c_k + c_1 \epsilon_{k-1} + \cdots + c_m \epsilon_{k-m}.$$

### F.3 THE ARMAX MODEL

Let us now return to the problem of representation of a linear system with stochastic inputs. We had arrived at the model

$$\bar{A}(s)y_k = \bar{B}(s)u_k + v_k. \quad (\text{F.7})$$

If the zero-mean stationary process  $\{v_k\}$  has rational spectrum, the preceding analysis and proposition show that there exists a zero-mean, uncorrelated stationary process  $\{\epsilon_k\}$  satisfying

$$D(s)v_k = C(s)\epsilon_k,$$

where  $C(s)$  and  $D(s)$  are polynomials, and  $C(s)$  has no roots inside the unit circle. Operating on both sides of Eq. (F.7) with  $D(s)$  and using the relation  $D(s)v_k = C(s)\epsilon_k$ , we obtain

$$A(s)y_k = B(s)u_k + C(s)\epsilon_k, \quad (\text{F.8})$$

where  $A(s) = D(s)\bar{A}(s)$  and  $B(s) = D(s)\bar{B}(s)$ . Since  $\bar{A}(0) = 1$ ,  $\bar{B}(0) = 0$ , we can write Eq. (F.8) as

$$y_k + \sum_{i=1}^m a_i y_{k-i} = \sum_{i=1}^m b_i u_{k-i} + \epsilon_k + \sum_{i=1}^m c_i \epsilon_{k-i},$$

for some integer  $m$  and scalars  $a_i, b_i, c_i, i = 1, \dots, m$ . This is the ARMAX model that we have used in Section 5.3.

## APPENDIX G: Formulating Problems of Decision Under Uncertainty

In this appendix we discuss various approaches for formulating problems of decision under uncertainty. After a brief discussion of the min-max approach, we focus on the expected utility approach, and we show how this approach can be theoretically justified even if the decision maker is sensitive to the “variability” or “risk” associated with the results of different decisions.

### G.1 THE PROBLEM OF DECISION UNDER UNCERTAINTY

A decision problem in one of its simplest and most abstract forms consists of three nonempty sets  $\mathcal{D}$ ,  $\mathcal{N}$ , and  $\mathcal{O}$ , a function  $f : \mathcal{D} \times \mathcal{N} \mapsto \mathcal{O}$ , and a complete and transitive relation  $\preceq$  on  $\mathcal{O}$ . Here

$\mathcal{D}$  is the set of possible decisions,

$\mathcal{N}$  indexes the uncertainty in the problem and may be called the set of “states of nature,”

$\mathcal{O}$  is the set of outcomes of the decision problem,

$f$  is the function that determines which outcome will result from a given decision and state of nature, i.e., if decision  $d \in \mathcal{D}$  is selected and state of nature  $n \in \mathcal{N}$  prevails, then the outcome  $f(d, n) \in \mathcal{O}$  occurs,

$\preceq$  is a relation that determines our preference among the outcomes.<sup>†</sup> Thus, for  $O_1, O_2 \in \mathcal{O}$ , by  $O_1 \preceq O_2$  we mean that outcome  $O_2$  is at least as preferable as outcome  $O_1$ . By completeness of the relation, we mean that every two elements of  $\mathcal{O}$  are related, i.e., given any  $O_1, O_2 \in \mathcal{O}$ , there are three possibilities: either  $O_1 \preceq O_2$  but not  $O_2 \preceq O_1$ , or  $O_2 \preceq O_1$  but not  $O_1 \preceq O_2$ , or both  $O_1 \preceq O_2$  and  $O_2 \preceq O_1$ . By transitivity we mean that  $O_1 \preceq O_2$  and  $O_2 \preceq O_3$  implies  $O_1 \preceq O_3$  for any three elements  $O_1, O_2, O_3 \in \mathcal{O}$ .

### Example G.1

Consider an individual that may bet \$1 on the toss of a coin or not bet at all. If he bets and guesses correctly, he wins \$1 and if he does not guess correctly, he loses \$1. Here  $\mathcal{D}$  consists of three elements

$$\mathcal{D} = \{\text{bet on heads, bet on tails, not bet}\},$$

$\mathcal{N}$  consists of two elements

$$\mathcal{N} = \{\text{heads, tails}\},$$

and  $\mathcal{O}$  consists of three elements, the three possible final fortunes of the player

$$\mathcal{O} = \{\$0, \$1, \$2\}.$$

The preference relation on  $\mathcal{O}$  is the natural one, i.e.,  $0 \preceq 1$ ,  $0 \preceq 2$ ,  $1 \preceq 2$ , and the values of the function  $f$  are given by

$$f(H, H) = \$2, \quad f(T, H) = \$0, \quad f(\text{not bet}, H) = \$1,$$

$$f(H, T) = \$0, \quad f(T, T) = \$2, \quad f(\text{not bet}, T) = \$1.$$

Now the relative order by which we rank outcomes is usually clear in any given situation. On the other hand, for the decision problem to be completely formulated, we need a *ranking among decisions* that is consistent in a well-defined sense with our ranking of outcomes. Furthermore, to facilitate a mathematical or computational analysis, this ranking should be determined by a numerical function  $F$  that maps the set of decisions  $\mathcal{D}$  to the set of real numbers  $\mathbb{R}$  and is such that

$$d_1 \preceq d_2 \quad \text{if and only if} \quad F(d_1) \leq F(d_2), \quad \text{for all } d_1, d_2 \in \mathcal{D}, \quad (\text{G.1})$$

<sup>†</sup> The symbol  $\preceq$  in this appendix will be used (somewhat loosely) to denote a preference relation within either the set of outcomes or the set of decisions. The precise meaning should be clear from the context, and hopefully the use of the same symbol to denote different preference relations will create no confusion.

where the notation  $d_1 \preceq d_2$  implies that the decision  $d_2$  is at least as preferable as the decision  $d_1$ .

It is by no means clear how one should go about determining and characterizing a ranking among decisions. For example, in the gambling example above, different people will have different preferences as to accepting or refusing the gamble. In fact, the method by which one goes from a ranking of outcomes to a ranking of decisions is a central issue in decision theory. There are a number of approaches and viewpoints, and we now proceed to discuss some of these.

### Payoff Functions, Dominant, and Noninferior Decisions

Let us consider the case where it is possible to assign to each element of  $\mathcal{O}$  a real number in a way that the order between elements of  $\mathcal{O}$  agrees with the usual order of the corresponding numbers. In particular, we assume that there exists a real-valued function  $G : \mathcal{O} \rightarrow \mathbb{R}$  with the property

$$G(O_1) \leq G(O_2) \quad \text{if and only if} \quad O_1 \preceq O_2, \quad \text{for all } O_1, O_2 \in \mathcal{O}. \quad (\text{G.2})$$

Such a  $G$  does not always exist (see Exercise G.2). However, its existence can be guaranteed under quite general assumptions. In particular, one may show that it exists if  $\mathcal{O}$  is a countable set. Also if  $G$  exists, it is far from unique, since if  $\Phi$  is any monotonically increasing function  $\Phi : \mathbb{R} \mapsto \mathbb{R}$ , the composite function  $\Phi \cdot G$  [defined by  $(\Phi \cdot G)(O) = \Phi(G(O))$ ] has the same property (G.2) as  $G$ . For instance, in the example given earlier, a function  $G : \{0, 1, 2\} \mapsto \mathbb{R}$  satisfies Eq. (G.2) if and only if  $G(0) < G(1) < G(2)$  and there is an infinity of such functions.

For any choice of  $G$  satisfying Eq. (G.2), we define the function  $J : \mathcal{D} \times \mathcal{N} \mapsto \mathbb{R}$  by means of

$$J(d, n) = G(f(d, n))$$

and call it a *payoff function*.

Given a payoff function  $J$ , it is possible to obtain a complete ranking of decisions by means of a numerical function in the *special case of certainty* (the case where the set  $\mathcal{N}$  of states of nature consists of a single element  $\bar{n}$ ). By defining

$$F(d) = J(d, \bar{n}),$$

we have

$$d_1 \preceq d_2 \quad \text{if and only if} \quad F(d_1) \leq F(d_2) \quad \text{if and only if} \quad f(d_1, \bar{n}) \preceq f(d_2, \bar{n}),$$

and the numerical function  $F$  defines a complete ranking of decisions.

In the case where there is uncertainty, i.e., when  $\mathcal{N}$  contains more than one element, the order on  $\mathcal{O}$  induces only a *partial order* on  $\mathcal{D}$  by means of the relations

$$\begin{aligned} d_1 \preceq d_2 &\text{ if and only if } F(d_1) \leq F(d_2), \text{ for all } n \in \mathcal{N} \\ &\text{if and only if } f(d_1, n) \leq f(d_2, n), \text{ for all } n \in \mathcal{N}. \end{aligned} \quad (\text{G.3})$$

In this partial order, it is not necessary that every two elements of  $\mathcal{D}$  be related, i.e., for some  $d, d' \in \mathcal{D}$  we may have neither  $d \preceq d'$  nor  $d' \preceq d$ . If, however, for two decisions  $d_1, d_2 \in \mathcal{D}$  we have  $d_1 \preceq d_2$  in the sense of Eq. (G.3), then we can conclude that  $d_2$  is at least as preferable as  $d_1$  since the resulting outcome  $f(d_2, n)$  is at least as preferable as  $f(d_1, n)$ , *regardless of the state of nature n that will occur*.

A decision  $d^* \in \mathcal{D}$  is called a *dominant decision* if

$$d \preceq d^*, \quad \text{for all } d \in \mathcal{D},$$

where  $\preceq$  is understood in the sense of the partial order defined by Eq. (G.3). Naturally such a decision need not exist, but if it does exist, then it may be viewed as optimal. Unfortunately, in most problems of interest to an analyst there exists no dominant decision. For instance, this is so in the gambling example G.1, as the reader can easily verify. In fact no two decisions are related in the sense of Eq. (G.3) for this example.

In the absence of a dominant decision, one can consider the set  $\mathcal{D}_m \subset \mathcal{D}$  of all *noninferior decisions*, where  $d_m \in \mathcal{D}_m$  if for every  $d \in \mathcal{D}$  the relation  $d_m \preceq d$  implies  $d \preceq d_m$  in the sense of the partial order defined by Eq. (G.3). In terms of a payoff function  $J$ , noninferior decisions may be characterized by

$d_m \in \mathcal{D}_m$  if and only if there is no  $d \in \mathcal{D}$  such that

$$J(d_m, n) \leq J(d, n) \text{ for all } n \in \mathcal{N} \text{ and}$$

$$J(d_m, n) < J(d, n) \text{ for some } n \in \mathcal{N}.$$

Clearly it makes sense to consider only the decisions in  $\mathcal{D}_m$  as candidates for optimality since any decision that is not in  $\mathcal{D}_m$  is dominated by one that belongs to  $\mathcal{D}_m$ . Furthermore, it may be proved that the set  $\mathcal{D}_m$  is nonempty when the set  $\mathcal{D}$  is a finite set, so that at least for this case there exists at least one noninferior decision. However, in practice the set  $\mathcal{D}_m$  of noninferior decisions often is either difficult to determine or contains too many elements. For instance, in the gambling example given earlier, the reader may verify that every decision is noninferior.

Whenever the partial order of Eq. (G.3) fails to produce a satisfactory ranking among decisions, one must turn to other approaches to formulate the decision problem. The approaches that we will examine assume a notion of a *generalized outcome* of a decision and introduce a complete order on the set of these generalized outcomes that is consistent with the original order on the set of outcomes  $\mathcal{O}$ . The complete order on the set of generalized outcomes in turn induces a complete order on the set of decisions.

### The Min-Max Approach

In the min-max (or max-min) approach we take the point of view that the generalized outcome of a decision  $d$  is the set of all possible outcomes resulting from  $d$ :

$$f(d, \mathcal{N}) = \{O \in \mathcal{O} \mid \text{there exists } n \in \mathcal{N} \text{ with } f(d, n) = O\}.$$

In addition, we adopt a pessimistic attitude and rank the sets  $f(d, \mathcal{N})$  on the basis of their worst possible element. In particular, we introduce a complete order on the set of all subsets of  $\mathcal{O}$  by means of the relation

$$\mathcal{O}_1 \preceq \mathcal{O}_2 \text{ if and only if } \inf_{O \in \mathcal{O}_1} G(O) \leq \inf_{O \in \mathcal{O}_2} G(O), \text{ for all } \mathcal{O}_1, \mathcal{O}_2 \subset \mathcal{O}, \quad (\text{G.4})$$

where  $\mathcal{O}_1, \mathcal{O}_2$  is any pair of subsets of  $\mathcal{O}$ , and  $G$  is a numerical function consistent with the order on  $\mathcal{O}$  in accordance with Eq. (G.2). From Eq. (G.4) we have a complete order on the set of decisions  $\mathcal{D}$  by means of

$$d_1 \preceq d_2 \text{ if and only if } f(d_1, \mathcal{N}) \preceq f(d_2, \mathcal{N})$$

$$\text{if and only if } \inf_{n \in \mathcal{N}} G(f(d_1, n)) \leq \inf_{n \in \mathcal{N}} G(f(d_2, n)),$$

or in terms of a payoff function  $J$ ,

$$d_1 \preceq d_2 \text{ if and only if } \inf_{n \in \mathcal{N}} J(d_1, n) \leq \inf_{n \in \mathcal{N}} J(d_2, n).$$

Thus, by using the min-max approach, the decision problem is formulated concretely in that it reduces to maximizing over  $\mathcal{D}$  the numerical function

$$F(d) = \inf_{n \in \mathcal{N}} J(d, n).$$

Furthermore, it can be easily shown that the elements of  $\mathcal{D}$  that maximize  $F(d)$  above will not change if  $J$  is replaced by  $\Phi \cdot J$ , where  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$  is any monotonically increasing function. Nonetheless, the min-max approach is pessimistic in nature and will often produce an unduly conservative decision. Characteristically, in the gambling example G.1, the optimal decision according to the min-max approach is to refuse the gamble.

We next discuss another approach for formulating decision problems. This approach is quantifying the likelihood of various states of nature through probabilities.

## G.2 EXPECTED UTILITY THEORY AND RISK

In many decision problems under uncertainty we have additional information about the mechanism by which states of nature occur. In particular,

we are often in a position to know that these states occur in accordance with a given probabilistic mechanism, which may depend on the decision  $d$  adopted. To be specific, assume for convenience that the set of states of nature  $\mathcal{N}$  is either a finite set or a countable set† and that for every decision  $d \in \mathcal{D}$  we know that states of nature occur according to a given probability law  $P(\cdot | d)$  defined on  $\mathcal{N}$ . Now each decision  $d \in \mathcal{D}$  specifies the probability of each outcome via the function  $f(d, \cdot)$  and the relation

$$P_d(O) = P\left(\{n \mid f(d, n) = O\} \mid d\right), \quad \text{for all } O \in \mathcal{O}.$$

In this relation,  $P_d(O)$  denotes the probability that the outcome  $O$  will occur when the decision  $d$  is adopted. One may view the probability law  $P_d$  associated with each  $d \in \mathcal{D}$  as a "probabilistic outcome" (or "generalized outcome" to use the term of the preceding section) corresponding to  $d$ , since  $P_d$  specifies the probabilistic mechanism by which outcomes occur once  $d$  is selected. We shall also use the term *lottery*‡ for a probability law on the set of outcomes. In the gambling example G.1, the decision "bet on heads" has as a generalized outcome the probability law (or lottery)  $(1/2, 0, 1/2)$  on the set of outcomes  $\mathcal{O} = \{\$0, \$1, \$2\}$ . The decision "bet on tails" has the same generalized outcome, while the decision "not bet" has as a generalized outcome the probability law  $(0, 1, 0)$ .

The basic idea of the expected utility approach is the following. We already have a complete ranking of the outcomes, i.e., the elements of  $\mathcal{O}$ . If we had a complete ranking of all *lotteries* on the set of outcomes (presumably consistent with the original ranking on  $\mathcal{O}$  in the sense that if the outcome  $O_1$  is preferable to the outcome  $O_2$ , then the lottery assigning probability 1 to  $O_1$  is preferable to the lottery assigning probability 1 to outcome  $O_2$ ), then we could in turn obtain a complete ranking of all decisions in  $\mathcal{D}$ . This is true simply because we could rank any two decisions  $d_1, d_2 \in \mathcal{D}$  according to the relative order of their corresponding lotteries  $P_{d_1}, P_{d_2}$ , i.e., by means of the relation

$$d_1 \preceq d_2 \text{ if and only if } P_{d_1} \preceq P_{d_2}.$$

The fundamental premise of the expected utility approach is to assume at the outset that the *decision maker has a complete ranking of all lotteries on the set of outcomes*, i.e., the decision maker is in a position to express his preference between any two probability laws on the set of

† If  $\mathcal{N}$  is not countable, it is necessary to introduce a probability space structure on  $\mathcal{N}$  and  $\mathcal{O}$  as in Appendix C. Furthermore, it is necessary that the function  $f(d, \cdot)$  satisfy certain (measurability) assumptions.

‡ The term "lottery" is associated with conceptually convenient device of viewing outcomes as prizes of some sort and viewing a fixed probabilistic mechanism for winning a prize as a lottery.

outcomes. This in turn settles the question of ranking decisions in view of the preceding relation. Furthermore, if there exists a numerical function  $G$  by means of which preferences on the set of lotteries can be expressed,

$$P_{d_1} \preceq P_{d_2} \text{ if and only if } G(P_{d_1}) \leq G(P_{d_2}),$$

then decisions can be ranked by means of a numerical function  $F$ ,

$$d_1 \preceq d_2 \text{ if and only if } F(d_1) \leq F(d_2),$$

where  $F(d) = G(P_d)$  for all  $d \in \mathcal{D}$ .

The aspect of this formulation that is analytically very appealing, however, is that the ordering of decisions can be expressed not only by a function  $G$  as above, but also by means of an essentially unique numerical function called the *utility function*. This function, denoted  $U$ , maps the space of outcomes into the set of real numbers and satisfies

$$d_1 \preceq d_2 \text{ if and only if } P_{d_1} \preceq P_{d_2}$$

$$\text{if and only if } E\{U(f(d_1, n)) \mid d_1\} \leq E\{U(f(d_2, n)) \mid d_2\}, \quad (G.5)$$

where the expectations are taken with respect to the corresponding probability law  $P(\cdot | d)$  on  $\mathcal{N}$ . The problem of selecting an optimal decision is thus reduced to the problem of maximizing over  $\mathcal{D}$  the expected value of the numerical function  $U$ .

To clarify the problem formulation based on the approach of this section and to illustrate the advantages resulting from the introduction of a utility function, let us consider an example.

### Example G.2

Consider a problem of allocating one unit of capital between two investment opportunities A and B. Opportunity A yields \$1.5 per dollar invested with certainty, while opportunity B yields \$1 per dollar invested with probability 1/2 and \$3 per dollar invested with probability 1/2. The problem is to decide on the fractions  $d$  and  $(1 - d)$  of the capital to be invested in opportunities A and B, respectively, where  $0 \leq d \leq 1$ .

In terms of the framework of the decision problem of Section G.1, the set of decisions  $\mathcal{D}$  consists of the interval  $[0, 1]$ , i.e., the set of values that the fraction  $d$  invested in A can take. The set of states of nature  $\mathcal{N}$  consists of two elements  $n_1, n_2$ , where  $n_1$ : B yields \$1 per dollar invested, and  $n_2$ : B yields \$3 per dollar invested. The set of outcomes  $\mathcal{O}$  may be taken to be the interval  $[1, 3]$ , which is the set of possible final fortunes of the investor resulting from all possible decisions and states of nature. The function  $f$  that determines the outcome corresponding to any decision  $d$  and state of nature  $n$  is given by

$$f(d, n) = \begin{cases} 1.5d + (1 - d) & \text{if } n = n_1, \\ 1.5d + 3(1 - d) & \text{if } n = n_2. \end{cases}$$

The preference relation on the set of outcomes is the natural one, i.e., a final fortune  $O_1$  is at least as preferable as a final fortune  $O_2$  if  $O_1$  is numerically greater than or equal to  $O_2$  (i.e.,  $O_2 \preceq O_1$  if  $O_2 \leq O_1$ ).

Let us note that since  $B$  has a higher expected rate of return, the decision that maximizes expected value of profit is to invest exclusively in opportunity  $B$  ( $d^* = 0$ ). On the other hand the optimal decision based on the max-min approach is to invest exclusively in  $A$  ( $d^* = 1$ ) since in this approach one maximizes profit based on the assumption that the most unfavorable state of nature will occur. Mathematically this can be verified by noting that  $d^* = 1$  maximizes over  $[0, 1]$  the function  $F(d)$  given by

$$F(d) = \min\{1.5d + (1-d), 1.5d + 3(1-d)\}.$$

Note that the approach of maximizing expected profit and the max-min approach lead to very different decisions. Yet it is safe to assume that many decision makers would settle on a decision that differs from both decisions mentioned above and that invests a positive fraction of the capital in both opportunities  $A$  and  $B$ .

Now in the expected utility approach, the fundamental assumption is that the decision maker has a complete ranking of all lotteries on the set of outcomes. In other words, given any two probability distributions on the interval of final fortunes  $[0, 3]$ , the decision maker can express his preference between the two, in the sense that he can point out the probability distribution in accordance with which he would rather have his final fortune selected. Now the probability distribution on the set of final fortunes corresponding to a decision  $d$  is the one that assigns probability  $1/2$  to  $(1.5d + (1-d))$  and probability  $1/2$  to  $(1.5d + 3(1-d))$ . According to the expected utility approach, a decision  $d$  is optimal if its corresponding probability distribution is at least as preferable as all other probability distributions of the type described above. It should be clear, however, that a mathematical formulation of the corresponding optimization problem is very cumbersome since it is difficult to visualize or conjecture the form of a numerical function by means of which these probability distributions can be ranked. On the other hand, let us assume that a utility function  $U$  satisfying Eq. (G.5) exists (and it does exist under mild assumptions, as will be indicated shortly). Then an optimal decision is one that solves the problems

$$\begin{aligned} & \text{maximize } E\{U(f(d, n))\} \\ & \text{subject to } 0 \leq d \leq 1. \end{aligned}$$

Substituting the problem data, we have

$$E\{U(f(d, n))\} = \frac{1}{2}\left(U(1.5d + (1-d)) + U(1.5d + 3(1-d))\right)$$

so the maximization problem is conveniently formulated.

As an example, let us assume that the decision maker's utility function is quadratic of the form

$$U(O) = \alpha O - O^2,$$

where  $\alpha$  is some scalar. We require that  $6 < \alpha$  so that  $U(O)$  is increasing in the interval  $[0, 3]$ . This is necessary for the original preference relation on the set of outcomes to be consistent with the one specified by the utility function. Solution of the maximization problem above yields the optimal decision  $d^*$ , where

$$d^* = \begin{cases} 0 & \text{if } 6 \leq \alpha, \\ (8-\alpha)/5 & \text{if } 6 < \alpha < 8. \end{cases}$$

Note that for  $6 < \alpha < 8$ , a positive fraction of the capital is invested in opportunity  $A$  even though it offers a return that is less than the average return of  $B$ .

It should be noted, of course, that different decision makers faced with the same decision problem may have different utility functions, so that before the problem can be numerically solved, the form of the utility function must be specified. This can be done experimentally if necessary (see Exercise G.3). However, the importance of the notion of a utility function satisfying Eq. (G.5) lies primarily with the fact that under relatively mild assumptions, it exists and can serve as the starting point of analysis of the decision problem. The reason is that important conclusions about optimal decisions can often be obtained based on either incomplete knowledge of the utility function or fairly general assumptions on its form.

We provide below the theorem of existence of a utility function for the case where the set of outcomes  $\mathcal{O}$  is a finite set. For more general cases, see the book by Fishburn [Fis70].

Consider the set  $\mathcal{O}$  of outcomes and assume that it is a finite set,  $\mathcal{O} = \{O_1, O_2, \dots, O_N\}$ . Let  $\mathcal{P}$  be the set of all probability laws  $P = (p_1, p_2, \dots, p_N)$  on  $\mathcal{O}$ , where  $p_i$  is the probability of outcome  $O_i$ ,  $i = 1, \dots, N$ . For any  $P_1, P_2 \in \mathcal{P}$ ,  $P_1 = (p_1^1, \dots, p_N^1)$ ,  $P_2 = (p_1^2, \dots, p_N^2)$ , and any  $\alpha \in [0, 1]$ , we use the notation

$$\alpha P_1 + (1-\alpha)P_2 = (\alpha p_1^1 + (1-\alpha)p_1^2, \dots, \alpha p_N^1 + (1-\alpha)p_N^2).$$

Let us make the following assumptions:

A.1 There exists a complete and transitive relation  $\preceq$  on  $\mathcal{P}$ . (For any  $P_1, P_2 \in \mathcal{P}$ , we write  $P_1 \sim P_2$  if  $P_1 \preceq P_2$  and  $P_2 \preceq P_1$ , and we write  $P_1 \prec P_2$  if  $P_1 \preceq P_2$  but not  $P_2 \preceq P_1$ .)

A.2 If  $P_1 \sim P_2$ , then for all  $\alpha \in [0, 1]$  and all  $P \in \mathcal{P}$

$$\alpha P_1 + (1-\alpha)P \sim \alpha P_2 + (1-\alpha)P.$$

A.3 If  $P_1 \prec P_2$ , then for all  $\alpha \in (0, 1]$  and all  $P \in \mathcal{P}$

$$\alpha P_1 + (1-\alpha)P \prec \alpha P_2 + (1-\alpha)P.$$

A.4 If  $P_1 \prec P_2 \prec P_3$ , there exists an  $\alpha \in (0, 1)$  such that

$$\alpha P_1 + (1-\alpha)P_3 \sim P_2.$$

Before proving the expected utility theorem, let us briefly discuss the above assumptions. It is convenient for interpretation purposes to view each of the outcomes  $O_1, O_2, \dots, O_N$  as a monetary prize. Consider any probability law  $(p_1, p_2, \dots, p_N)$  on the set of outcomes. Imagine a pointer that spins in the center of a circle divided into  $N$  regions, and assume that it spins in a way that when it stops it is equally likely to be pointing in any direction. The region associated with each prize  $O_i$ ,  $i = 1, \dots, N$ , occupies a fraction  $p_i$  of the circumference of the circle. Then we associate with  $P$  the game (or lottery) whereby we spin the wheel and win the prize corresponding to the region within which the pointer stops. Now given any two probability laws  $P_1$  and  $P_2$  and a scalar  $\alpha \in [0, 1]$ , we can associate with the probability law

$$\alpha P_1 + (1 - \alpha) P_2$$

the following game. A pointer is spun in the center of a circle divided in two regions, say 1 and 2, occupying respective fractions  $\alpha$  and  $(1 - \alpha)$  of its circumference. Depending on whether the pointer stops in region 1 or region 2, the game corresponding to  $P_1$  and  $P_2$  is played and a prize is won accordingly.

Assumption A.1 requires that we are able to state our preference between games such as the above, which correspond to any two probability laws  $P_1$  and  $P_2$ . Furthermore, our preference relation must be transitive, i.e., if  $P_1 \preceq P_2$  and  $P_2 \preceq P_3$ , then  $P_1 \preceq P_3$ . This is the basic assumption, which forms the core of the expected utility approach. Assumptions A.2 and A.3 have obvious interpretations and both seem reasonable. Assumption A.4 is a continuity assumption requiring that if  $P_1 \prec P_2 \prec P_3$ , one is indifferent to the game associated with  $P_2$  and a game whose outcome decides with respective probabilities  $\alpha$  and  $(1 - \alpha)$  whether the game associated with  $P_1$  or  $P_3$  will be played. This assumption is inconsistent with a worst-case viewpoint whereby one ranks lotteries according to the worst outcome that can occur with positive probability, and has been the subject of some controversy. For example, consider the extreme situation where there are three outcomes  $O_1 = \text{death}$ ,  $O_2 = \text{receive nothing}$ , and  $O_3 = \text{receive \$1}$ . Then it appears reasonable that any probability law that assigns a positive probability to  $O_1$  (death) cannot be preferable or equivalent to any probability law that assigns a zero probability to  $O_1$ . Yet Assumption A.4 requires that for some  $\alpha$  with  $0 < \alpha < 1$  we are indifferent between the status quo and a game whereby we receive \\$1 with probability  $(1 - \alpha)$  and die with probability  $\alpha$ . On the other hand, it is possible to argue that if the probability of death  $\alpha$  is extremely close to zero, then this might actually be the case.

The following theorem is the central result of the expected utility theory. It states that a preference relation on the set of all lotteries, which satisfies Assumptions A.1-A.4 can be characterized numerically by means of an essentially unique function, the utility function. Note that *this result*

concerns an arbitrary preference relation on lotteries on the set of outcomes and is thus completely decoupled from any decision problem that one may be considering.

**Proposition G.1:** Under Assumptions A.1-A.4, there exists a real-valued function  $U : \mathcal{O} \mapsto \mathbb{R}$ , called *utility function*, such that for all  $P_1, P_2 \in \mathcal{P}$ ,

$$P_1 \preceq P_2 \text{ if and only if } E_{P_1}\{U(O)\} \leq E_{P_2}\{U(O)\},$$

where we denote by  $E_P\{\cdot\}$  the expected value with respect to a probability law  $P$ . Furthermore,  $U$  is unique up to a positive linear transformation, i.e., if  $U^*$  is another function with the above property, there exists a positive scalar  $s_1$  and a scalar  $s_2$ , such that

$$U^*(O) = s_1 U(O) + s_2, \quad \text{for all } O \in \mathcal{O}.$$

**Proof:** We first show the following statement:

S If  $P_1 \prec P_3$ , and  $P_2$  is such that  $P_1 \preceq P_2 \preceq P_3$ , then there exists a unique scalar  $\alpha \in [0, 1]$  such that

$$\alpha P_1 + (1 - \alpha) P_3 \sim P_2. \quad (\text{G.7})$$

Furthermore, if  $P'_2$  is such that  $P_1 \preceq P_2 \preceq P'_2 \preceq P_3$  and  $\alpha'$  corresponds to  $P'_2$  as in Eq. (G.7), then  $\alpha \geq \alpha'$ .

Indeed if  $P_1 \sim P_2 \prec P_3$ , then  $\alpha = 1$  is the unique scalar satisfying Eq. (G.7), since if for some  $\alpha \in [0, 1)$  we had

$$\alpha P_1 + (1 - \alpha) P_3 \sim P_2 \sim \alpha P_1 + (1 - \alpha) P_2,$$

then Assumption A.3 would be contradicted. Similarly, if  $P_1 \prec P_2 \sim P_3$ , then  $\alpha = 0$  is the unique scalar satisfying Eq. (G.7). Assume now that  $P_1 \prec P_2 \prec P_3$ . Then by Assumption A.4, there exists an  $\alpha_1 \in (0, 1)$  satisfying Eq. (G.7). Assume that  $\alpha_1$  is not unique and there exists another scalar  $\alpha_2 \in (0, 1)$  such that Eq. (G.7) is satisfied, i.e.,

$$\alpha_1 P_1 + (1 - \alpha_1) P_3 \sim P_2 \sim \alpha_2 P_1 + (1 - \alpha_2) P_3. \quad (\text{G.8})$$

Let us assume that  $0 < \alpha_1 < \alpha_2 < 1$ . Then we have

$$P_3 = \frac{\alpha_2 - \alpha_1}{1 - \alpha_1} P_3 + \frac{1 - \alpha_2}{1 - \alpha_1} P_1, \quad (\text{G.9})$$

$$\alpha_2 P_1 + (1 - \alpha_2) P_3 = \alpha_1 P_1 + (1 - \alpha) \left\{ \frac{\alpha_2 - \alpha_1}{1 - \alpha_1} P_1 + \frac{1 - \alpha_2}{1 - \alpha_1} P_3 \right\}. \quad (\text{G.10})$$

Since  $P_1 \prec P_3$ , we have by Assumption A.3 and Eq. (G.9)

$$\frac{\alpha_2 - \alpha_1}{1 - \alpha_1} P_1 + \frac{1 - \alpha_2}{1 - \alpha_1} P_3 \prec \frac{\alpha_2 - \alpha_1}{1 - \alpha_1} P_1 + \frac{1 - \alpha_2}{1 - \alpha_1} P_3 = P_3.$$

Again, using Assumption A.3 and Eq. (G.10), we have

$$\alpha_2 P_1 + (1 - \alpha_2) P_3 \prec \alpha_1 P_1 + (1 - \alpha_1) P_3.$$

However, this contradicts Eq. (G.8) and hence the uniqueness of the scalar  $\alpha$  in Eq. (G.7) is proved.

To show that  $P_1 \preceq P_2 \preceq P'_2 \preceq P_3$  implies  $\alpha \geq \alpha'$ , assume the contrary, i.e.,  $\alpha < \alpha'$ . Then we have, using Assumption A.3,

$$\begin{aligned} P'_2 &\sim \alpha' P_1 + (1 - \alpha') P_3 \\ &= (1 - \alpha + \alpha') \left\{ \frac{\alpha}{1 - \alpha + \alpha'} P_1 + \frac{1 - \alpha'}{1 - \alpha + \alpha'} P_3 \right\} + (\alpha' - \alpha) P_1 \\ &\prec (1 - \alpha + \alpha') \left\{ \frac{\alpha}{1 - \alpha + \alpha'} P_1 + \frac{1 - \alpha'}{1 - \alpha + \alpha'} P_3 \right\} + (\alpha' - \alpha) P_1 \\ &= \alpha P_1 + (1 - \alpha) P_3 \\ &\sim P_2. \end{aligned}$$

Hence  $P'_2 \prec P_2$ , which contradicts the assumption  $P_2 \preceq P'_2$ . It follows that  $\alpha \geq \alpha'$  and statement S is proved.

Now consider the probability laws

$$\bar{P}_1 = (1, 0, \dots, 0), \quad \bar{P}_2 = (0, 1, \dots, 0), \quad \dots, \quad \bar{P}_N = (0, 0, \dots, 1).$$

Assume without loss of generality that  $\bar{P}_1 \preceq \bar{P}_2 \preceq \dots \preceq \bar{P}_N$  and assume further that  $\bar{P}_1 \prec \bar{P}_N$  (if  $\bar{P}_1 \sim \bar{P}_2 \sim \dots \sim \bar{P}_N$ , the proof of the proposition is trivial). Let  $A_1, A_N$  be any scalars with  $A_1 < A_N$  and define

$$U(O_1) = A_1, \quad U(O_N) = A_N.$$

Let  $\alpha_i$ ,  $i = 1, \dots, n$ , be the unique scalar  $\alpha_i \in [0, 1]$  such that

$$\alpha_i \bar{P}_1 + (1 - \alpha_i) \bar{P}_N \sim \bar{P}_i, \quad i = 1, \dots, N, \quad (\text{G.11})$$

and define

$$U(O_i) = A_i = \alpha_i A_1 + (1 - \alpha_i) A_N, \quad i = 1, \dots, N. \quad (\text{G.12})$$

We shall prove that the function  $U : \mathcal{O} \mapsto \mathbb{R}$  defined above has the desired property (G.6). Indeed for any probability law  $P = (p_1, \dots, p_N)$ , it is easily

seen that  $\bar{P}_1 \prec P \prec \bar{P}_N$ , and thus we can define  $\alpha(P)$  to be the unique scalar in  $[0, 1]$  such that

$$\alpha(P) \bar{P}_1 + (1 - \alpha(P)) \bar{P}_N \sim P. \quad (\text{G.13})$$

From statement S we obtain for all  $P, P'$

$$P \preceq P' \text{ if and only if } \alpha(P) \geq \alpha(P'). \quad (\text{G.14})$$

Now from Eq. (G.11), we have

$$\begin{aligned} P &= \sum_{i=1}^N p_i \bar{P}_i \\ &\sim \sum_{i=1}^N p_i (\alpha_i \bar{P}_1 + (1 - \alpha_i) \bar{P}_N) \\ &\sim \sum_{i=1}^N p_i \alpha_i \bar{P}_1 + \left( 1 - \sum_{i=1}^N p_i \alpha_i \right) \bar{P}_N. \end{aligned} \quad (\text{G.15})$$

Comparing Eqs. (G.13) and (G.15), we obtain

$$\alpha(P) = \sum_{i=1}^N p_i \alpha_i,$$

and from Eq. (G.14),

$$P_1 \preceq P_2 \text{ if and only if } \sum_{i=1}^N p_i^1 \alpha_i \geq \sum_{i=1}^N p_i^2 \alpha_i. \quad (\text{G.16})$$

From Eq. (G.12), we have  $\alpha_i = (A_N - A_i)/(A_N - A_1)$ , and substituting in Eq. (G.16), we obtain

$$P_1 \preceq P_2 \text{ if and only if } \sum_{i=1}^N p_i^1 A_i \leq \sum_{i=1}^N p_i^2 A_i,$$

which is equivalent to the desired relation (G.6).

There remains to show that the function  $U$  defined by Eq. (G.12) is unique up to a positive linear transformation. Indeed if  $U^*$  were another utility function satisfying Eq. (G.6), then by denoting  $U^*(O_i) = A'_i$ ,  $i = 1, \dots, N$ , we would have from Eqs. (G.11) and (G.6)

$$U^*(O_i) = \alpha_i U^*(O_1) + (1 - \alpha_i) U^*(O_N).$$

It follows that

$$\alpha_i = \frac{A_N - A_i}{A_N - A_1} = \frac{A_N^* - A_i^*}{A_N^* - A_1^*},$$

from which

$$A_i^* = \frac{A_N^* - A_1^*}{A_N - A_1} A_i + A_N^* - \frac{A_N(A_N^* - A_1^*)}{A_N - A_1}.$$

This proves the theorem. Q.E.D.

Returning now to the decision problem, once we assume the existence of a preference relation on the set of lotteries that is characterized by a utility function, we can rank decisions as follows: Given the probability law  $P(\cdot | d)$  on the set of states of nature  $\mathcal{N}$ , every decision  $d \in \mathcal{D}$  induces a probability law (or lottery)  $P_d$  on the set of outcomes  $\mathcal{O}$ . Under the assumptions of the expected utility theorem, there exists a utility function  $U : \mathcal{O} \mapsto \mathbb{R}$  such that for any  $d_1, d_2 \in \mathcal{D}$

$$P_{d_1} \preceq P_{d_2} \text{ if and only if } E_{P_{d_1}}\{U(O)\} \leq E_{P_{d_2}}\{U(O)\}.$$

We have, however,

$$E_{P_d}\{U(O)\} = E\{U(f(d, n)) \mid d\}, \quad \text{for all } d \in \mathcal{D},$$

where the expectation on the left is taken with respect to  $P_d$  and the expectation on the right is taken with respect to the probability law  $P(\cdot | d)$  on  $\mathcal{N}$ . Hence

$$P_{d_1} \preceq P_{d_2} \text{ if and only if } E\{U(f(d_1, n)) \mid d_1\} \leq E\{U(f(d_2, n)) \mid d_2\}.$$

By ranking decisions  $d \in \mathcal{D}$  in accordance with the ranking of the corresponding  $P_d$ , i.e.,

$$d_1 \preceq d_2 \text{ if and only if } P_{d_1} \preceq P_{d_2}$$

$$\text{if and only if } E\{U(f(d_1, n)) \mid d_1\} \leq E\{U(f(d_2, n)) \mid d_2\},$$

we obtain a complete order on the set  $\mathcal{D}$  induced by the utility function  $U$ . The optimal decision is found by maximization of the numerical function  $F : \mathcal{D} \mapsto \mathbb{R}$ , where

$$F(d) = E\{U(f(d, n)) \mid d\}$$

and the decision problem is formulated in a way that is amenable to mathematical analysis.

### The Notion of Risk

Consider a decision maker possessing a utility function  $U$  defined over an interval  $X$  of real numbers. We say that the decision maker is *risk averse* if

$$E_P\{U(x)\} \leq U(E_P\{x\}) \quad (\text{G.17})$$

for every probability distribution  $P$  on  $X$  for which the expected value above is finite. In other words, a decision maker is risk averse if he always prefers the expected value of the lottery over the lottery itself. Such behavior characterizes most decision makers. One may show that risk aversion is equivalent to concavity of the utility function (see Appendix A for the definition and properties of concave and convex functions.) On the other hand, we say that the decision maker is *risk preferring* if the opposite inequality holds in Eq. (G.17), which is the case of a convex utility function. A gambler playing an unbiased roulette and receiving no reward or pleasure from gambling per se is a typical example of a risk preferring decision maker. Finally, a decision maker having a linear utility function is said to be *risk neutral*.

The notion of risk is important since it captures a basic attribute of the attitudes of the decision maker and often characterizes significant aspects of his behavior. An important and widely accepted measure of risk has been proposed by Pratt [Pra64]. He introduced the function

$$r(x) = -\frac{U''(x)}{U'(x)}, \quad (\text{G.18})$$

where  $U'$  and  $U''$  denote the first and second derivative of  $U$ , and it is assumed that  $U'(x) \neq 0$  for all  $x$ . This function, called the *index of absolute risk aversion*, measures locally (at the point  $x$ ) the risk aversion of the decision maker. It can be interpreted as follows.

Let  $x$  be a gamble over the set of real numbers (i.e., a random variable) with given distribution and expected value  $\bar{x} = E\{x\}$ . Let us denote by  $y$  the amount of insurance the decision maker is willing to pay in order to avoid the gamble  $x$ , and instead receive the expected value  $\bar{x}$  of the gamble. In other words,  $y$  is such that

$$U(\bar{x} - y) = E\{U(x)\}. \quad (\text{G.19})$$

Intuitively,  $y$  provides a natural measure of risk aversion. Using a Taylor series expansion around  $\bar{x}$ , we have

$$U(\bar{x} - y) = U(\bar{x}) - yU'(\bar{x}) + o(y), \quad (\text{G.20})$$

where by  $o(y)$  we denote a quantity that is negligible compared with the scalar  $\alpha$  provided  $\alpha$  is close to zero, i.e.,  $\lim_{\alpha \rightarrow 0} (o(\alpha)/\alpha) = 0$ . Also we have

$$\begin{aligned} E\{U(x)\} &= E\left\{U(\bar{x}) + (x - \bar{x})U'(\bar{x}) + \frac{1}{2}(x - \bar{x})^2U''(\bar{x}) + o((x - \bar{x})^2)\right\} \\ &= U(\bar{x}) + \frac{1}{2}\sigma^2U''(\bar{x}) + E\left\{o((x - \bar{x})^2)\right\}, \end{aligned} \quad (\text{G.21})$$

where  $\sigma^2$  is the variance of  $x$ . From Eqs. (G.19)-(G.21), we have

$$yU'(\bar{x}) = -\frac{1}{2}\sigma^2U''(\bar{x}) + o(y) + E\left\{o((x - \bar{x})^2)\right\}.$$

From this equation and Eq. (G.18) it follows that the amount of insurance or risk premium  $y$  that the decision maker is willing to pay is proportional (up to first order) to the index of absolute risk aversion  $r(\bar{x})$  at the mean  $\bar{x}$  of the gamble, thus justifying the use of  $r$  as a measure of local risk aversion. Notice that in the investment Example G.2, we have  $r(y) = 2/(\alpha - 2y)$ , so  $r(y)$  tends to decrease as  $\alpha$  increases. This fact is reflected in the optimal investment, where an increasing fraction of the capital is invested in the risky asset as  $\alpha$  is increased.

The index  $r(x)$  often plays an important role in the analysis of behavior of decision makers. It is generally accepted that for most decision makers,  $r(x)$  is a decreasing or at least nonincreasing function of  $x$ , i.e., the decision maker more readily accepts risk as his wealth is increased. On the other hand, for the quadratic utility function  $U(x) = -\frac{1}{2}x^2 + bx + c$ , the index  $r(x)$  is equal to  $(b - x)^{-1}$  and is an increasing function of  $x$  (for  $x < b$ ). For this reason the quadratic utility function is often considered inappropriate or at least accepted with reservation in economics applications, despite the analytical simplifications resulting from its use.

### Example G.3

An individual with given initial wealth  $\alpha$  wishes to invest part of it in a risky asset offering a rate of return  $e$ , and the rest in a secure asset offering rate of return  $s > 0$ . We assume that  $s$  is known with certainty while  $e$  is a random variable with known probability distribution  $P$ . If  $x$  is the amount invested in the risky asset, then the final wealth of the decision maker is given by

$$y = s(\alpha - x) + ex = s\alpha + (e - s)x.$$

The decision to be made by the individual is to choose  $x$  so as to maximize

$$J(x) = E\{U(y)\} = E\left\{U(s\alpha + (e - s)x)\right\}$$

subject to the constraint  $x \geq 0$ . We assume that  $U$  is a concave, monotonically increasing, twice continuously differentiable function with negative second derivative, and with index of absolute risk aversion

$$r(y) = -\frac{U''(y)}{U'(y)}.$$

We also assume that the probability distribution of  $e$  is such that all expected values appearing below are finite, and furthermore we assume that the utility function  $U$  is such that the maximization problem has a solution (the necessary and sufficient conditions for this have an interesting economic interpretation, which is discussed in Bertsekas [Ber74]).

Now given  $\alpha$ , the amount  $x^*$  to be invested in the risky asset is determined from the necessary conditions

$$\frac{dJ(x^*)}{dx} = E\left\{(e - s)U'(s\alpha + (e - s)x^*)\right\} = 0, \quad \text{if } x^* > 0, \quad (\text{G.22})$$

$$\frac{dJ(x^*)}{dx} \leq 0, \quad \text{if } x^* = 0.$$

Now since  $U'$  is everywhere positive it follows that if  $E\{(e - s)\} > 0$ , then we cannot have  $x^* = 0$  since

$$\frac{dJ(0)}{dx} = E\{(e - s)\}U'(s\alpha) > 0.$$

Hence  $E\{(e - s)\} > 0$  implies  $x^* > 0$ , or in words, a positive amount will be invested in the risky asset if its expected rate of return is greater than the rate of return of the secure asset.

Assume now that  $E\{(e - s)\} > 0$  and denote by  $x^*(\alpha)$  the amount invested in the risky asset when the initial wealth is  $\alpha$ . We would like to investigate the effects of changes in initial wealth  $\alpha$  on the amount  $x^*(\alpha)$  invested. By differentiating Eq. (G.22) with respect to  $\alpha$  we obtain

$$E\left\{(e - s)U''(s\alpha + (e - s)x^*(\alpha))(s\alpha + (e - s)(dx^*(\alpha)/d\alpha))\right\} = 0,$$

from which

$$\frac{dx^*(\alpha)}{d\alpha} = -\frac{E\left\{(e - s)U''(s\alpha + (e - s)x^*(\alpha))\right\}}{E\left\{(e - s)^2U''(s\alpha + (e - s)x^*(\alpha))\right\}}.$$

Since the denominator is always negative and the constant  $s$  is positive, the sign of  $dx^*(\alpha)/d\alpha$  is the same as the sign of

$$E\left\{(e - s)U''(s\alpha + (e - s)x^*(\alpha))\right\},$$

which using the definition of the index of absolute risk aversion  $r(y)$  is equal to

$$f(\alpha) = E\left\{(e - s)U'(s\alpha + (e - s)x^*(\alpha))r(s\alpha + (e - s)x^*(\alpha))\right\}.$$

Now assume that  $r(y)$  is monotonically decreasing, i.e.,

$$r(y_1) > r(y_2) \quad \text{if} \quad y_1 < y_2.$$

Then we have

$$(e - s)r(s\alpha + (e - s)x^*(\alpha)) \leq (e - r)r(s\alpha)$$

with strict inequality if  $e \neq s$ , and from the preceding relations, we obtain

$$\begin{aligned} f(\alpha) &> -r(s\alpha)E\{(e - s)U'(s\alpha + (e - s)x^*(\alpha))\} \\ &= -r(s\alpha)\frac{dJ(x^*(\alpha))}{dx} \\ &= 0. \end{aligned}$$

Thus we have  $f(\alpha) > 0$  and hence  $dx^*(\alpha)/d\alpha > 0$  if  $r(y)$  is monotonically decreasing. Similarly, we obtain  $dx^*(\alpha)/d\alpha < 0$  if  $r(y)$  is monotonically increasing. In words, the individual, given more wealth, will invest more (less) in the risky asset if his utility function has decreasing (increasing) index of absolute risk aversion. Aside from its intrinsic value, this result illustrates the important role of the index of risk aversion in shaping significant aspects of a decision maker's behavior.

### G.3 STOCHASTIC OPTIMAL CONTROL PROBLEMS

The class of decision problems considered so far in this appendix is very broad. In this book we are interested in a subclass of decision problems that involves a dynamic system. Such systems have an input-output description and furthermore in such systems, inputs are selected sequentially after observing past outputs. This allows the possibility of feedback. Let us first give an abstract description of these problems.

Let us consider a system characterized by three sets  $U$ ,  $W$ , and  $Y$ , and a function  $S : U \times W \rightarrow Y$ . We call  $U$  the *input set*,  $W$  the *uncertainty set*,  $Y$  the *output set*, and  $S$  the *system function*. Thus an input  $u \in U$  and an uncertain quantity  $w \in W$  produce an output  $y = S(u, w)$  through the system function  $S$  (see Fig. G.1). Implicit here is the assumption that the choice of the input  $u$  is somehow controlled by a decision maker or device to be designed, while  $w$  is chosen by nature according to some mechanism, probabilistic or not.

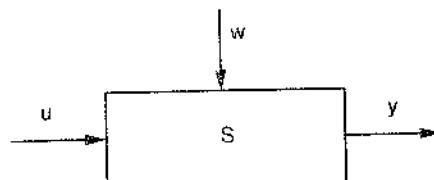


Figure G.1 Structure of an uncertain system:  $u$  is the input,  $w$  is the uncertain state of nature,  $y$  is the output, and  $S$  is the system function.

In many problems that evolve in time, the input is a time function or sequence, and there may be a possibility of observing the output  $y$  as it evolves in time. Naturally, this output may provide some information about the uncertain quantity  $w$ , which may be fruitfully taken into account in choosing the input  $u$  by means of a feedback mechanism.

Let us say that a function  $\pi : Y \mapsto U$  is a *feedback controller* (otherwise called *policy* or *decision function*) for the system if for each  $w \in W$  the equation

$$u = \pi(S(u, w))$$

has a unique solution (dependent on  $w$ ) for  $u$ . Thus for any fixed  $w$ , a feedback controller  $\pi$  generates a unique input  $u$  and hence a unique output  $y$  (see Fig. G.2). In any practical situation, the class of admissible feedback controllers is further restricted by causality (present inputs should not depend on future outputs), and possibly other constraints.

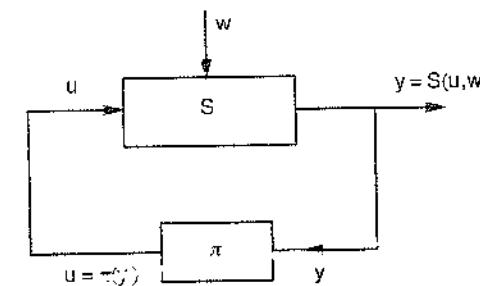


Figure G.2 Structure of a feedback controller  $\pi$ . We require that for each  $w \in W$  the equation

$$u = \pi(S(u, w))$$

has a unique solution in  $u$ .

Given the system  $(U, W, Y, S)$  and a set of admissible controllers  $\Pi$ , it is possible to formulate a decision problem in accordance with the theory of the previous section. We take  $\Pi$  as the decision set and  $W$  as the set of states of nature. We take as the set of outcomes the Cartesian product of  $U$ ,  $W$ , and  $Y$ , i.e.,

$$\mathcal{O} = (U \times W \times Y).$$

Now a feedback controller  $\pi \in \Pi$  and a state of nature  $w \in W$  generate a unique outcome  $(u, w, y)$ , where  $u$  is the unique solution of the equation  $u = \pi(S(u, w))$  and  $y = S(u, w)$ . Thus we may write  $(u, w, y) = f(\pi, w)$ , where  $f$  is some function determined by the system function  $S$ .

If  $G$  is a numerical function ordering our preferences on  $\mathcal{O}$ ,  $J$  is the corresponding payoff function for the decision problem above, and a max-min viewpoint is adopted, then the problem becomes one of finding  $\pi \in \Pi$  that maximizes

$$F(\pi) = \min_{w \in W} J(\pi, w) = \min_{w \in W} G(u, w, y),$$

where  $u$  and  $y$  are expressed in terms of  $\pi$  and  $w$  by means of  $u = \pi(S(u, w))$  and  $y = S(u, w)$  (min here denotes least upper bound over the corresponding set  $W$ ).

If  $w$  is selected in accordance with a known probabilistic mechanism, i.e., a given probability law that may depend on  $\pi$ , and the function  $S$  and the elements of  $\Pi$  satisfy suitable (measurability) assumptions, then it is possible to use a utility function  $U$  to formulate the decision problem as one of finding  $\pi \in \Pi$  that maximizes

$$F(\pi) = E\{U^{(u, \pi, w)}_{(x_0, \mu_0, \dots, \mu_{N-1})}\},$$

where  $u$  and  $y$  are expressed in terms of  $\pi$  and  $w$  by means of  $u = \pi(S(u, w))$  and  $y = S(u, w)$ .

While in the formulations just given, we have reduced the problem to one of decision under certainty [the problem of maximizing over  $\Pi$  the numerical function  $F(\pi)$ ], this is not an easy problem. The reason is that due to the feedback possibility, *the set  $\Pi$  is a set of functions* (of the system output). This renders inapplicable deterministic optimization techniques, such as those based on linear and nonlinear programming, or Pontryagin's minimum principle. Dynamic programming offers some possibility of analysis by decomposing the problem of minimizing  $F(\pi)$  into a sequence of much simpler optimization problems that are solved backwards in time, as discussed in Chapter 1.

Finally, let us indicate how to convert the basic problem of Section 1.2 into the general form given in this section. Referring to the discrete time dynamic system

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots, N-1, \quad (\text{G.23})$$

introduced in Section 1.2, the system input is the control sequence  $u = \{u_0, u_1, \dots, u_{N-1}\}$ , the uncertainty is  $w = \{w_0, w_1, \dots, w_{N-1}\}$  (perhaps together with the initial state  $x_0$ , if  $x_0$  is uncertain), the output is the state sequence  $y = \{x_0, x_1, \dots, x_N\}$ , and the system function is determined in the obvious manner from the system equation (G.23). The class  $\Pi$  of admissible feedback controllers is the set of sequences of functions  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , where  $\mu_k$  is a function that depends on the output  $y$  exclusively through the state  $x_k$ . Furthermore,  $\mu_k$  must satisfy constraints such as  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k$  and  $k$ .

## EXERCISES

### G.1

Show that there exists a function  $G : \mathcal{O} \rightarrow \mathbb{R}$  satisfying relation (G.2) provided the set  $\mathcal{O}$  is countable. Show also that if the set of decisions is finite, there exists at least one noninferior decision.

### G.2

Let  $\mathcal{O} = [-1, 1]$ . Define an order on  $\mathcal{O}$  by means of

$$O_1 \prec O_2 \text{ if and only if } |O_1| < |O_2| \text{ or } O_1 < O_2 = |O_1|.$$

Show that there exists no real-valued function  $G$  on  $\mathcal{O}$  such that

$$O_1 \prec O_2 \text{ if and only if } G(O_1) < G(O_2), \quad \text{for all } O_1, O_2 \in \mathcal{O}.$$

*Hint:* Assume the contrary and associate with every  $O \in (0, 1)$  a rational number  $r(O)$  such that

$$G(-O) < r(O) < G(O).$$

Show that if  $O_1 \neq O_2$ , then  $r(O_1) \neq r(O_2)$ .

### G.3 (Experimental Measurement of Utility)

Consider an individual faced with a decision problem with a finite collection of outcomes  $O_1, O_2, \dots, O_N$ . Assume that the individual has a preference relation over the set of lotteries on the set of outcomes satisfying Assumptions A.1-A.4 of the expected utility theorem, and hence a utility function over the set of outcomes exists. Suppose also that  $O_1 \preceq O_2 \preceq \dots \preceq O_N$  and furthermore that  $O_1 \prec O_N$ .

- (a) Show that the following method will determine a utility function. Define  $U(O_1) = 0, U(O_N) = 1$ . Let  $p_i$  with  $0 \leq p_i \leq 1$  be the probability for which one is indifferent between the lottery  $\{(1 - p_i), 0, \dots, 0, p_i\}$  and  $O_i$  occurring with certainty. Then let  $U(O_i) = p_i$ . Try the procedure on yourself for  $O_i = 100i$  with  $i = 0, 1, \dots, 10$ .
- (b) Show that the following procedure will also yield a utility function. Determine  $U(O_{N-1})$  as in (a), but set

$$U(O_{N-2}) = \tilde{p}_{N-2}U(O_{N-1}),$$

where  $\tilde{p}_{N-2}$  is the probability for which one is indifferent between the lottery  $\{(1 - \tilde{p}_{N-2}), 0, \dots, 0, \tilde{p}_{N-2}, 0\}$  and  $O_{N-2}$  occurring with certainty.

Similarly, set  $U(O_i) = \tilde{p}_i U(O_{i-1})$ , where  $\tilde{p}_i$  is the appropriate probability. Again try this procedure on yourself for  $O_i = 100i$  with  $i = 0, 1, \dots, 10$ , and compare the results with the ones obtained in part (a).

- (c) Devise a third procedure whereby the utilities  $U(O_1)$ ,  $U(O_2)$  are specified initially and  $U(O_i)$ ,  $i = 3, \dots, N$ , is obtained from  $U(O_{i-2})$ ,  $U(O_{i-1})$  through a comparison of the type considered above. Again try this procedure on yourself for  $O_i = 100i$  with  $i = 0, 1, \dots, 10$ .

#### G.4

Suppose that two persons, A and B, want to make a bet. Person A will pay \$1 to person B if a certain event occurs and person B will pay  $x$  dollars to person A if the event does not occur. Person A believes that the probability of the event occurring is  $p_A$  with  $0 < p_A < 1$ , while person B believes that this probability is  $p_B$  with  $0 < p_B < 1$ . Suppose that the utility functions  $U_A$  and  $U_B$  of persons A and B are strictly increasing functions of monetary gain. Let  $\alpha$ ,  $\beta$  be such that

$$U_A(\alpha) = \frac{U_A(0) - p_A U_A(-1)}{1 - p_A}, \quad U_B(-\beta) = \frac{U_B(0) - p_B U_B(1)}{1 - p_B}$$

Show that if  $\alpha < \beta$ ,  $\beta < 1$  any value of  $x$  between  $\alpha$  and  $\beta$  is a mutually satisfactory bet.

## References

- [ABC65] Atkinson, R. C., Bower, G. H., and Crothers, E. J., 1965. An Introduction to Mathematical Learning Theory, Wiley, N. Y.
- [ABF93] Arapostathis, A., Borkar, V., Fernandez-Gaucherand, E., Ghosh, M., and Marcus, S., 1993. "Discrete-Time Controlled Markov Processes with Average Cost Criterion: A Survey," SIAM J. on Control and Optimization, Vol. 31, pp. 282-344.
- [ABG49] Arrow, K. J., Blackwell, D., and Girshick, M. A., 1949. "Bayes and Minimax Solutions of Sequential Design Problems," Econometrica, Vol. 17, pp. 213-244.
- [AGK77] Athans, M., Ku, R., and Gershwin, S. B., 1977. "The Uncertainty Threshold Principle," IEEE Trans. on Automatic Control, Vol. AC-22, pp. 491-495.
- [AHM51] Arrow, K. J., Harris, T., and Marschak, J., 1951. "Optimal Inventory Policy," Econometrica, Vol. 19, pp. 250-272.
- [AKS58] Arrow, K. J., Karlin, S., and Scarf, H., 1958. Studies in the Mathematical Theory of Inventory and Production, Stanford Univ. Press, Stanford, CA.
- [Abr90] Abramson, B., 1990. "Expected-Outcome: A General Model of Static Evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, pp. 182-193.
- [AdG86] Adams, M., and Guillemin, V., 1986. Measure Theory and Probability, Wadsworth and Brooks, Monterey, CA.
- [AsG75] Ash, R. B., and Gardner, M. F., 1975. Topics in Stochastic Processes, Academic Press, N. Y.
- [AnM79] Anderson, B. D. O., and Moore, J. B., 1979. Optimal Filtering, Prentice-Hall, Englewood Cliffs, N. J.
- [AoL69] Aoki, M., and Li, M. T., 1969. "Optimal Discrete-Time Control Systems with Cost for Observation," IEEE Trans. Automatic Control, Vol. AC-14, pp. 165-175.
- [AsW73] Åström, K. J., and Wittenmark, B., 1973. "On Self-Tuning Regulators," Automatica, Vol. 9, pp. 185-199.
- [AsW84] Åström, K. J., and Wittenmark, B., 1984. Computer Controlled Systems, Prentice-Hall, Englewood Cliffs, N. J.
- [AsW94] Åström, K. J., and Wittenmark, B., 1994. Adaptive Control, (2nd Ed.), Prentice-Hall, Englewood Cliffs, N. J.
- [Ash70] Ash, R. B., 1970. Basic Probability Theory, Wiley, N. Y.
- [Ash72] Ash, R. B., 1972. Real Analysis and Probability, Academic Press, N. Y.

- [Ast83] Åström, K. J., 1983. "Theory and Applications of Adaptive Control – A Survey." *Automatica*, Vol. 19, pp. 471-486.
- [Ath66] Athans, M., and Falb, P., 1966. *Optimal Control*, McGraw-Hill, N. Y.
- [BGM95] Bertsekas, D. P., Guerriero, F., and Musmanno, R., 1995. "Parallel Shortest Path Methods for Globally Optimal Trajectories," *High Performance Computing: Technology, Methods, and Applications*, (J. Dongarra et al., Eds.), Elsevier.
- [BGM96] Bertsekas, D. P., Guerriero, F., and Musmanno, R., 1996. "Parallel Label Correcting Methods for Shortest Paths," *J. Optimization Theory Appl.*, Vol. 88, 1996, pp. 297-320.
- [BMS99] Boltyanski, V., Martini, H., and Soltan, V., 1999. *Geometric Methods and Optimization Problems*, Kluwer, Boston.
- [BNO03] Bertsekas, D. P., and A. Nedic, A., and A. E. Ozdaglar, 2003. *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA.
- [BTW97] Bertsekas, D. P., Tsitsiklis, J. N., and W. C., 1997. "Rollout Algorithms for Combinatorial Optimization," *Heuristics*, Vol. 3, pp. 245-262.
- [BaB95] Basar, T., and Bernhard, P., 1995. *H $\infty$  Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, Birkhäuser, Boston, MA.
- [Bar81] Bar-Shalom, Y., 1981. "Stochastic Dynamic Programming: Caution and Prob-ing," *IEEE Trans. on Automatic Control*, Vol. AC-26, pp. 1184-1195.
- [Bas91] Basar, T., 1991. "Optimum Performance Levels for Minimax Filters, Predictors, and Smoothers," *Systems and Control Letters*, Vol. 16, pp. 309-317.
- [Bas00] Basar, T., 2000. "Risk-Averse Designs: From Exponential Cost to Stochastic Games." In T. F. Djaferis and I. C. Schick, (Eds.), *System Theory: Modeling, Analysis and Control*, Kluwer, Boston, pp. 131-144.
- [BeC99] Bertsekas, D. P., and Castanon, D. A., 1999. "Rollout Algorithms for Stochastic Scheduling Problems," *Heuristics*, Vol. 5, pp. 89-108.
- [BeC04] Bertsekas, D. P., and Castanon, D. A., 2004. Unpublished Collaboration.
- [BeD62] Bellman, R., and Dreyfus, S., 1962. *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, N. J.
- [BeD02] Bertsimas, D., and Demir, R., 2002. "An Approximate Dynamic Programming Approach to Multi-Dimensional Knapsack Problems," *Management Science*, Vol. 4, pp. 550-565.
- [BeG92] Bertsekas, D. P., and Gallagher, R. G., 1992. *Data Networks* (2nd Edition), Prentice-Hall, Englewood Cliffs, N. J.
- [BeN98] Ben-Tal, A., and Nemirovski, A., 1998. "Robust Convex Optimization," *Math. of Operations Research*, Vol. 23, pp. 769-805.
- [BeN01] Ben-Tal, A., and Nemirovski, A., 2001. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, Phila., PA
- [BeP03] Bertsimas, D., and Popescu, I., 2003. "Revenue Management in a Dynamic Network Environment," *Transportation Science*, Vol. 37, pp. 257-277.
- [BeR71a] Bertsekas, D. P., and Rhodes, I. B., 1971. "Recursive State Estimation for a Set-Membership Description of the Uncertainty," *IEEE Trans. Automatic Control*, Vol. AC-16, pp. 117-128.

- [BeR71b] Bertsekas, D. P., and Rhodes, I. B., 1971. "On the Minimax Reachability of Target Sets and Target Tubes," *Automatica*, Vol. 7, pp. 233-247.
- [BeR73] Bertsekas, D. P., and Rhodes, I. B., 1973. "Sufficiently Informative Functions and the Minimax Feedback Control of Uncertain Dynamic Systems," *IEEE Trans. Automatic Control*, Vol. AC-18, pp. 117-124.
- [BeS78] Bertsekas, D. P., and Shreve, S. E., 1978. *Stochastic Optimal Control: The Discrete Time Case*, Academic Press, N. Y.; republished by Athena Scientific, Belmont, MA, 1996; can be downloaded from the author's website.
- [BcS03] Bertsimas, D., and Sim, M., 2003. "Robust Discrete Optimization and Network Flows," *Math. Programming*, Series B, Vol. 98, pp. 49-71.
- [BeT89] Bertsekas, D. P., and Tsitsiklis, J. N., 1989. *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J.; republished by Athena Scientific, Belmont, MA, 1997.
- [BeT91] Bertsekas, D. P., and Tsitsiklis, J. N., 1991. "An Analysis of Stochastic Shortest Path Problems," *Math. Operations Res.*, Vol. 16, pp. 580-595.
- [BeT96] Bertsekas, D. P., and Tsitsiklis, J. N., 1996. *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.
- [BcT97] Bertsimas, D., and Tsitsiklis, J. N., 1997. *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA.
- [Bel57] Bellman, R., 1957. *Dynamic Programming*, Princeton University Press, Princeton, N. J.
- [Ber70] Bertsekas, D. P., 1970. "On the Separation Theorem for Linear Systems, Quadratic Criteria, and Correlated Noise," Unpublished Report, Electronic Systems Lab., Massachusetts Institute of Technology.
- [Ber71] Bertsekas, D. P., 1971. "Control of Uncertain Systems With a Set-Membership Description of the Uncertainty," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA (available in scanned form from the author's www site).
- [Ber72a] Bertsekas, D. P., 1972. "Infinite Time Reachability of State Space Regions by Using Feedback Control," *IEEE Trans. Automatic Control*, Vol. AC-17, pp. 604-613.
- [Ber72b] Bertsekas, D. P., 1972. "On the Solution of Some Minimax Control Problems," Proc. 1972 IEEE Decision and Control Conf., New Orleans, LA.
- [Ber74] Bertsekas, D. P., 1974. "Necessary and Sufficient Conditions for Existence of an Optimal Portfolio," *J. Econ. Theory*, Vol. 8, pp. 235-247.
- [Ber75] Bertsekas, D. P., 1975. "Convergence of Discretization Procedures in Dynamic Programming," *IEEE Trans. Automatic Control*, Vol. AC-20, pp. 415-419.
- [Ber76] Bertsekas, D. P., 1976. *Dynamic Programming and Stochastic Control*, Academic Press, N. Y.
- [Ber82a] Bertsekas, D. P., 1982. "Distributed Dynamic Programming," *IEEE Trans. Automatic Control*, Vol. AC-27, pp. 610-616.
- [Ber82b] Bertsekas, D. P., 1982. *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, N. Y.; republished by Athena Scientific, Belmont, MA, 1996.
- [Ber93] Bertsekas, D. P., 1993. "A Simple and Fast Label Correcting Algorithm for Shortest Paths," *Networks*, Vol. 23, pp. 703-709.
- [Ber97] Bertsekas, D. P., 1997. "Differential Training of Rollout Policies," *Proc. of the*

- 35th Allerton Conference on Communication, Control, and Computing, Allerton Park, IL.
- [Ber98a] Bertsekas, D. P., 1998. Network Optimization: Continuous and Discrete Models. Athena Scientific, Belmont, MA.
- [Ber98b] Bertsekas, D. P., 1998. "A New Value Iteration Method for the Average Cost Dynamic Programming Problem," SIAM J. on Control and Optimization, Vol. 36, pp. 742-759.
- [Ber99] Bertsekas, D. P., 1999. Nonlinear Programming, (2nd Ed.), Athena Scientific, Belmont, MA.
- [Bil97] Birge, J. R., and Louveaux, 1997. Introduction to Stochastic Programming, Springer-Verlag, New York, N. Y.
- [Bin95] Bishop, C. M. 1995. Neural Networks for Pattern Recognition, Oxford University Press, N. Y.
- [BIT00] Blondel, V. D., and Tsitsiklis, J. N., 2000. "A Survey of Computational Complexity Results in Systems and Control," Automatica, Vol. 36, pp. 1249-1274.
- [Bla99] Blanchini, F., 1999. "Set Invariance in Control - A Survey," Automatica, Vol. 35, pp. 1747-1768.
- [BoV79] Borkar, V., and Varaiya, P. P., 1979. "Adaptive Control of Markov Chains. I: Finite Parameter Set," IEEE Trans. Automatic Control, Vol. AC-24, pp. 953-958.
- [CGC04] Chang, H. S., Givan, R. L., and Chong, E. K. P., 2004. "Parallel Rollout for Online Solution of Partially Observable Markov Decision Processes," Discrete Event Dynamic Systems, Vol. 14, pp. 309-341.
- [CaB04] Camacho, E. F., and Bordons, C., 2004. Model Predictive Control, 2nd Edition, Springer-Verlag, New York, N. Y.
- [ChT89] Chow, C.-S., and Tsitsiklis, J. N., 1989. "The Complexity of Dynamic Programming," Journal of Complexity, Vol. 5, pp. 466-488.
- [ChT91] Chow, C.-S., and Tsitsiklis, J. N., 1991. "An Optimal One-Way Multigrid Algorithm for Discrete-Time Stochastic Control," IEEE Trans. on Automatic Control, Vol. AC-36, 1991, pp. 898-914.
- [Che72] Chernoff, H., 1972. "Sequential Analysis and Optimal Design," Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA.
- [Chr97] Christodouleas, J. D., 1997. "Solution Methods for Multiprocessor Network Scheduling Problems with Application to Railroad Operations," Ph.D. Thesis, Operations Research Center, Massachusetts Institute of Technology.
- [Chu60] Chung, K. L., 1960. Markov Chains with Stationary Transition Probabilities, Springer-Verlag, N. Y.
- [CoL55] Coddington, E. A., and Levinson, N., 1955. Theory of Ordinary Differential Equations, McGraw-Hill, N. Y.
- [DeG70] DeGroot, M. H., 1970. Optimal Statistical Decisions, McGraw-Hill, N. Y.
- [DeP84] Deo, N., and Pang, C., 1984. "Shortest Path Problems: Taxonomy and Annotation," Networks, Vol. 14, pp. 275-323.
- [Del89] Deller, J. R., 1989. "Set Membership Identification in Digital Signal Processing," IEEE ASSP Magazine, Oct., pp. 4-20.
- [DoS80] Doshi, B., and Shreve, S., 1980. "Strong Consistency of a Modified Maximum Likelihood Estimator for Controlled Markov Chains," J. of Applied Probability, Vol. 17, pp. 726-734.

- [Dre65] Dreyfus, S. D., 1965. Dynamic Programming and the Calculus of Variations, Academic Press, N. Y.
- [Dre69] Dreyfus, S. D., 1969. "An Appraisal of Some Shortest-Path Algorithms," Operations Research, Vol. 17, pp. 395-412.
- [Eck68] Eckles, J. F., 1968. "Optimum Maintenance with Incomplete Information," Operations Res., Vol. 16, pp. 1058-1067.
- [Elm78] Elmaghraby, S. E., 1978. Activity Networks: Project Planning and Control by Network Models, Wiley-Interscience, N. Y.
- [Fal87] Falcone, M., 1987. "A Numerical Approach to the Infinite Horizon Problem of Deterministic Control Theory," Appl. Math. Opt., Vol. 15, pp. 1-13.
- [FeM94] Fernandez-Gaucherand, E., and Markus, S. I., 1994. "Risk Sensitive Optimal Control of Hidden Markov Models," Proc. 33rd IEEE Conf. Dec. Control, Lake Buena Vista, Fla.
- [FeV02] Ferris, M. C., and Voelker, M. M., 2002. "Neuro-Dynamic Programming for Radiation Treatment Planning," Numerical Analysis Group Research Report NA-02/06, Oxford University Computing Laboratory, Oxford University.
- [FeV04] Ferris, M. C., and Voelker, M. M., 2004. "Fractionation in Radiation Treatment Planning," Mathematical Programming B, Vol. 102, pp. 387-413.
- [Fel68] Feller, W., 1968. An Introduction to Probability Theory and its Applications, Wiley, N. Y.
- [Fis70] Fishburn, P. C., 1970. Utility Theory for Decision Making, Wiley, N. Y.
- [For56] Ford, L. R., Jr., 1956. "Network Flow Theory," Report P-923, The Rand Corporation, Santa Monica, CA.
- [For73] Forney, G. D., 1973. "The Viterbi Algorithm," Proc. IEEE, Vol. 61, pp. 268-278.
- [Fox71] Fox, B. L., 1971. "Finite State Approximations to Denumerable State Dynamic Programs," J. Math. Anal. Appl., Vol. 34, pp. 665-670.
- [GaP88] Gallo, G., and Pallottino, S., 1988. "Shortest Path Algorithms," Annals of Operations Research, Vol. 7, pp. 3-79.
- [Gal99] Gallager, R. G., 1999. Discrete Stochastic Processes, Kluwer, Boston.
- [GoR85] Gonzalez, R., and Rolman, E., 1985. "On Deterministic Control Problems: An Approximation Procedure for the Optimal Cost, Parts I, II," SIAM J. Control Optimization, Vol. 23, pp. 242-285.
- [GoS84] Goodwin, G. C., and Sin, K. S. S., 1984. Adaptive Filtering, Prediction, and Control, Prentice-Hall, Englewood Cliffs, N. J.
- [GrA66] Groen, G. J., and Atkinson, R. C., 1966. "Models for Optimizing the Learning Process," Psychol. Bull., Vol. 66, pp. 309-320.
- [GuF63] Gunckel, T. L., and Franklin, G. R., 1963. "A General Solution for Linear Sampled-Data Control," Trans. ASME Ser. D. J. Basic Engrg., Vol. 85, pp. 197-201.
- [GuM01] Guerricco, F., and Musmanno, R., 2001. "Label Correcting Methods to Solve Multicriteria Shortest Path Problems," J. Optimization Theory Appl., Vol. 111, pp. 589-613.

- [GuM03] Guerriero, F., and Mancini, M., 2003. "A Cooperative Parallel Rollout Algorithm for the Sequential Ordering Problem," *Parallel Computing*, Vol. 29, pp. 663-677.
- [HMS55] Holt, C. C., Modigliani, F., and Simon, H. A., 1955. "A Linear Decision Rule for Production and Employment Scheduling," *Management Sci.*, Vol. 2, pp. 1-30.
- [HPG96] Helmsen, J., Puckett, E. G., Colella, P., and Dorr, M., 1996. "Two New Methods for Simulating Photolithography Development," *SPIE*, Vol. 2726, pp. 253-261.
- [Hal82] Hajek, B., and van Loon, T., 1982. "Decentralized Dynamic Control of a Multiaccess Broadcast Channel," *IEEE Trans. Automatic Control*, Vol. AC-27, pp. 559-569.
- [Hak70] Hakansson, N. H., 1970. "Optimal Investment and Consumption Strategies under Risk for a Class of Utility Functions," *Econometrica*, Vol. 38, pp. 587-607.
- [Hak71] Hakansson, N. H., 1971. "On Myopic Portfolio Policies, With and Without Serial Correlation of Yields," *The Journal of Business of the University of Chicago*, Vol. 44, pp. 324-334.
- [Han80] Hansen, P., 1980. "Bicriterion Path Problems," in *Multiple-Criteria Decision Making: Theory and Applications*, Edited by G. Fandel and T. Gal, Springer Verlag, Heidelberg, Germany, pp. 109-127.
- [Hay98] Haykin, S., 1998. *Neural Networks: A Comprehensive Foundation*, (2nd Ed.), McMillan, N. Y.
- [Hes66] Hestenes, M. R., 1966. *Calculus of Variations and Optimal Control Theory*, Wiley, N. Y.
- [Her89] Hernández-Lerma, O., 1989. *Adaptive Markov Control Processes*, Springer-Verlag, N. Y.
- [HoK71] Hoffman, K., and Kunze, R., 1971. *Linear Algebra*, Prentice-Hall, Englewood Cliffs, N. J.
- [IEE71] IEEE Trans. Automatic Control, 1971. Special Issue on Linear-Quadratic Gaussian Problem, Vol. AC-16.
- [IoS96] Ioannou, P. A., and Sun, J., 1996. *Robust Adaptive Control*, Prentice-Hall, Englewood Cliffs, N. J.
- [JBE94] James, M. R., Baras, J. S., and Elliott, R. J., 1994. "Risk-Sensitive Control and Dynamic Games for Partially Observed Discrete-Time Nonlinear Systems," *IEEE Trans. on Automatic Control*, Vol. AC-39, pp. 780-792.
- [Jac73] Jacobson, D. H., 1973. "Optimal Stochastic Linear Systems With Exponential Performance Criteria and their Relation to Deterministic Differential Games," *IEEE Trans. Automatic Control*, Vol. AC-18, pp. 124-131.
- [Jaf84] Jaffe, J. M., 1984. "Algorithms for Finding Paths with Multiple Constraints," *Networks*, Vol. 14, pp. 95-116.
- [Jew63] Jewell, W., 1963. "Markov Renewal Programming I and II," *Operations Research*, Vol. 2, pp. 938-971.
- [JoT61] Joseph, P. D., and Tou, J. T., 1961. "On Linear Control Theory," *AIEE Trans.*, Vol. 80 (II), pp. 193-196.
- [KKK95] Krstic, M., Kanellakopoulos, I., Kokotovic, P., 1995. *Nonlinear and Adaptive Control Design*, J. Wiley, N. Y.
- [KGB82] Kimemia, J., Gershwin, S. B., and Bertsekas, D. P., 1982. "Computation of Production Control Policies by a Dynamic Programming Technique," in *Analysis and Optimization of Systems*, A. Bensoussan and J. L. Lions (eds.), Springer-Verlag, N. Y., pp. 243-269.

- [KLB92] Kosut, R. L., Lau, M. K., and Boyd, S. P., 1992. "Set-Membership Identification of Systems with Parametric and Nonparametric Uncertainty," *IEEE Trans. on Automatic Control*, Vol. AC-37, pp. 929-941.
- [KaD66] Karush, W., and Dear, E. E., 1966. "Optimal Stimulus Presentation Strategy for a Stimulus Sampling Model of Learning," *J. Math. Psychology*, Vol. 3, pp. 15-47.
- [KaK58] Kalman, R. E., and Koepcke, R. W., 1958. "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes," *Trans. ASME*, Vol. 80, pp. 1820-1826.
- [KaW94] Kall, P., and Wallace, S. W., 1994. *Stochastic Programming*, Wiley, Chichester, UK.
- [Kal60] Kalman, R. E., 1960. "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME Ser. D. J. Basic Engrg.*, Vol. 82, pp. 35-45.
- [KeS60] Kemeny, J. G., and Snell, J. L., 1960. *Finite Markov Chains*, Van Nostrand-Reinhold, N. Y.
- [KeG88] Keerthi, S. S., and Gilbert, E. G., 1988. "Optimal, Infinite Horizon Feedback Laws for a General Class of Constrained Discrete Time Systems: Stability and Moving-Horizon Approximations," *J. Optimization Theory Appl.*, Vol. 57, pp. 265-293.
- [Kim82] Kimemia, J., 1982. "Hierarchical Control of Production in Flexible Manufacturing Systems," Ph.D. Thesis, Dep. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
- [KuA77] Ku, R., and Athans, M., 1977. "Further Results on the Uncertainty Threshold Principle," *IEEE Trans. on Automatic Control*, Vol. AC-22, pp. 866-868.
- [KuD92] Kushner, H. J., and Dupuis, P. G., 1992. *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, N. Y.
- [KuL82] Kumar, P. R., and Lin, W., 1982. "Optimal Adaptive Controllers for Unknown Markov Chains," *IEEE Trans. Automatic Control*, Vol. AC-27, pp. 765-774.
- [KuV86] Kumar, P. R., and Varaiya, P. P., 1986. *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, Englewood Cliffs, N. J.
- [KuV97] Kurzhanski, A., and Valyi, I., 1997. *Ellipsoidal Calculus for Estimation and Control*, Birkhäuser, Boston, MA.
- [Kum83] Kumar, P. R., 1983. "Optimal Adaptive Control of Linear-Quadratic-Gaussian Systems," *SIAM J. on Control and Optimization*, Vol. 21, pp. 163-178.
- [Kum85] Kumar, P. R., 1985. "A Survey of Some Results in Stochastic Adaptive Control," *SIAM J. on Control and Optimization*, Vol. 23, pp. 329-380.
- [Kus90] Kushner, H. J., 1990. "Numerical Methods for Continuous Control Problems in Continuous Time," *SIAM J. on Control and Optimization*, Vol. 28, pp. 999-1048.
- [Las85] Lasserre, J. B., 1985. "A Mixed Forward-Backward Dynamic Programming Method Using Parallel Computation," *J. Optimization Theory Appl.*, Vol. 45, pp. 165-168.
- [Lev84] Levy, D., 1984. *The Chess Computer Handbook*, B. T. Batsford Ltd., London.
- [LiR71] Lippman, S. A., and Ross, S. M., 1971. "The Streetwalker's Dilemma: A Job-Shop Model," *SIAM J. of Appl. Math.*, Vol. 20, pp. 336-342.

- [LjS83] Ljung, L., and Soderstrom, T., 1983. *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA.
- [Lju86] Ljung, L., 1986. *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, N. J.
- [Lov91a] Lovejoy, W. S., 1991. "Computationally Feasible Bounds for Partially Observed Markov Decision Processes," *Operations Research*, Vol. 39, pp. 162-175.
- [Lov91b] Lovejoy, W. S., 1991. "A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes," *Annals of Operations Research*, Vol. 18, pp. 47-66.
- [Lue69] Luenberger, D. G., 1969. *Optimization by Vector Space Methods*, Wiley, N. Y.
- [Lue84] Luenberger, D. G., 1984. *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA.
- [MTH90] Miettinen, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L. K., and Dean, T., 1990. "Solving very Large Weakly Coupled Markov Decision Processes," Proc. of the Fifteenth National Conference on Artificial Intelligence, Madison, WI, pp. 165-172.
- [MMB02] McGovern, A., Moss, E., and Barto, A., 2002. "Building a Basic Building Block Scheduler Using Reinforcement Learning and Rollouts," *Machine Learning*, Vol. 49, pp. 141-160.
- [MPP04] Meloni, C., Pacciarelli, D., and Pranzo, M., 2004. "A Rollout Metaheuristic for Job Shop Scheduling Problems," *Annals of Operations Research*, Vol. 131, pp. 215-235.
- [MRR00] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., 2000. "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36, pp. 789-814.
- [Mac02] Maciejowski, J. M., 2002. *Predictive Control with Constraints*, Addison-Wesley, Reading, MA.
- [Mar84] Martins, E. Q. V., 1984. "On a Multicriteria Shortest Path Problem," *European J. of Operational Research*, Vol. 16, pp. 236-245.
- [May01] Mayne, D. Q., 2001. "Control of Constrained Dynamic Systems," *European Journal of Control*, Vol. 7, pp. 87-99.
- [McQ66] MacQueen, J., 1966. "A Modified Dynamic Programming Method for Markovian Decision Problems," *J. Math. Anal. Appl.*, Vol. 14, pp. 38-43.
- [Mik79] Mikhailov, V. A., 1979. *Methods of Random Multiple Access*, Candidate Engineering Thesis, Moscow Institute of Physics and Technology, Moscow.
- [MoL99] Morari, M., and Lee, J. H., 1999. "Model Predictive Control: Past, Present, and Future," *Computers and Chemical Engineering*, Vol. 23, pp. 667-682.
- [Mos68] Mossin, J., 1968. "Optimal Multi-Period Portfolio Policies," *J. Business*, Vol. 41, pp. 215-229.
- [NeW88] Nemhauser, G. L., and Wolsey, L. A., 1988. *Integer and Combinatorial Optimization*, Wiley, N. Y.
- [New75] Newborn, M., 1975. *Computer Chess*, Academic Press, N. Y.
- [Nic66] Nicholson, T., 1966. "Finding the Shortest Route Between Two Points in a Network," *The Computer Journal*, Vol. 9, pp. 275-280.
- [Nil71] Nilsson, N. J., 1971. *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, N. Y.

- [Nil80] Nilsson, N. J., 1971. *Principles of Artificial Intelligence*, Morgan-Kaufmann, San Mateo, Ca.
- [PBG65] Pontryagin, L. S., Boltyanski, V., Gamkrelidze, R., and Mishchenko, E., 1965. *The Mathematical Theory of Optimal Processes*, Interscience Publishers, Inc., N. Y.
- [PBT98] Polymenakos, L. C., Bertsekas, D. P., and Tsitsiklis, J. N., 1998. "Efficient Algorithms for Continuous-Space Shortest Path Problems," *IEEE Trans. on Automatic Control*, Vol. 43, pp. 278-283.
- [PaS82] Papadimitriou, C. H., and Steiglitz, K., 1982. *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, N. J.
- [PaT87] Papadimitriou, C. H., and Tsitsiklis, J. N., 1987. "The Complexity of Markov Decision Processes," *Math. Operations Res.*, Vol. 12, pp. 441-450.
- [Pap74] Pape, V., 1974. "Implementation and Efficiency of Moore Algorithms for the Shortest Path Problem," *Math. Progr.*, Vol. 7, pp. 212-222.
- [Pap65] Papoulis, A., 1965. *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, N. Y.
- [Pea84] Pearl, J., 1984. *Heuristics*, Addison-Wesley, Reading, MA.
- [Pic90] Picone, J., 1990. "Continuous Speech Recognition Using Hidden Markov Models," *IEEE ASSP Magazine*, July Issue, pp. 26-41.
- [Pin95] Pinedo, M., 1995. *Scheduling: Theory, Algorithms, and Systems*, Prentice-Hall, Englewood Cliffs, N. J.
- [Pra64] Pratt, J. W., 1964. "Risk Aversion in the Small and in the Large," *Econometrica*, Vol. 32, pp. 300-307.
- [Pre95] Prekopa, A., 1995. *Stochastic Programming*, Kluwer, Boston.
- [PrS94] Proakis, J. G., and Salehi, M., 1994. *Communication Systems Engineering*, Prentice-Hall, Englewood Cliffs, N. J.
- [Rab89] Rabiner, L. R., 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of the IEEE*, Vol. 77, pp. 257-286.
- [Roc70] Rockafellar, R. T., 1970. *Convex Analysis*, Princeton University Press, Princeton, N. J.
- [Ros70] Ross, S. M., 1970. *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, CA.
- [Ros83] Ross, S. M., 1983. *Introduction to Stochastic Dynamic Programming*, Academic Press, N. Y.
- [Ros85] Ross, S. M., 1985. *Probability Models*, Academic Press, Orlando, Fla.
- [Roy88] Royden, H. L., 1988. *Principles of Mathematical Analysis*, (3rd Ed.), McGraw-Hill, N. Y.
- [Rud76] Rudin, W., 1976. *Real Analysis*, (3rd Ed.), McGraw-Hill, N. Y.
- [Rus97] Rust, J., 1997. "Using Randomization to Break the Curse of Dimensionality," *Econometrica*, Vol. 65, pp. 487-516.
- [SBB89] Sastry, S., Bodson, M., and Bartram, J. F., 1989. *Adaptive Control: Stability, Convergence, and Robustness*, Prentice-Hall, Englewood Cliffs, N. J.
- [SGC02] Savagaonkar, U., Givan, R., and Chong, E. K. P., 2002. "Sampling Techniques for Zero-Sum, Discounted Markov Games," in Proc. 40th Allerton Conference on Com-

- munication. Control and Computing, Monticello, Ill.
- [Sam69] Samuelson, P. A., 1969. "Lifetime Portfolio Selection by Dynamic Stochastic Programming," *Review of Economics and Statistics*, Vol. 51, pp. 239-246.
- [Sar87] Sargent, T. J., 1987. *Dynamic Macroeconomic Theory*, Harvard Univ. Press, Cambridge, MA.
- [Sca60] Scarf, H., 1960. "The Optimality of  $(s, S)$  Policies for the Dynamic Inventory Problem," *Proceedings of the 1st Stanford Symposium on Mathematical Methods in the Social Sciences*, Stanford University Press, Stanford, CA.
- [Sch68] Schwerpke, F. C., 1968. "Recursive State Estimation; Unknown but Bounded Errors and System Inputs," *IEEE Trans. Automatic Control*, Vol. AC-13.
- [Sch74] Schwerpke, F. C., 1974. *Uncertain Dynamic Systems*, Academic Press, N. Y.
- [Sch97] Schaeffer, J., 1997. *One Jump Ahead*, Springer-Verlag, N. Y.
- [Sec00] Secomandi, N., 2000. "Comparing Neuro-Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands," *Computers and Operations Res.*, Vol. 27, pp. 1201-1225.
- [Sec01] Secomandi, N., 2001. "A Rollout Policy for the Vehicle Routing Problem with Stochastic Demands," *Operations Research*, Vol. 49, pp. 796-802.
- [Sec03] Secomandi, N., 2003. "Analysis of a Rollout Approach to Sequentializing Problems with Stochastic Routing Applications," *J. of Heuristics*, Vol. 9, pp. 321-352.
- [Sel99] Sethian, J. A., 1999. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, N. Y.
- [Set99b] Sethian, J. A., 1999. "Fast Marching Methods," *SIAM Review*, Vol. 41, pp. 199-235.
- [Sha50] Shannon, C., 1950. "Programming a Digital Computer for Playing Chess," *Phil. Mag.*, Vol. 41, pp. 356-375.
- [Shi64] Shiryaev, A. N., 1964. "On Markov Sufficient Statistics in Non-Additive Bayes Problems of Sequential Analysis," *Theory of Probability and Applications*, Vol. 9, pp. 604-618.
- [Shi66] Shiryaev, A. N., 1966. "On the Theory of Decision Functions and Control by an Observation Process with Incomplete Data," *Selected Translations in Math. Statistics and Probability*, Vol. 6, pp. 162-188.
- [Shr81] Shreve, S. E., 1981. "A Note on Optimal Switching Between Two Activities," *Naval Research Logistics Quarterly*, Vol. 28, pp. 185-190.
- [Sim56] Simon, H. A., 1956. "Dynamic Programming Under Uncertainty with a Quadratic Criterion Function," *Econometrica*, Vol. 24, pp. 74-81.
- [Skl88] Sklar, B., 1988. *Digital Communications: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, N. J.
- [SL91] Slotine, J.-J. E., and Li, W., 1991. *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, N. J.
- [SmS73] Smallwood, R. D., and Sondik, E. J., 1973. "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon," *Operations Res.*, Vol. 21, pp. 1071-1088.

- [Sma71] Smallwood, R. D., 1971. "The Analysis of Economic Teaching Strategies for a Simple Learning Model," *J. Math. Psychology*, Vol. 8, pp. 285-301.
- [Son71] Sondik, E. J., 1971. "The Optimal Control of Partially Observable Markov Processes," Ph.D. Dissertation, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.
- [StW91] Stewart, B. S., and White, C. C., 1991. "Multiobjective A\*," *J. ACM*, Vol. 38, pp. 775-814.
- [Sti94] Stirzaker, D., 1994. *Elementary Probability*, Cambridge University Press, Cambridge.
- [StL89] Stokey, N. L., and Lucas, R. E., 1989. *Recursive Methods in Economic Dynamics*, Harvard University Press, Cambridge, MA.
- [Str65] Striebel, C. T., 1965. "Sufficient Statistics in the Optimal Control of Stochastic Systems," *J. Math. Anal. Appl.*, Vol. 12, pp. 576-592.
- [Str76] Strang, G., 1976. *Linear Algebra and its Applications*, Academic Press, N. Y.
- [SuB98] Sutton, R., and Barto, A. G., 1998. *Reinforcement Learning*, MIT Press, Cambridge, MA.
- [TeG96] Tesauro, G., and Galperin, G. R., 1996. "On-Line Policy Improvement Using Monte Carlo Search," presented at the 1996 Neural Information Processing Systems Conference, Denver, CO; also in M. Mozer et al. (eds.), *Advances in Neural Information Processing Systems 9*, MIT Press (1997).
- [ThW66] Thau, F. E., and Witsenhausen, H. S., 1966. "A Comparison of Closed-Loop and Open-Loop Optimum Systems," *IEEE Trans. Automatic Control*, Vol. AC-11, pp. 619-621.
- [The54] Theil, H., 1954. "Econometric Models and Welfare Maximization," *Weltwirtschafts Arch.*, Vol. 72, pp. 60-83.
- [Tsi84a] Tsitsiklis, J. N., 1984. "Convexity and Characterization of Optimal Policies in a Dynamic Routing Problem," *J. Optimization Theory Appl.*, Vol. 44, pp. 105-136.
- [Tsi84b] Tsitsiklis, J. N., 1984. "Periodic Review Inventory Systems with Continuous Demand and Discrete Order Sizes," *Management Sci.*, Vol. 30, pp. 1250-1254.
- [Tsi87] Tsitsiklis, J. N., 1987. "Analysis of a Multiaccess Control Scheme," *IEEE Trans. Automatic Control*, Vol. AC-32, pp. 1017-1020.
- [Tsi95] Tsitsiklis, J. N., 1995. "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control*, Vol. AC-40, pp. 1528-1538.
- [TuP03] Tu, F., and Pattipati, K. R., 2003. "Rollout Strategies for Sequential Fault Diagnosis," *IEEE Trans. on Systems, Man and Cybernetics, Part A*, pp. 86-99.
- [YuB04] Yu, H., and Bertsekas, D. P., 2004. "Discretized Approximations for POMDP with Average Cost," Proc. of 20th Conference on Uncertainty in Artificial Intelligence, Banff, Canada.
- [VaW89] Varaiya, P., and Wets, R. J-B., 1989. "Stochastic Dynamic Optimization Approaches and Computation," *Mathematical Programming: State of the Art*, M. Iri and K. Tanabe (eds.), Kluwer, Boston, pp. 309-332.
- [VeI65] Veinott, A. F., Jr., 1965. "The Optimal Inventory Policy for Batch Ordering," *Operations Res.*, Vol. 13, pp. 424-432.

- [Vei66] Veinott, A. F., Jr., 1966. "The Status of Mathematical Inventory Theory." *Management Sci.*, Vol. 12, pp. 745-777.
- [Vin74] Vincke, P., 1974. "Problemes Multicriteres," *Cahiers du Centre d' Etudes de Recherche Operationnelle*, Vol. 16, pp. 425-439.
- [Vit67] Viterbi, A. J., 1967. "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Info. Theory*, Vol. IT-13, pp. 260-269.
- [WCG03] Wu, G., Chong, E. K. P., and Givan, R. L., 2003. "Congestion Control Using Policy Rollout," *Proc. 2nd IEEE CDC*, Maui, Hawaii, pp. 4825-4830.
- [Wal47] Wald, A., 1947. *Sequential Analysis*, Wiley, N. Y.
- [WeP80] Weiss, G., and Pinedo, M., 1980. "Scheduling Tasks with Exponential Service Times on Nonidentical Processors to Minimize Various Cost Functions," *J. Appl. Prob.*, Vol. 17, pp. 187-202.
- [WhH80] White, C. C., and Harrington, D. P., 1980. "Application of Jensen's Inequality to Adaptive Suboptimal Design," *J. Optimization Theory Appl.*, Vol. 32, pp. 89-99.
- [WhS89] White, C. C., and Scherer, W. T., 1989. "Solution Procedures for Partially Observed Markov Decision Processes," *Operations Res.*, Vol. 30, pp. 791-797.
- [Whi63] Whittle, P., 1963. *Prediction and Regulation by Linear Least-Square Methods*, English Universities Press, London.
- [Whi69] White, D. J., 1969. *Dynamic Programming*, Holden-Day, San Francisco, CA.
- [Whi78] Whitt, W., 1978. "Approximations of Dynamic Programs I," *Math. Operations Res.*, Vol. 3, pp. 231-243.
- [Whi79] Whitt, W., 1979. "Approximations of Dynamic Programs II," *Math. Operations Res.*, Vol. 4, pp. 179-185.
- [Whi82] Whittle, P., 1982. *Optimization Over Time*, Wiley, N. Y., Vol. 1, 1982, Vol. 2, 1983.
- [Whi90] Whittle, P., 1990. *Risk-Sensitive Optimal Control*, Wiley, N. Y.
- [Wit66] Witsenhausen, H. S., 1966. "Minimax Control of Uncertain Systems," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- [Wit68] Witsenhausen, H. S., 1968. "Sets of Possible States of Linear Systems Given Perturbed Observations," *IEEE Trans. Automatic Control*, Vol. AC-13.
- [Wit69] Witsenhausen, H. S., 1969. "Inequalities for the Performance of Suboptimal Uncertain Systems," *Automatica*, Vol. 5, pp. 507-512.
- [Wit70] Witsenhausen, H. S., 1970. "On Performance Bounds for Uncertain Systems," *SIAM J. on Control*, Vol. 8, pp. 55-89.
- [Wit71] Witsenhausen, H. S., 1971. "Separation of Estimation and Control for Discrete-Time Systems," *Proc. IEEE*, Vol. 59, pp. 1557-1566.
- [Wol98] Wolsey, L. A., 1998. *Integer Programming*, Wiley, N. Y.
- [WuB99] Wu, C. C., and Bertsekas, D. P., 1999. "Distributed Power Control Algorithms for Wireless Networks," unpublished report, available from the author's www site.
- [YDR05] Yan, X., Diaconis, P., Rusmevichientong, P., and Van Roy, B., 2005. "Solitaire: Man Versus Machine," *Advances in Neural Information Processing Systems*, Vol. 17, to appear.

# INDEX

## A

- A*\* algorithm, 87, 95
- ARMAX model, 238, 503, 506
- Adaptive control, 289
- Adjoint equation, 118, 130
- Admissible policy, 13, 219
- Aggregation, 319
- Aggregation probabilities, 320
- Alpha-beta pruning, 332
- Asset selling, 176, 278, 420
- Asynchronous algorithms, 102
- Augmentation of state, 35
- Autoregressive process, 239
- Average cost problem, 403, 421, 441

## B

- Backward shift operator, 236
- Basic problem, 12, 218
- Bayes' rule, 475
- Bellman, 51
- Bellman's equation, 404, 408, 418, 426, 440, 443
- Best-first search, 86
- Brachistochrone problem, 133
- Branch-and-bound algorithm, 88
- Breadth-first search, 84
- Breakthrough problem, 338, 368, 388, 396, 397

## C

- CEC, 283
- Calculus of variations, 108, 120
- Capacity expansion, 207
- Caution, 289
- Certainty equivalence principle, 28, 161
- Certainty equivalent control, 283, 293, 309
- Chess, 11, 15, 32, 327
- Closed-loop control, 4
- Closed set, 465
- Coarse grid, 324
- Communicating states, 478
- Compact set, 465
- Composition of functions, 466
- Concave function, 467
- Conditional probability, 475
- Constrained DP, 95, 397
- Constrained shortest path, 91

- Constrained controllability, 370, 374
- Constraint feasibility problem, 91
- Constraint qualification, 470
- Continuous function, 465
- Continuously differentiable, 466
- Controllability, 152, 370
- Control law, 13
- Control trajectory, 106
- Convergence of sequences, 465
- Convex function, 467
- Convex set, 467
- Convolutional coding, 74
- Correlated disturbances, 37, 181, 271
- Cost-to-go function, 24
- Countable set, 460
- Covariance matrix, 474
- Critical path analysis, 68
- Cumulative distribution function, 473

## D

- D'Esopo-Pape method, 86
- DP algorithm, 18, 222, 256
- DP algorithm proof, 23, 44, 48
- Data networks, 97, 102
- Decision function, 525
- Delays, 35
- Depth-first search, 85, 327
- Detectability, 159
- Differential cost, 429
- Dijkstra's algorithm, 86, 99, 101, 384, 389
- Disaggregation probabilities, 320
- Discounted cost, 52, 403, 417, 438
- Discretization, 324, 382
- Distributed computation, 102
- Distribution function, 473
- Disturbance, 13
- Dominant decision, 510
- Dual control, 289

## E

- Eigenvalue, 462
- Eigenvector, 462
- Euclidean space, 460
- Event, 472
- Existence of optimal solutions, 469
- Expected value, 474

Exponential cost function, 53, 202  
Exponential distribution, 437

**F**

Fast marching method, 389

Feature extraction, 326

Feature vectors, 326

Feedback controller, 525

First passage time, 411, 480

Flexible manufacturing, 316

Forecasts, 38, 175, 195, 202, 304

Fortified rollout algorithm, 355

Forward DP algorithm, 66

Forward shift operator, 237

Four queens problem, 77

Full rank, 461

**G**

Gambling, 208

Gauss-Markov estimator, 499

Gaussian random vector, 234, 273, 483, 484

Gradient, 466

Gradient matrix, 466

Greedy algorithm, 339, 349

**H**

Hamilton-Jacobi-Bellman equation, 109

Hamiltonian function, 118

Hard aggregation, 321

Hidden Markov model, 70

Hypothesis testing, 266

**I**

Identifiability, 291, 293

Improper policy, 406

Independent random variables, 474

Infimum, 460

Information vector, 219

Inner product, 460

Interchange argument, 186, 189

Inventory control, 3, 21, 162, 204

Investment problems, 60, 170

Irreducible Markov chain, 479

Isoperimetric problem, 144

Iterative deepening, 334

**K**

$K$ -convexity, 166

Kalman filter, 192, 234, 481, 491

Killer heuristic, 334

**L**

L'Hôpital's problem, 145

LLL strategy, 86, 97, 98  
Label correcting method, 81, 384, 389

Label setting method, 86, 384, 389

Limit inferior, 465

Limit point, 465

Limit superior, 465

Limited lookahead policy, 304

Linear independence, 461

Linear programming, 416

Linear quadratic problems, 27, 114, 122, 148, 202, 229, 241, 270

**M**

Markov chains, 477

Mean first passage time, 411, 480

Memoryless property, 437

Minimax algorithm, 331

Minimum principle, 119, 129

Minimum-time problem, 136

Minimum variance control, 236, 296

Minimax control problems, 46, 197, 332, 374, 388, 398

Model predictive control, 366, 369, 376, 388, 398

Monotonicity property, 59

Moving average process, 239

Multiaccess communication, 219, 287

Multiobjective DP, 93

Multiobjective shortest path, 92

Multiplicative cost, 54

Multistep lookahead, 304, 359

Myopic policy, 175

**N**

Noninferior decision, 92, 510

Noninferior solution, 92

Norm, 460

**O**

OLFC, 300

Observability, 152

One-step-lookahead rule, 184

Open-loop control, 4, 301, 376

Open-loop feedback control, 300, 376

Open set, 465

Optimal cost function, 14

Optimal value function, 14

Optimality conditions, 470

Optimality principle, 18, 93

Optimization in policy space, 386

**P**

POLFC, 303

Partially myopic policy, 175

Partial open-loop feedback control, 303, 376

Payoff function, 509

Pole-zero cancellation, 243

Policy, 13, 525

Policy iteration, 336, 414, 419, 432

Pontryagin minimum principle, 115, 119

Portfolio analysis, 170

Positive definite matrix, 463

Positive semidefinite matrix, 463

Principle of optimality, 18

Probability density function, 474

Probability distribution, 473

Probability space, 472

Probing, 289

Proper policy, 406

Pursuit-evasion game, 215

**Q**

Q-factor, 342, 361, 363

Quadratic cost, 27, 114, 122, 148, 229, 240, 369

Queueing control, 10, 34

**R**

Random variable, 473

Rank of a matrix, 461

Rational spectrum, 504

Reachability, 197, 201, 214, 215, 370, 374, 388

Recurrent state, 479

Relative cost, 429

Relative value iteration, 431

Replacement problems, 8, 34

Riccati equation, 114, 151

Riccati equation convergence, 153

Risk, 17, 53, 521

Rolling horizon, 367

Rollout algorithm, 307, 335, 372, 376

**S**

SLF method, 86, 97, 98

Scenarios, 313

Scheduling problems, 7, 19, 186

Self-tuning regulator, 298

Sequential consistency, 349

Sequential improvement, 353

Semi-Markov problems, 435

Semilinear systems, 55, 391

Separation theorem, 233

Sequential hypothesis testing, 266

Sequential probability ratio, 270

Set-membership estimation, 191

Set-membership models, 191, 373, 388

Shortest path problem, 65, 384, 389, 406  
Singular problem, 139

Slotted Aloha, 220

Soft aggregation, 321

Speech recognition, 73

Stabilizability, 159

Stable filter, 238

Stable system, 153

State trajectory, 106

Stationary policy, 405

Stochastic matrix, 477

Stochastic programming, 310

Stochastic shortest paths, 384, 403, 405

Stopping problems, 176

Sufficient statistic, 252

Supremum, 460

Symmetric matrix, 461

**T**

Terminating process, 53

Terminating rollout algorithm, 348

Tetris, 41

Time lags, 35

Total probability theorem, 475

Transient state, 479

Transition probabilities, 478

Transition rate, 437

Transpose, 461

Traveling salesman problem, 78, 347, 349

**U**

Uncertainty threshold principle, 160

Uncontrollable state components, 39

Uncorrelated random variables, 474

Unknown-but-bounded disturbances, 197

Utility function, 173, 513, 517, 526

Utility theory, 511

**V**

Value iteration, 413, 418, 430, 445

Value of information, 14

Vehicle routing, 315

Viterbi algorithm, 73, 288

**W**

Weierstrass theorem, 469

*Dynamic Programming  
and Optimal Control*  
*Volume II*

Dimitri P. Bertsekas

Massachusetts Institute of Technology



Athena Scientific, Belmont, Massachusetts

Athena Scientific  
Post Office Box 391  
Belmont, Mass. 02178-9998  
U.S.A.  
Email: athenasc@world.std.com

Cover Design: Ann Gallager



© 1995 Dimitri P. Bertsekas

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

Portions of this volume are adapted and reprinted from the author's *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, 1987, by permission of Prentice-Hall, Inc.

#### Publisher's Cataloging-in-Publication Data

Bertsekas, Dimitri P.  
Dynamic Programming and Optimal Control  
Includes Bibliography and Index  
1. Mathematical Optimization. 2. Dynamic Programming. I. Title.  
QA402.5 .B465 1995      519.703      95-075941

ISBN 1-886529-12-1 (Vol. I)

ISBN 1-886529-13-2 (Vol. II)

ISBN 1-886529-11-6 (Vol. I and II)

## Contents

### 1. Infinite Horizon - Discounted Problems

✓ 1.1. Minimization of Total Cost - Introduction . . . . .	p. 2
✓ 1.2. Discounted Problems with Bounded Cost per Stage . . . . .	p. 9
1.3. Finite-State Systems - Computational Methods . . . . .	p. 16
1.3.1. Value Iteration and Error Bounds . . . . .	p. 19
1.3.2. Policy Iteration . . . . .	p. 35
1.3.3. Adaptive Aggregation . . . . .	p. 41
1.3.4. Linear Programming . . . . .	p. 49
✓ 1.4. The Role of Contraction Mappings . . . . .	p. 52
1.5. Scheduling and Multiarmed Bandit Problems . . . . .	p. 51
1.6. Notes, Sources, and Exercises . . . . .	p. 61

### 2. Stochastic Shortest Path Problems

✓ 2.1. Main Results . . . . .	p. 78
2.2. Computational Methods . . . . .	p. 87
2.2.1. Value Iteration . . . . .	p. 88
2.2.2. Policy Iteration . . . . .	p. 91
2.3. Simulation-Based Methods . . . . .	p. 94
2.3.1. Policy Evaluation by Monte-Carlo Simulation . . . . .	p. 95
2.3.2. Q-Learning . . . . .	p. 99
2.3.3. Approximations . . . . .	p. 101
2.3.4. Extensions to Discounted Problems . . . . .	p. 118
2.3.5. The Role of Parallel Computation . . . . .	p. 120
2.4. Notes, Sources, and Exercises . . . . .	p. 121

### 3. Undiscounted Problems

3.1. Unbounded Costs per Stage . . . . .	p. 131
3.2. Linear Systems and Quadratic Cost . . . . .	p. 150
3.3. Inventory Control . . . . .	p. 153
3.4. Optimal Stopping . . . . .	p. 155
3.5. Optimal Gambling Strategies . . . . .	p. 160

3.6. Nonstationary and Periodic Problems . . . . .	p. 167
3.7. Notes, Sources, and Exercises . . . . .	p. 172

#### **4. Average Cost per Stage Problems**

4.1. Preliminary Analysis . . . . .	p. 184
4.2. Optimality Conditions . . . . .	p. 191
4.3. Computational Methods . . . . .	p. 202
4.3.1. Value Iteration . . . . .	p. 202
4.3.2. Policy Iteration . . . . .	p. 213
4.3.3. Linear Programming . . . . .	p. 221
4.3.4. Simulation-Based Methods . . . . .	p. 222
4.4. Infinite State Space . . . . .	p. 226
4.5. Notes, Sources, and Exercises . . . . .	p. 229

#### **5. Continuous-Time Problems**

5.1. Uniformization . . . . .	p. 242
5.2. Queueing Applications . . . . .	p. 250
5.3. Semi-Markov Problems . . . . .	p. 261
5.4. Notes, Sources, and Exercises . . . . .	p. 273

## CONTENTS OF VOLUME I

#### **1. The Dynamic Programming Algorithm**

1.1. Introduction	
1.2. The Basic Problem	
1.3. The Dynamic Programming Algorithm	
1.4. State Augmentation	
1.5. Some Mathematical Issues	
1.6. Notes, Sources, and Exercises	

#### **2. Deterministic Systems and the Shortest Path Problem**

2.1. Finite-State Systems and Shortest Paths	
2.2. Some Shortest Path Applications	
2.2.1. Critical Path Analysis	
2.2.2. Hidden Markov Models and the Viterbi Algorithm	
2.3. Shortest Path Algorithms	
2.3.1. Label Correcting Methods	
2.3.2. Auction Algorithms	
2.4. Notes, Sources, and Exercises	

#### **3. Deterministic Continuous-Time Optimal Control**

3.1. Continuous-Time Optimal Control	
3.2. The Hamilton–Jacobi–Bellman Equation	
3.3. The Pontryagin Minimum Principle	
3.3.1. An Informal Derivation Using the HJB Equation	
3.3.2. A Derivation Based on Variational Ideas	
3.3.3. The Minimum Principle for Discrete-Time Problems	
3.4. Extensions of the Minimum Principle	
3.4.1. Fixed Terminal State	
3.4.2. Free Initial State	
3.4.3. Free Terminal Time	
3.4.4. Time-Varying System and Cost	
3.4.5. Singular Problems	
3.5. Notes, Sources, and Exercises	

#### **4. Problems with Perfect State Information**

4.1. Linear Systems and Quadratic Cost	
4.2. Inventory Control	
4.3. Dynamic Portfolio Analysis	
4.4. Optimal Stopping Problems	
4.5. Scheduling and the Interchange Argument	
4.6. Notes, Sources, and Exercises	

## 5. Problems with Imperfect State Information

- 5.1. Reduction to the Perfect Information Case
- 5.2. Linear Systems and Quadratic Cost
- 5.3. Minimum Variance Control of Linear Systems
- 5.4. Sufficient Statistics and Finite-State Markov Chains
- 5.5. Sequential Hypothesis Testing
- 5.6. Notes, Sources, and Exercises

## 6. Suboptimal and Adaptive Control

- 6.1. Certainty Equivalent and Adaptive Control
  - 6.1.1. Caution, Probing, and Dual Control
  - 6.1.2. Two-Phase Control and Identifiability
  - 6.1.3. Certainty Equivalent Control and Identifiability
  - 6.1.4. Self-Tuning Regulators
- 6.2. Open-Loop Feedback Control
- 6.3. Limited Lookahead Policies and Applications
  - 6.3.1. Flexible Manufacturing
  - 6.3.2. Computer Chess
- 6.4. Approximations in Dynamic Programming
  - 6.4.1. Discretization of Optimal Control Problems
  - 6.4.2. Cost-to-Go Approximation
  - 6.4.3. Other Approximations
- 6.5. Notes, Sources, and Exercises

## 7. Introduction to Infinite Horizon Problems

- 7.1. An Overview
- 7.2. Stochastic Shortest Path Problems
- 7.3. Discounted Problems
- 7.4. Average Cost Problems
- 7.5. Notes, Sources, and Exercises

## Appendix A: Mathematical Review

## Appendix B: On Optimization Theory

## Appendix C: On Probability Theory

## Appendix D: On Finite-State Markov Chains

## Appendix E: Least-Squares Estimation and Kalman Filtering

## Appendix F: Modeling of Stochastic Linear Systems

## ABOUT THE AUTHOR

Dimitri Bertsekas studied Mechanical and Electrical Engineering at the National Technical University of Athens, Greece, and obtained his Ph.D. in system science from the Massachusetts Institute of Technology. He has held faculty positions with the Engineering-Economic Systems Dept., Stanford University and the Electrical Engineering Dept. of the University of Illinois, Urbana. He is currently Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. He consults regularly with private industry and has held editorial positions in several journals. He has been elected Fellow of the IEEE.

Professor Bertsekas has done research in a broad variety of subjects from control theory, optimization theory, parallel and distributed computation, data communication networks, and systems analysis. He has written numerous papers in each of these areas. This book is his fourth on dynamic programming and optimal control.

### Other books by the author:

- 1) *Dynamic Programming and Stochastic Control*, Academic Press, 1976.
- 2) *Stochastic Optimal Control: The Discrete-Time Case*, Academic Press, 1978 (with S. E. Shreve; translated in Russian).
- 3) *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982 (translated in Russian).
- 4) *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, 1987.
- 5) *Data Networks*, Prentice-Hall, 1987 (with R. G. Gallager; translated in Russian and Japanese); 2nd Edition 1992.
- 6) *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, 1989 (with J. N. Tsitsiklis).
- 7) *Linear Network Optimization: Algorithms and Codes*, M.I.T. Press 1991.

## *Preface*

This two-volume book is based on a first-year graduate course on dynamic programming and optimal control that I have taught for over twenty years at Stanford University, the University of Illinois, and the Massachusetts Institute of Technology. The course has been typically attended by students from engineering, operations research, economics, and applied mathematics. Accordingly, a principal objective of the book has been to provide a unified treatment of the subject, suitable for a broad audience. In particular, problems with a continuous character, such as stochastic control problems, popular in modern control theory, are simultaneously treated with problems with a discrete character, such as Markovian decision problems, popular in operations research. Furthermore, many applications and examples, drawn from a broad variety of fields, are discussed.

The book may be viewed as a greatly expanded and pedagogically improved version of my 1987 book "Dynamic Programming: Deterministic and Stochastic Models," published by Prentice-Hall. I have included much new material on deterministic and stochastic shortest path problems, as well as a new chapter on continuous-time optimal control problems and the Pontryagin Maximum Principle, developed from a dynamic programming viewpoint. I have also added a fairly extensive exposition of simulation-based approximation techniques for dynamic programming. These techniques, which are often referred to as "neuro-dynamic programming" or "reinforcement learning," represent a breakthrough in the practical application of dynamic programming to complex problems that involve the dual curse of large dimension and lack of an accurate mathematical model. Other material was also augmented, substantially modified, and updated.

With the new material, however, the book grew so much in size that it became necessary to divide it into two volumes: one on finite horizon, and the other on infinite horizon problems. This division was not only natural in terms of size, but also in terms of style and orientation. The first volume is more oriented towards modeling, and the second is more oriented towards mathematical analysis and computation. To make the first volume self-contained for instructors who wish to cover a modest amount of infinite horizon material in a course that is primarily oriented towards modeling,

conceptualization, and finite horizon problems, I have added a final chapter that provides an introductory treatment of infinite horizon problems.

Many topics in the book are relatively independent of the others. For example Chapter 2 of Vol. I on shortest path problems can be skipped without loss of continuity, and the same is true for Chapter 3 of Vol. I, which deals with continuous-time optimal control. As a result, the book can be used to teach several different types of courses.

- (a) A two-semester course that covers both volumes.
- (b) A one-semester course primarily focused on finite horizon problems that covers most of the first volume.
- (c) A one-semester course focused on stochastic optimal control that covers Chapters 1, 4, 5, and 6 of Vol. I, and Chapters 1, 2, and 4 of Vol. II.
- (d) A one-quarter engineering course that covers the first three chapters and parts of Chapters 4 through 6 of Vol. I.
- (e) A one-quarter mathematically oriented course focused on infinite horizon problems that covers Vol. II.

The mathematical prerequisite for the text is knowledge of advanced calculus, introductory probability theory, and matrix-vector algebra. A summary of this material is provided in the appendixes. Naturally, prior exposure to dynamic system theory, control, optimization, or operations research will be helpful to the reader, but based on my experience, the material given here is reasonably self-contained.

The book contains a large number of exercises, and the serious reader will benefit greatly by going through them. Solutions to all exercises are compiled in a manual that is available to instructors from Athena Scientific or from the author. Many thanks are due to the several people who spent long hours contributing to this manual, particularly Steven Shreve, Eric Loederman, Lakis Polymenakos, and Cynara Wu.

Dynamic programming is a conceptually simple technique that can be adequately explained using elementary analysis. Yet a mathematically rigorous treatment of general dynamic programming requires the complicated machinery of measure-theoretic probability. My choice has been to bypass the complicated mathematics by developing the subject in generality, while claiming rigor only when the underlying probability spaces are countable. A mathematically rigorous treatment of the subject is carried out in my monograph "Stochastic Optimal Control: The Discrete Time Case," Academic Press, 1978, coauthored by Steven Shreve. This monograph complements the present text and provides a solid foundation for the

subjects developed somewhat informally here.

Finally, I am thankful to a number of individuals and institutions for their contributions to the book. My understanding of the subject was sharpened while I worked with Steven Shreve on our 1978 monograph. My interaction and collaboration with John Tsitsiklis on stochastic shortest paths and approximate dynamic programming have been most valuable. Michael Caramanis, Emmanuel Fernandez-Gaucherand, Pierre Humbert, Leinart Ljung, and John Tsitsiklis taught from versions of the book, and contributed several substantive comments and homework problems. A number of colleagues offered valuable insights and information, particularly David Castanon, Eugene Feinberg, and Krishna Pattipati. NSF provided research support. Prentice-Hall graciously allowed the use of material from my 1987 book. Teaching and interacting with the students at MIT have kept up my interest and excitement for the subject.

Dimitri P. Bertsekas  
bertsekas@lids.mit.edu

*Infinite Horizon –  
Discounted Problems*

**Contents**

1.1. Minimization of Total Cost – Introduction . . . . .	p. 2
1.2. Discounted Problems with Bounded Cost per Stage . . . . .	p. 9
1.3. Finite-State Systems – Computational Methods . . . . .	p. 16
1.3.1. Value Iteration and Error Bounds . . . . .	p. 19
1.3.2. Policy Iteration . . . . .	p. 35
1.3.3. Adaptive Aggregation . . . . .	p. 44
1.3.4. Linear Programming . . . . .	p. 49
1.4. The Role of Contraction Mappings . . . . .	p. 52
1.5. Scheduling and Multiarmed Bandit Problems . . . . .	p. 54
1.6. Notes, Sources, and Exercises . . . . .	p. 64

This volume focuses on stochastic optimal control problems with an infinite number of decision stages (an infinite horizon). An introduction to these problems was presented in Chapter 7 of Vol. I. Here, we provide a more comprehensive analysis. In particular, we do not assume a finite number of states and we also discuss the associated analytical and computational issues in much greater depth.

We recall from Chapter 7 of Vol. I that there are four classes of infinite horizon problems of major interest.

- (a) Discounted problems with bounded cost per stage.
- (b) Stochastic shortest path problems.
- (c) Discounted and undiscounted problems with unbounded cost per stage.
- (d) Average cost per stage problems.

Each one of the first four chapters of the present volume considers one of the above problem classes, while the final chapter extends the analysis to continuous-time problems with a countable number of states. Throughout this volume we concentrate on the perfect information case, where each decision is made with exact knowledge of the current system state. Imperfect state information problems can be treated, as in Chapter 5 of Vol. I, by reformulation into perfect information problems involving a sufficient statistic.

## 1.1 MINIMIZATION OF TOTAL COST – INTRODUCTION

We now formulate the total cost minimization problem, which is the subject of this chapter and the next two. This is an infinite horizon, stationary version of the basic problem of Chapter 1 of Vol. I.

### Total Cost Infinite Horizon Problem

Consider the stationary discrete-time dynamic system

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, 1, \dots, \quad (1.1)$$

where for all  $k$ , the state  $x_k$  is an element of a space  $S$ , the control  $u_k$  is an element of a space  $C$ , and the random disturbance  $w_k$  is an element of a space  $D$ . We assume that  $D$  is a countable set. The control  $u_k$  is constrained to take values in a given nonempty subset  $U(x_k)$  of  $C$ , which depends on the current state  $x_k$  [ $u_k \in U(x_k)$ , for all  $x_k \in S$ ]. The random disturbances  $w_k$ ,  $k = 0, 1, \dots$ , have identical statistics and are characterized by probabilities  $P(\cdot | x_k, u_k)$  defined on  $D$ , where  $P(w_k | x_k, u_k)$  is the

probability of occurrence of  $w_k$ , when the current state and control are  $x_k$  and  $u_k$ , respectively. The probability of  $w_k$  may depend explicitly on  $x_k$  and  $u_k$  but not on values of prior disturbances  $w_{k-1}, \dots, w_0$ .

Given an initial state  $x_0$ , we want to find a policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , where  $\mu_k : S \mapsto C$ ,  $\mu_k(x_k) \in U(x_k)$ , for all  $x_k \in S$ ,  $k = 0, 1, \dots$ , that minimizes the cost function †

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} E_{w_k}^{\mu_k} \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}, \quad (1.2)$$

subject to the system equation constraint (1.1). The cost per stage  $g : S \times C \times D \mapsto \mathbb{R}$  is given, and  $\alpha$  is a positive scalar referred to as the *discount factor*.

We denote by  $H$  the set of all *admissible* policies  $\pi$ , that is, the set of all sequences of functions  $\pi = \{\mu_0, \mu_1, \dots\}$  with  $\mu_k : S \mapsto C$ ,  $\mu_k(x) \in U(x)$  for all  $x \in S$ ,  $k = 0, 1, \dots$ . The optimal cost function  $J^*$  is defined by

$$J^*(x) = \min_{\pi \in H} J_\pi(x), \quad x \in S.$$

A *stationary policy* is an admissible policy of the form  $\pi = \{\mu, \mu, \dots\}$ , and its corresponding cost function is denoted by  $J_\mu$ . For brevity, we refer to  $\{\mu, \mu, \dots\}$  as the stationary policy  $\mu$ . We say that  $\mu$  is optimal if  $J_\mu(x) = J^*(x)$  for all states  $x$ .

Note that, while we allow arbitrary state and control spaces, we require that the disturbance space be countable. This is necessary to avoid the mathematical complications discussed in Section 1.5 of Vol. I. The countability assumption, however, is satisfied in many problems of interest, notably for deterministic optimal control problems and problems with a finite or a countable number of states. For other problems, our main results can typically be proved (under additional technical conditions) by following the same line of argument as the one given here, but also by dealing with the mathematical complications of various measure-theoretic frameworks; see [BeS78].

The cost  $J_\pi(x_0)$  given by Eq. (1.2) represents the limit of expected finite horizon costs. These costs are well defined as discussed in Section

† In what follows we will generally impose appropriate assumptions on the cost per stage  $g$  and the discount factor  $\alpha$  that guarantee that the limit defining the total cost  $J_\pi(x_0)$  exists. If this limit is not known to exist, we use instead the definition

$$J_\pi(x_0) = \limsup_{N \rightarrow \infty} E_{w_k}^{\mu_k} \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\},$$

1.5 of Vol. I. Another possibility would be to minimize over  $\pi$  the expected infinite horizon cost

$$E_{\substack{w_k \\ k=0,1,\dots}} \left\{ \sum_{k=0}^{\infty} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}.$$

Such a cost would require a far more complex mathematical formulation (a probability measure on the space of all disturbance sequences; see [BeS78]). However, we mention that, under the assumptions that we will be using, the preceding expression is equal to the cost given by Eq. (1.2). This may be proved by using the monotone convergence theorem (see Section 3.1) and other stochastic convergence theorems, which allow interchange of limit and expectation under appropriate conditions.

### The DP Algorithm for the Finite-Horizon Version of the Problem

Consider any admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , any positive integer  $N$ , and any function  $J : S \mapsto \mathbb{R}$ . Suppose that we accumulate the costs of the first  $N$  stages, and to them we add the terminal cost  $\alpha^N J(x_N)$ , for a total expected cost

$$E_{\substack{w_k \\ k=0,1,\dots}} \left\{ \alpha^N J(x_N) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}.$$

The minimum of this cost over  $\pi$  can be calculated by starting with  $\alpha^N J(x)$  and by carrying out  $N$  iterations of the corresponding DP algorithm of Section 1.3 of Vol. I. This algorithm expresses the optimal  $(N-k)$ -stage cost starting from state  $x$ , denoted by  $J_k(x)$ , as the minimum of the expected sum of the cost of stage  $N-k$  and the optimal  $(N-k-1)$ -stage cost starting from the next state. It is given by

$$J_k(x) = \min_{u \in U(x)} E \{ \alpha^{N-k} g(x, u, w) + J_{k+1}(f(x, u, w)) \}, \quad k = 0, 1, \dots, N-1, \quad (1.3)$$

with the initial condition

$$J_N(x) = \alpha^N J(x).$$

For all initial states  $x$ , the optimal  $N$ -stage cost is the function  $J_0(x)$  obtained from the last step of the DP algorithm.

Let us consider for all  $k$  and  $x$ , the functions  $V_k$  given by

$$V_k(x) = \frac{J_{N-k}(x)}{\alpha^{N-k}}.$$

Then  $V_N(x)$  is the optimal  $N$ -stage cost  $J_0(x)$ , while the DP recursion (1.3) can be equivalently be written in terms of the functions  $V_k$  as

$$V_{k+1}(x) = \min_{u \in U(x)} E \{ g(x, u, w) + \alpha V_k(f(x, u, w)) \}, \quad k = 0, 1, \dots, N-1,$$

with the initial condition

$$V_0(x) = J(x).$$

The above algorithm can be used to calculate *all* the optimal finite horizon cost functions with a *single* DP recursion. In particular, suppose that we have computed the optimal  $(N-1)$ -stage cost function  $V_{N-1}$ . Then, to calculate the optimal  $N$ -stage cost function  $V_N$ , we do not need to execute the  $N$ -stage DP algorithm. Instead, we can calculate  $V_N$  using the one-stage iteration

$$V_N(x) = \min_{u \in U(x)} E \{ g(x, u, w) + \alpha V_{N-1}(f(x, u, w)) \}.$$

More generally, starting with some terminal cost function, we can consider applying repeatedly the DP iteration as above. With each application, we will be obtaining the optimal cost function of some finite horizon problem. The horizon of this problem will be longer by one stage over the horizon of the preceding problem. Note that this convenience is possible only because we are dealing with a stationary system and a common cost function  $g$  for all stages.

### Some Shorthand Notation

The preceding method of calculating finite horizon optimal costs motivates the introduction of two mappings that play an important theoretical role and provide a convenient shorthand notation in expressions that would be too complicated to write otherwise.

For any function  $J : S \mapsto \mathbb{R}$ , we consider the function obtained by applying the DP mapping to  $J$ , and we denote it by  $\dagger$

$$(TJ)(x) = \min_{u \in U(x)} E \{ g(x, u, w) + \alpha J(f(x, u, w)) \}, \quad x \in S. \quad (1.4)$$

Since  $(TJ)(\cdot)$  is itself a function defined on the state space  $S$ , we view  $T$  as a mapping that transforms the function  $J$  on  $S$  into the function  $TJ$  on  $S$ . Note that  $TJ$  is the *optimal cost function for the one-stage problem that has stage cost  $g$  and terminal cost  $\alpha J$* .

<sup>†</sup> Whenever we use the mapping  $T$ , we will impose sufficient assumptions to guarantee that the expected value involved in Eq. (1.4) is well defined.

Similarly, for any function  $J : S \mapsto \mathbb{R}$  and any control function  $\mu : S \mapsto C$ , we denote

$$(T_\mu J)(x) = \min_w \{g(x, \mu(x), w) + \alpha J(f(x, \mu(x), w))\}, \quad x \in S. \quad (1.5)$$

Again,  $T_\mu J$  may be viewed as the cost function associated with  $\mu$  for the one-stage problem that has stage cost  $g$  and terminal cost  $\alpha J$ .

We will denote by  $T^k$  the composition of the mapping  $T$  with itself  $k$  times; that is, for all  $k$  we write

$$(T^k J)(x) = (T(T^{k-1} J))(x), \quad x \in S.$$

Thus  $T^k J$  is the function obtained by applying the mapping  $T$  to the function  $T^{k-1} J$ . For convenience, we also write

$$(T^0 J)(x) = J(x), \quad x \in S.$$

Similarly,  $T_\mu^k J$  is defined by

$$(T_\mu^k J)(x) = (T_\mu(T_\mu^{k-1} J))(x), \quad x \in S,$$

and

$$(T_\mu^0 J)(x) = J(x), \quad x \in S.$$

It can be verified by induction that  $(T^k J)(x)$  is the optimal cost for the  $k$ -stage,  $\alpha$ -discounted problem with initial state  $x$ , cost per stage  $g$ , and terminal cost function  $\alpha^k J$ . Similarly,  $(T_\mu^k J)(x)$  is the cost of a policy  $\{\mu_0, \mu_1, \dots\}$  for the same problem. To illustrate the case where  $k = 2$ , note that

$$\begin{aligned} (T^2 J)(x) &= \min_{u \in U(x)} \min_w \{g(x, u, w) + \alpha(T J)(f(x, u, w))\} \\ &= \min_{u_0 \in U(x)} \min_{w_0} \left\{ g(x, u_0, w_0) + \alpha \min_{u_1 \in U(f(x, u_0, w_0))} \min_{w_1} \{g(f(x, u_0, w_0), u_1, w_1) \right. \\ &\quad \left. + \alpha J(f(f(x, u_0, w_0), u_1, w_1))\} \right\} \\ &= \min_{u_0 \in U(x)} \min_{w_0} \left\{ g(x, u_0, w_0) + \alpha \min_{u_1 \in U(f(x, u_0, w_0))} \min_{w_1} \{ \alpha g(f(x, u_0, w_0), u_1, w_1) \right. \\ &\quad \left. + \alpha^2 J(f(f(x, u_0, w_0), u_1, w_1)) \} \right\}. \end{aligned}$$

The last expression can be recognized as the DP algorithm for the 2-stage,  $\alpha$ -discounted problem with initial state  $x$ , cost per stage  $g$ , and terminal cost function  $\alpha^2 J$ .

Finally, consider a  $k$ -stage policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{k-1}\}$ . Then, the expression  $(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{k-1}} J)(x)$  is defined recursively for  $i = 0, \dots, k-2$  by

$$(T_{\mu_i} T_{\mu_{i+1}} \cdots T_{\mu_{k-1}} J)(x) = (T_{\mu_i}(T_{\mu_{i+1}} \cdots T_{\mu_{k-1}} J))(x)$$

and represents the cost of the policy  $\pi$  for the  $k$ -stage,  $\alpha$ -discounted problem with initial state  $x$ , cost per stage  $g$ , and terminal cost function  $\alpha^k J$ .

### Some Basic Properties

The following monotonicity property plays a fundamental role in the developments of this volume.

**Lemma 1.1: (Monotonicity Lemma)** For any functions  $J : S \mapsto \mathbb{R}$  and  $J' : S \mapsto \mathbb{R}$ , such that

$$J(x) \leq J'(x), \quad \text{for all } x \in S,$$

and for any function  $\mu : S \mapsto C$  with  $\mu(x) \in U(x)$ , for all  $x \in S$ , we have

$$(T^k J)(x) \leq (T^k J')(x), \quad \text{for all } x \in S, k = 1, 2, \dots,$$

$$(T_\mu^k J)(x) \leq (T_\mu^k J')(x), \quad \text{for all } x \in S, k = 1, 2, \dots$$

**Proof:** The result follows by viewing  $(T^k J)(x)$  and  $(T_\mu^k J)(x)$  as  $k$ -stage problem costs, since as the terminal cost function increases uniformly so will the  $k$ -stage costs. (One can also prove the lemma by using a straightforward induction argument.) **Q.E.D.**

For any two functions  $J : S \mapsto \mathbb{R}$  and  $J' : S \mapsto \mathbb{R}$ , we write

$$J \leq J' \quad \text{if } J(x) \leq J'(x) \text{ for all } x \in S.$$

With this notation, Lemma 1.1 is stated as

$$J \leq J' \quad \Rightarrow \quad T^k J \leq T^k J', \quad k = 1, 2, \dots,$$

$$J \leq J' \quad \Rightarrow \quad T_\mu^k J \leq T_\mu^k J', \quad k = 1, 2, \dots$$

Let us also denote by  $c : S \mapsto \mathbb{R}$  the unit function that takes the value 1 identically on  $S$ :

$$c(x) = 1, \quad \text{for all } x \in S. \quad (1.6)$$

We have from the definitions (1.4) and (1.5) of  $T$  and  $T_\mu$ , for any function  $J : S \mapsto \mathbb{R}$  and scalar  $r$

$$(T(J + rc))(x) = (TJ)(x) + \alpha r, \quad x \in S,$$

$$(T_\mu(J + rc))(x) = (T_\mu J)(x) + \alpha r, \quad x \in S.$$

More generally, the following lemma can be verified by induction using the preceding two relations.

**Lemma 1.2:** For every  $k$ , function  $J : S \mapsto \mathbb{R}$ , stationary policy  $\mu_k$  and scalar  $r$ , we have

$$(T^k(J + re))(x) = (T^k J)(x) + \alpha^k r, \quad \text{for all } x \in S, \quad (1.7)$$

$$(T_\mu^k(J + re))(x) = (T_\mu^k J)(x) + \alpha^k r, \quad \text{for all } x \in S. \quad (1.8)$$

### A Preview of Infinite Horizon Results

It is worth at this point to speculate on the type of results that we will be aiming for.

- (a) *Convergence of the DP Algorithm.* Let  $J_0$  denote the zero function [ $J_0(x) = 0$  for all  $x$ ]. Since the infinite horizon cost of a policy is, by definition, the limit of its  $k$ -stage costs as  $k \rightarrow \infty$ , it is reasonable to speculate that the optimal infinite horizon cost is equal to the limit of the optimal  $k$ -stage costs; that is,

$$J^*(x) = \lim_{k \rightarrow \infty} (T^k J_0)(x), \quad x \in S. \quad (1.9)$$

This means that if we start with the zero function  $J_0$  and iterate with the DP algorithm indefinitely, we will get in the limit the optimal cost function  $J^*$ . Also, for  $\alpha < 1$  and a bounded function  $J$ , a terminal cost  $\alpha^k J$  diminishes with  $k$ , so it is reasonable to speculate that, if  $\alpha < 1$ ,

$$J^*(x) = \lim_{k \rightarrow \infty} (T^k J)(x), \quad \text{for all } x \in S \text{ and bounded functions } J. \quad (1.10)$$

- (b) *Bellman's Equation.* Since by definition we have for all  $x \in S$

$$(T^{k+1} J_0)(x) = \min_{u \in U(x)} E \{g(x, u, w) + \alpha(T^k J_0)(f(x, u, w))\}, \quad (1.11)$$

it is reasonable to speculate that if  $\lim_{k \rightarrow \infty} T^k J_0 = J^*$  as in (a) above, then we must have by taking limit as  $k \rightarrow \infty$ ,

$$J^*(x) = \min_{u \in U(x)} E \{g(x, u, w) + \alpha J^*(f(x, u, w))\}, \quad x \in S, \quad (1.12)$$

or, equivalently,

$$J^* = TJ^*. \quad (1.13)$$

This is known as *Bellman's equation* and asserts that the optimal cost function  $J^*$  is a fixed point of the mapping  $T$ . We will see that

Bellman's equation holds for all the total cost minimization problems that we will consider, although depending on our assumptions, its proof can be quite complex.

- (c) *Characterization of Optimal Stationary Policies.* If we view Bellman's equation as the DP algorithm taken to its limit as  $k \rightarrow \infty$ , it is reasonable to speculate that if  $\mu(x)$  attains the minimum in the right-hand side of Bellman's equation for all  $x$ , then the stationary policy  $\mu$  is optimal.

Most of the analysis of total cost infinite horizon problems revolves around the above three issues and also around the issue of efficient computation of  $J^*$  and an optimal stationary policy. For the discounted cost problems with bounded cost per stage considered in this chapter, and for stochastic shortest path problems under our assumptions of Chapter 2, the preceding conjectures are correct. For problems with unbounded costs per stage and for stochastic shortest path problems where our assumptions of Chapter 2 are violated, there may be counterintuitive mathematical phenomena that invalidate some of the preceding conjectures. This illustrates that infinite horizon problems should be approached carefully and with mathematical precision.

## 1.2 DISCOUNTED PROBLEMS WITH BOUNDED COST PER STAGE

We now discuss the simplest type of infinite horizon problem. We assume the following:

**Assumption D (Discounted Cost – Bounded Cost per Stage):**  
The cost per stage  $g$  satisfies

$$|g(x, u, w)| \leq M, \quad \text{for all } (x, u, w) \in S \times C \times D, \quad (2.1)$$

where  $M$  is some scalar. Furthermore,  $0 < \alpha < 1$ .

Boundedness of the cost per stage is not as restrictive as might appear. It holds for problems where the spaces  $S$ ,  $C$ , and  $D$  are finite sets. Even if these spaces are not finite, during the computational solution of the problem they will ordinarily be approximated by finite sets. Also, it is often possible to reformulate the problem so that it is defined over bounded regions of the state and control spaces over which the cost is bounded.

The following proposition shows that the DP algorithm converges to the optimal cost function  $J^*$  for an arbitrary bounded starting function  $J$ . This will follow as a consequence of Assumption D, which implies that the "tail" of the cost after stage  $N$ , that is,

$$\lim_{K \rightarrow \infty} E \left\{ \sum_{k=N}^K \alpha^k g(x_k, \mu_k(x_k), w_k) \right\},$$

diminishes to zero as  $N \rightarrow \infty$ . Furthermore, when a terminal cost  $\alpha^N J(x_N)$  is added to the  $N$ -stage cost, its effect diminishes to zero as  $N \rightarrow \infty$  if  $J$  is bounded.

**Proposition 2.1: (Convergence of the DP Algorithm)** For any bounded function  $J : S \mapsto \mathbb{R}$ , the optimal cost function satisfies  $\forall x \in S$ ,

$$J^*(x) = \lim_{N \rightarrow \infty} (T^N J)(x), \quad \text{for all } x \in S. \quad (2.2)$$

**Proof:** For every positive integer  $K$ , initial state  $x_0 \in S$ , and policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , we break down the cost  $J_\pi(x_0)$  into the portions incurred over the first  $K$  stages and over the remaining stages

$$\begin{aligned} J_\pi(x_0) &= \lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &= E \left\{ \sum_{k=0}^{K-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\quad + \lim_{N \rightarrow \infty} E \left\{ \sum_{k=K}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}. \end{aligned}$$

Since by Assumption D we have  $|g(x_k, \mu_k(x_k), w_k)| \leq M$ , we also obtain

$$\left| \lim_{N \rightarrow \infty} E \left\{ \sum_{k=K}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \right| \leq M \sum_{k=K}^{\infty} \alpha^k = \frac{\alpha^K M}{1-\alpha}.$$

Using these relations, it follows that

$$\begin{aligned} J_\pi(x_0) &- \frac{\alpha^K M}{1-\alpha} - \alpha^K \max_{x \in S} |J(x)| \\ &\leq E \left\{ \alpha^K J(x_K) + \sum_{k=0}^{K-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\leq J_\pi(x_0) + \frac{\alpha^K M}{1-\alpha} + \alpha^K \max_{x \in S} |J(x)|. \end{aligned}$$

By taking the minimum over  $\pi$ , we obtain for all  $x_0$  and  $K$ ,

$$\begin{aligned} J^*(x_0) &- \frac{\alpha^K M}{1-\alpha} - \alpha^K \max_{x \in S} |J(x)| \\ &\leq (T^K J)(x_0) \\ &\leq J^*(x_0) + \frac{\alpha^K M}{1-\alpha} + \alpha^K \max_{x \in S} |J(x)|, \end{aligned} \quad (2.3)$$

and by taking the limit as  $K \rightarrow \infty$ , the result follows. **Q.E.D.**

Note that based on the preceding proposition, the DP algorithm may be used to compute at least an approximation to  $J^*$ . This computational method together with some additional methods will be examined in the next section.

Given any stationary policy  $\mu$ , we can consider a modified discounted problem, which is the same as the original except that the control constraint set contains only one element for each state  $x$ , the control  $\mu(x)$ ; that is, the control constraint set is  $\bar{U}(x) = \{\mu(x)\}$  instead of  $U(x)$ . Proposition 2.1 applies to this modified problem and yields the following corollary:

**Corollary 2.1.1:** For every stationary policy  $\mu$ , the associated cost function satisfies

$$J_\mu(x) = \lim_{N \rightarrow \infty} (T_\mu^N J)(x), \quad \text{for all } x \in S. \quad (2.4)$$

The next proposition shows that  $J^*$  is the unique solution of Bellman's equation.

**Proposition 2.2: (Bellman's Equation)** The optimal cost function  $J^*$  satisfies

$$J^*(x) = \min_{u \in U(x)} E \left\{ g(x, u, w) + \alpha J^*(f(x, u, w)) \right\}, \quad \text{for all } x \in S, \quad (2.5)$$

or, equivalently,

$$J^* = T J^*. \quad (2.6)$$

Furthermore,  $J^*$  is the unique solution of this equation within the class of bounded functions.

**Proof:** From Eq. (2.3), we have for all  $x \in S$  and  $N$ ,

$$J^*(x) - \frac{\alpha^N M}{1-\alpha} \leq (T^N J_0)(x) \leq J^*(x) + \frac{\alpha^N M}{1-\alpha},$$

where  $J_0$  is the zero function [ $J_0(x) = 0$  for all  $x \in S$ ]. Applying the mapping  $T$  to this relation and using the Monotonicity Lemma 1.1 as well as Lemma 1.2, we obtain for all  $x \in S$  and  $N$

$$(TJ^*)(x) - \frac{\alpha^{N+1} M}{1-\alpha} \leq (T^{N+1} J_0)(x) \leq (TJ^*)(x) + \frac{\alpha^{N+1} M}{1-\alpha}.$$

Since  $(T^{N+1} J_0)(x)$  converges to  $J^*(x)$  (cf. Prop. 2.1), by taking the limit as  $N \rightarrow \infty$  in the preceding relation, we obtain  $J^* = TJ^*$ .

To show uniqueness, observe that if  $J$  is bounded and satisfies  $J = TJ$ , then  $J = \lim_{N \rightarrow \infty} T^N J$ , so by Prop. 2.1, we have  $J = J^*$ . **Q.E.D.**

Based on the same reasoning we used to obtain Cor. 2.1.1 from Prop. 2.1, we have:

**Corollary 2.2.1:** For every stationary policy  $\mu$ , the associated cost function satisfies

$$J_\mu(x) = \underset{w}{\mathbb{E}} \{g(x, \mu(x), w) + \alpha J_\mu(f(x, \mu(x), w))\}, \quad \text{for all } x \in S, \quad (2.7)$$

or, equivalently,

$$J_\mu = T_\mu J_\mu.$$

Furthermore,  $J_\mu$  is the unique solution of this equation within the class of bounded functions.

The next proposition characterizes stationary optimal policies.

**Proposition 2.3: (Necessary and Sufficient Condition for Optimality)** A stationary policy  $\mu$  is optimal if and only if  $\mu(x)$  attains the minimum in Bellman's equation (2.5) for each  $x \in S$ ; that is,

$$TJ^* = T_\mu J^*. \quad (2.8)$$

**Proof:** If  $TJ^* = T_\mu J^*$ , then using Bellman's equation ( $J^* = TJ^*$ ), we have  $J^* = T_\mu J^*$ , so by the uniqueness part of Cor. 2.2.1, we obtain  $J^* = J_\mu$ ;

that is,  $\mu$  is optimal. Conversely, if the stationary policy  $\mu$  is optimal, we have  $J^* = J_\mu$ , which by Cor. 2.2.1, yields  $J^* = T_\mu J^*$ . Combining this with Bellman's equation ( $J^* = TJ^*$ ), we obtain  $TJ^* = T_\mu J^*$ . **Q.E.D.**

Note that Prop. 2.3 implies the existence of an optimal stationary policy when the minimum in the right-hand side of Bellman's equation is attained for all  $x \in S$ . In particular, when  $U(x)$  is finite for each  $x \in S$ , an optimal stationary policy is guaranteed to exist.

We finally show the following convergence rate estimate for any bounded function  $J$ :

$$\max_{x \in S} |(T^k J)(x) - J^*(x)| \leq \alpha^k \max_{x \in S} |J(x) - J^*(x)|, \quad k = 0, 1, \dots$$

This relation is obtained by combining Bellman's equation and the following result:

**Proposition 2.4:** For any two bounded functions  $J : S \mapsto \mathbb{R}$ ,  $J' : S \mapsto \mathbb{R}$ , and for all  $k = 0, 1, \dots$ , there holds

$$\max_{x \in S} |(T^k J)(x) - (T^k J')(x)| \leq \alpha^k \max_{x \in S} |J(x) - J'(x)|. \quad (2.9)$$

**Proof:** Denote

$$c = \max_{x \in S} |J(x) - J'(x)|.$$

Then we have

$$J(x) - c \leq J'(x) \leq J(x) + c, \quad x \in S.$$

Applying  $T^k$  in this relation and using the Monotonicity Lemma 1.1 as well as Lemma 1.2, we obtain

$$(T^k J)(x) - \alpha^k c \leq (T^k J')(x) \leq (T^k J)(x) + \alpha^k c, \quad x \in S.$$

It follows that

$$|(T^k J)(x) - (T^k J')(x)| \leq \alpha^k c, \quad x \in S,$$

which proves the result. **Q.E.D.**

As earlier, we have:

**Corollary 2.4.1:** For any two bounded functions  $J : S \mapsto \mathbb{R}$ ,  $J' : S \mapsto \mathbb{R}$ , and any stationary policy  $\mu$ , we have

$$\max_{x \in S} |(T_\mu^k J)(x) - (T_\mu^k J')(x)| \leq \alpha^k \max_{x \in S} |J(x) - J'(x)|, \quad k = 0, 1, \dots$$

### Example 2.1 (Machine Replacement)

Consider an infinite horizon discounted version of a problem we formulated in Section 1.1 of Vol. I. Here, we want to operate efficiently a machine that can be in any one of  $n$  states, denoted  $1, 2, \dots, n$ . State 1 corresponds to a machine in perfect condition. The transition probabilities  $p_{ij}$  are given. There is a cost  $g(i)$  for operating for one time period the machine when it is in state  $i$ . The options at the start of each period are to (a) let the machine operate one more period in the state it currently is, or (b) replace the machine with a new machine (state 1) at a cost  $R$ . Once replaced, the machine is guaranteed to stay in state 1 for one period; in subsequent periods, it may deteriorate to states  $j \geq 1$  according to the transition probabilities  $p_{1j}$ . We assume an infinite horizon and a discount factor  $\alpha \in (0, 1)$ , so the theory of this section applies.

Bellman's equation (cf. Prop. 2.2) takes the form

$$J^*(i) = \min \left[ R + g(1) + \alpha J^*(1), g(i) + \alpha \sum_{j=1}^n p_{ij} J^*(j) \right], \quad i = 1, \dots, n.$$

By Prop. 2.3, a stationary policy is optimal if it replaces at states  $i$  where

$$R + g(1) + \alpha J^*(1) < g(i) + \alpha \sum_{j=1}^n p_{ij} J^*(j),$$

and it does not replace at states  $i$  where

$$R + g(1) + \alpha J^*(1) > g(i) + \alpha \sum_{j=1}^n p_{ij} J^*(j).$$

We can use the convergence of the DP algorithm (cf. Prop. 2.1) to characterize the optimal cost function using properties of the finite horizon cost functions. In particular, the DP algorithm starting from the zero function takes the form

$$J_0(i) = 0,$$

$$(TJ_0)(i) = \min [R + g(1), g(i)],$$

$$(T^k J_0)(i) = \min \left[ R + g(1) + \alpha(T^{k-1} J_0)(1), g(i) + \alpha \sum_{j=1}^n p_{ij} (T^{k-1} J_0)(j) \right].$$

Assume that  $g(i)$  is nondecreasing in  $i$ , and that the transition probabilities satisfy

$$\sum_{j=1}^n p_{ij} J(j) \leq \sum_{j=1}^n p_{(i+1)j} J(j), \quad i = 1, \dots, n-1, \quad (2.10)$$

for all functions  $J(j)$ , which are monotonically nondecreasing in  $i$ . It can be shown that this assumption is satisfied if and only if, for every  $k$ ,  $\sum_{j=1}^n p_{ij} J(j)$  is monotonically nondecreasing in  $i$  (see [Ros83b], p. 252). The assumption (2.10) is satisfied in particular if

$$p_{ij} = 0, \quad \text{if } j < i,$$

i.e., the machine cannot go to a better state with usage, and

$$p_{ij} \leq p_{(i+1)j}, \quad \text{if } i < j,$$

i.e., there is greater chance of ending at a bad state  $j$  if we start at a worse state  $i$ . Since  $g(i)$  is nondecreasing in  $i$ , we have that  $(TJ_0)(i)$  is nondecreasing in  $i$ , and in view of the assumption (2.10) on the transition probabilities, the same is true for  $(T^2 J_0)(i)$ . Similarly, it is seen that, for all  $k$ ,  $(T^k J_0)(i)$  is nondecreasing in  $i$  and so is its limit

$$J^*(i) = \lim_{k \rightarrow \infty} (T^k J_0)(i).$$

This is intuitively clear: the optimal cost should not decrease as the machine starts at a worse initial state. It follows that the function

$$g(i) + \alpha \sum_{j=1}^n p_{ij} J^*(j)$$

is nondecreasing in  $i$ . Consider the set of states

$$S_R = \left\{ i \mid R + g(1) + \alpha J^*(1) \leq g(i) + \alpha \sum_{j=1}^n p_{ij} J^*(j) \right\},$$

and let

$$i^* = \begin{cases} \text{smallest state in } S_R & \text{if } S_R \text{ is nonempty,} \\ n+1 & \text{otherwise.} \end{cases}$$

Then, an optimal policy takes the form

$$\text{replace if and only if } i \geq i^*,$$

as shown in Fig. 1.2.1.

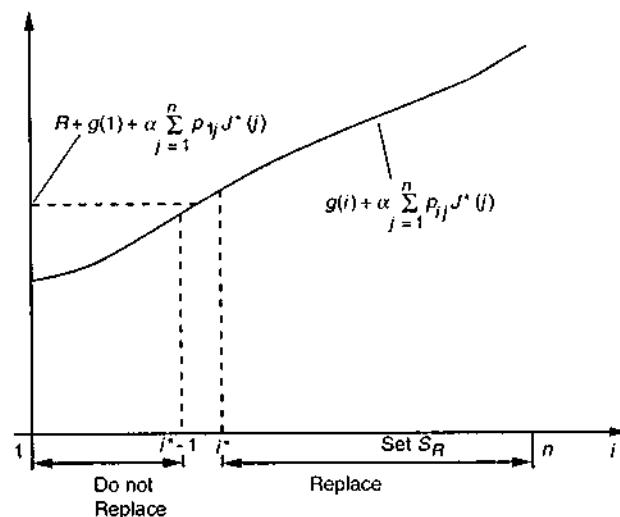


Figure 1.2.1 Determining the optimal policy in the machine replacement example.

### 1.3 FINITE-STATE SYSTEMS – COMPUTATIONAL METHODS

In this section we discuss several alternative approaches for numerically solving the discounted problem with bounded cost per stage. The first approach, value iteration, is essentially the DP algorithm and yields in the limit the optimal cost function and an optimal policy, as discussed in the preceding section. We will describe some variations aimed at accelerating convergence. Two other approaches, policy iteration and linear programming, terminate in a finite number of iterations (assuming the number of states and controls are finite). However, when the number of states is large, these approaches are impractical because of large overhead per iteration. Another approach, adaptive aggregation, bridges the gap between value iteration and policy iteration, and in a sense combines the best features of both methods.

In Section 2.3 we will consider some additional methods, which are well-suited for dynamic systems that are hard to model but relatively easy to simulate. In particular, we will assume in Section 2.3 that the transition probabilities of the problem are unknown, but the system's dynamics and cost structure can be observed through simulation. We will then discuss the methods of temporal differences and  $Q$ -learning, which also provide conceptual vehicles for approximate forms of value iteration and policy

iteration using, for example, neural networks.

Throughout this section we assume a discounted problem (Assumption D holds). We further assume that the state, control, and disturbance spaces underlying the problem are finite sets, so that we are dealing in effect with the control of a finite-state Markov chain.

We first translate some of our earlier analysis in a notation that is more convenient for Markov chains. Let the state space  $S$  consist of  $n$  states denoted by  $1, 2, \dots, n$ :

$$S = \{1, 2, \dots, n\}.$$

We denote by  $p_{ij}(u)$  the transition probabilities

$$p_{ij}(u) = P(x_{k+1} = j \mid x_k = i, u_k = u), \quad i, j \in S, u \in U(i).$$

These transition probabilities may be given a priori or may be calculated from the system equation

$$x_{k+1} = f(x_k, u_k, w_k)$$

and the known probability distribution  $P(\cdot \mid x, u)$  of the input disturbance  $w_k$ . Indeed, we have

$$p_{ij}(u) = P(W_{ij}(u) \mid i, u),$$

where  $W_{ij}(u)$  is the (finite) set

$$W_{ij}(u) = \{w \in D \mid f(i, u, w) = j\}.$$

To simplify notation, we assume that the cost per stage does not depend on  $w$ . This amounts to using expected cost per stage in all calculations, which makes no essential difference in the definitions of the mappings  $T$  and  $T_\mu$  of Eqs. (1.4) and (1.5), and in the subsequent analysis. Thus, if  $\tilde{g}(i, u, j)$  is the cost of using  $u$  at state  $i$  and moving to state  $j$ , we use as cost per stage the expected cost  $g(i, u)$  given by

$$g(i, u) = \sum_{j=1}^n p_{ij}(u) \tilde{g}(i, u, j).$$

The mappings  $T$  and  $T_\mu$  of Eqs. (1.4) and (1.5) can be written as

$$(TJ)(i) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J(j) \right], \quad i = 1, 2, \dots, n,$$

$$(T_\mu J)(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^n p_{ij}(\mu(i))J(j), \quad i = 1, 2, \dots, n.$$

Any function  $J$  on  $S$ , as well as the functions  $TJ$  and  $T_\mu J$  may be represented by the  $n$ -dimensional vectors

$$J = \begin{pmatrix} J(1) \\ \vdots \\ J(n) \end{pmatrix}, \quad TJ = \begin{pmatrix} (TJ)(1) \\ \vdots \\ (TJ)(n) \end{pmatrix}, \quad T_\mu J = \begin{pmatrix} (T_\mu J)(1) \\ \vdots \\ (T_\mu J)(n) \end{pmatrix}.$$

For a stationary policy  $\mu$ , we denote by  $P_\mu$  the transition probability matrix

$$P_\mu = \begin{pmatrix} p_{11}(\mu(1)) & \cdots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \cdots & p_{nn}(\mu(n)) \end{pmatrix},$$

and by  $g_\mu$  the cost vector

$$g_\mu = \begin{pmatrix} g(1, \mu(1)) \\ \vdots \\ g(n, \mu(n)) \end{pmatrix}.$$

We can then write in vector notation

$$T_\mu J = g_\mu + \alpha P_\mu J.$$

The cost function  $J_\mu$  corresponding to a stationary policy  $\mu$  is, by Cor. 2.2.1, the unique solution of the equation

$$J_\mu = T_\mu J_\mu = g_\mu + \alpha P_\mu J_\mu.$$

This equation should be viewed as a system of  $n$  linear equations with  $n$  unknowns, the components  $J_\mu(i)$  of the  $n$ -dimensional vector  $J_\mu$ . The equation can also be written as

$$(I - \alpha P_\mu)J_\mu = g_\mu,$$

or, equivalently,

$$J_\mu = (I - \alpha P_\mu)^{-1}g_\mu, \quad (3.1)$$

where  $I$  denotes the  $n \times n$  identity matrix. The invertibility of the matrix  $I - \alpha P_\mu$  is assured since we have proved that the system of equations representing  $J_\mu = T_\mu J_\mu$  has a unique solution for any vector  $g_\mu$  (cf. Cor. 2.2.1). For another way to see that  $I - \alpha P_\mu$  is an invertible matrix, note that the eigenvalues of any transition probability matrix lie within the unit circle of the complex plane. Thus no eigenvalue of  $\alpha P_\mu$  can be equal to 1, which is the necessary and sufficient condition for  $I - \alpha P_\mu$  to be invertible.

### 1.3.1 Value Iteration and Error Bounds

Here we start with any  $n$ -dimensional vector  $J$  and successively compute  $TJ$ ,  $T^2J$ , ... By Prop. 2.1, we have for all  $i$

$$\lim_{k \rightarrow \infty} (T^k J)(i) = J^*(i).$$

Furthermore, by Prop. 2.1 [using  $J' = J^*$  in Eq. (2.9)], the error sequence  $|(T^k J)(i) - J^*(i)|$  is bounded by a constant multiple of  $\alpha^k$ , for all  $i \in S$ . This method is called *value iteration* or *successive approximation*. The method can be substantially improved thanks to certain monotonic error bounds, which are easily obtained as a byproduct of the computation.

The following argument is helpful in understanding the nature of these bounds. Let us first break down the cost of a stationary policy  $\mu$  into the first stage cost and the remainder:

$$J_\mu(i) = g(i, \mu(i)) + \sum_{k=1}^{\infty} \alpha^k E\{g(x_k, \mu(x_k)) \mid x_0 = i\}.$$

It follows that

$$g_\mu + \left( \frac{\alpha \beta}{1 - \alpha} \right) c \leq J_\mu \leq g_\mu + \left( \frac{\alpha \bar{\beta}}{1 - \alpha} \right) c, \quad (3.2)$$

where  $c$  is the unit vector,  $c = (1, 1, \dots, 1)^t$ , and  $\underline{\beta}$  and  $\bar{\beta}$  are the minimum and maximum cost per stage:

$$\underline{\beta} = \min_i g(i, \mu(i)), \quad \bar{\beta} = \max_i g(i, \mu(i)).$$

Using the definition of  $\underline{\beta}$  and  $\bar{\beta}$ , we can strengthen the bounds (3.2) as follows:

$$\left( \frac{\underline{\beta}}{1 - \alpha} \right) c \leq g_\mu + \left( \frac{\alpha \underline{\beta}}{1 - \alpha} \right) c \leq J_\mu \leq g_\mu + \left( \frac{\alpha \bar{\beta}}{1 - \alpha} \right) c \leq \left( \frac{\bar{\beta}}{1 - \alpha} \right) c. \quad (3.3)$$

These bounds will now be applied in the context of the value iteration method.

Suppose that we have a vector  $J$  and we compute

$$T_\mu J = g_\mu + \alpha P_\mu J.$$

By subtracting this equation from the relation

$$J_\mu = g_\mu + \alpha P_\mu J_\mu,$$

we obtain

$$J_\mu - J = T_\mu J - J + \alpha P_\mu(J_\mu - J).$$

This equation can be viewed as a *variational* form of the equation  $J_\mu = T_\mu J_\mu$ , and implies that  $J_\mu - J$  is the cost vector associated with the stationary policy  $\mu$  and a cost per stage vector equal to  $T_\mu J - J$ . Therefore, the bounds (3.3) apply with  $J_\mu$  replaced by  $J_\mu - J$  and  $g_\mu$  replaced by  $T_\mu J - J$ . It follows that

$$\begin{aligned} \left(\frac{\gamma}{1-\alpha}\right)c &\leq T_\mu J - J + \left(\frac{\alpha\gamma}{1-\alpha}\right)c \\ &\leq J_\mu - J \\ &\leq T_\mu J - J + \left(\frac{\alpha\bar{\gamma}}{1-\alpha}\right)c \\ &\leq \left(\frac{\bar{\gamma}}{1-\alpha}\right)c, \end{aligned}$$

where

$$\underline{\gamma} = \min_i [(T_\mu J)(i) - J(i)], \quad \bar{\gamma} = \max_i [(T_\mu J)(i) - J(i)].$$

Equivalently, for every vector  $J$ , we have

$$J + \frac{c}{\alpha} \leq T_\mu J + \underline{c}c \leq J_\mu \leq T_\mu J + \bar{c}c \leq J + \frac{\bar{c}}{\alpha}c,$$

where

$$\underline{c} = \frac{\alpha\gamma}{1-\alpha}, \quad \bar{c} = \frac{\alpha\bar{\gamma}}{1-\alpha}.$$

The following proposition is obtained by a more sophisticated application of the preceding argument.

**Proposition 3.1:** For every vector  $J$ , state  $i$ , and  $k$ , we have

$$\begin{aligned} (T^k J)(i) + \underline{c}_k &\leq (T^{k+1} J)(i) + \underline{c}_{k+1} \\ &\leq J^*(i) \\ &\leq (T^{k+1} J)(i) + \bar{c}_{k+1} \\ &\leq (T^k J)(i) + \bar{c}_k, \end{aligned} \tag{3.4}$$

where

$$\underline{c}_k = \frac{\alpha}{1-\alpha} \min_{i=1,\dots,n} [(T^k J)(i) - (T^{k-1} J)(i)], \tag{3.5}$$

$$\bar{c}_k = \frac{\alpha}{1-\alpha} \max_{i=1,\dots,n} [(T^k J)(i) - (T^{k-1} J)(i)]. \tag{3.6}$$

**Proof:** Denote

$$\underline{z} = \min_{i=1,\dots,n} [(TJ)(i) - J(i)].$$

We have

$$J + \underline{z}c \leq TJ. \tag{3.7}$$

Applying  $T$  to both sides and using the monotonicity of  $T$ , we have

$$TJ + \alpha\underline{z}c \leq T^2J,$$

and, combining this relation with Eq. (3.7), we obtain

$$J + (1+\alpha)\underline{z}c \leq TJ + \alpha\underline{z}c \leq T^2J. \tag{3.8}$$

This process can be repeated, first applying  $T$  to obtain

$$TJ + (\alpha + \alpha^2)\underline{z}c \leq T^2J + \alpha^2\underline{z}c \leq T^3J,$$

and then using Eq. (3.7) to write

$$J + (1 + \alpha + \alpha^2)\underline{z}c \leq TJ + (\alpha + \alpha^2)\underline{z}c \leq T^2J + \alpha^2\underline{z}c \leq T^3J.$$

After  $k$  steps, this results in the inequalities

$$\begin{aligned} J + \left(\sum_{i=0}^k \alpha^i\right)\underline{z}c &\leq TJ + \left(\sum_{i=1}^k \alpha^i\right)\underline{z}c \\ &\leq T^2J + \left(\sum_{i=2}^k \alpha^i\right)\underline{z}c \\ &\leq \dots \\ &\leq T^{k+1}J. \end{aligned}$$

Taking the limit as  $k \rightarrow \infty$  and using the equality  $\underline{c}_1 = \alpha\underline{z}/(1-\alpha)$ , we obtain

$$J + \left(\frac{\underline{c}_1}{\alpha}\right)c \leq TJ + \underline{c}_1c \leq T^2J + \alpha\underline{c}_1c \leq J^*, \tag{3.9}$$

where  $\underline{c}_1$  is defined by Eq. (3.5). Replacing  $J$  by  $T^k J$  in this inequality, we have

$$T^{k+1}J + \underline{c}_{k+1}c \leq J^*,$$

which is the second inequality in Eq. (3.4).

From Eq. (3.8), we have

$$\alpha\underline{z} \leq \min_{i=1,\dots,n} [(T^2J)(i) - (TJ)(i)].$$

and consequently

$$\alpha c_1 \leq c_2.$$

Using this relation in Eq. (3.9) yields

$$TJ + \underline{c}_1 c \leq T^2 J + \underline{c}_2 c,$$

and replacing  $J$  by  $T^{k-1}J$ , we have the first inequality in Eq. (3.1). An analogous argument shows the last two inequalities in Eq. (3.4). Q.E.D.

We note that the preceding proof does not rely on the finiteness of the state space, and indeed Prop. 3.1 can be proved for an infinite state space (see also Exercise 1.9). The following example demonstrates the nature of the error bounds.

### Example 3.1 (Illustration of the Error Bounds)

Consider a problem where there are two states and two controls

$$S = \{1, 2\}, \quad C = \{u^1, u^2\}.$$

The transition probabilities corresponding to the controls  $u^1$  and  $u^2$  are as shown in Fig. 1.3.1; that is, the transition probability matrices are

$$P(u^1) = \begin{pmatrix} p_{11}(u^1) & p_{12}(u^1) \\ p_{21}(u^1) & p_{22}(u^1) \end{pmatrix} = \begin{pmatrix} 3/4 & 1/4 \\ 3/4 & 1/4 \end{pmatrix},$$

$$P(u^2) = \begin{pmatrix} p_{11}(u^2) & p_{12}(u^2) \\ p_{21}(u^2) & p_{22}(u^2) \end{pmatrix} = \begin{pmatrix} 1/4 & 3/4 \\ 1/4 & 3/4 \end{pmatrix}.$$

The transition costs are

$$g(1, u^1) = 2, \quad g(1, u^2) = 0.5, \quad g(2, u^1) = 1, \quad g(2, u^2) = 3,$$

and the discount factor is  $\alpha = 0.9$ . The mapping  $T$  is given for  $i = 1, 2$  by

$$(TJ)(i) = \min \left\{ g(i, u^1) + \alpha \sum_{j=1}^2 p_{ij}(u^1) J(j), g(i, u^2) + \alpha \sum_{j=1}^2 p_{ij}(u^2) J(j) \right\}.$$

The scalars  $\underline{c}_k$  and  $\bar{c}_k$  of Eqs. (3.5) and (3.6) are given by

$$\underline{c}_k = \frac{\alpha}{1-\alpha} \min \{ (T^k J)(1) - (T^{k-1} J)(1), (T^k J)(2) - (T^{k-1} J)(2) \},$$

$$\bar{c}_k = \frac{\alpha}{1-\alpha} \max \{ (T^k J)(1) - (T^{k-1} J)(1), (T^k J)(2) - (T^{k-1} J)(2) \}.$$

The results of the value iteration method starting with the zero function  $J_0$  [ $J_0(1) = J_0(2) = 0$ ] are shown in Fig. 1.3.2 and illustrate the power of the error bounds.

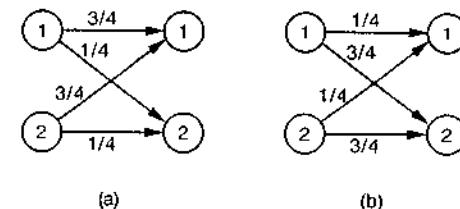


Figure 1.3.1 State transition diagram for Example 3.1: (a)  $u = u^1$ ; (b)  $u = u^2$ .

$k$	$(T^k J_0)(1)$	$(T^k J_0)(2)$	$(T^k J_0)(1) + \underline{c}_k$	$(T^k J_0)(1) + \bar{c}_k$	$(T^k J_0)(2) + \underline{c}_k$	$(T^k J_0)(2) + \bar{c}_k$
0	0	0				
1	0.500	1.000	5.000	9.500	5.500	10.000
2	1.287	1.562	6.350	8.375	6.625	8.650
3	1.841	2.220	6.856	7.767	7.232	8.144
4	2.414	2.745	7.129	7.540	7.460	7.870
5	2.896	3.247	7.232	7.417	7.583	7.768
6	3.343	3.686	7.287	7.371	7.629	7.712
7	3.740	4.086	7.308	7.345	7.654	7.692
8	4.099	4.441	7.319	7.336	7.663	7.680
9	4.422	4.767	7.324	7.331	7.669	7.676
10	4.713	5.057	7.326	7.329	7.671	7.671
11	4.974	5.319	7.327	7.328	7.672	7.673
12	5.209	5.554	7.327	7.328	7.672	7.673
13	5.421	5.766	7.327	7.328	7.672	7.673
14	5.612	5.957	7.328	7.328	7.672	7.672
15	5.783	6.128	7.328	7.328	7.672	7.672

Figure 1.3.2 Performance of the value iteration method with and without the error bounds of Prop. 3.1 for the problem of Example 3.1.

### Termination Issues - Optimality of the Obtained Policy

Let us now discuss how to use the error bounds to obtain an optimal

or near-optimal policy in a finite number of value iterations. We first note that given any  $J$ , if we compute  $TJ$  and a policy  $\mu$  attaining the minimum in the calculation of  $TJ$ , i.e.,  $T_\mu J = TJ$ , then we can obtain the following bound on the suboptimality of  $\mu$ :

$$\max_i [J_\mu(i) - J^*(i)] \leq \frac{\alpha}{1-\alpha} \left( \max_i [(TJ)(i) - J(i)] - \min_i [(TJ)(i) - J(i)] \right). \quad (3.10)$$

To see this, apply Eq. (3.4) with  $k = 1$  to obtain for all  $i$

$$\underline{c}_1 \leq J^*(i) - (TJ)(i) \leq \bar{c}_1,$$

and also apply Eq. (3.4) with  $k = 1$  and with  $T_\mu$  replacing  $T$  to obtain

$$\underline{c}_1 \leq J_\mu(i) - (T_\mu J)(i) = J_\mu(i) - (TJ)(i) \leq \bar{c}_1.$$

Subtracting the above two equations, we obtain the estimate (3.10).

In practice, one terminates the value iteration method when the difference  $(\bar{c}_k - \underline{c}_k)$  of the error bounds becomes sufficiently small. One can then take as final estimate of  $J^*$  the "median"

$$\hat{J}_k = T^k J + \left( \frac{\bar{c}_k + \underline{c}_k}{2} \right) c \quad (3.11)$$

or the "average"

$$\check{J}_k = T^k J + \frac{\alpha}{n(1-\alpha)} \sum_{i=1}^n ((T^k J)(i) - (T^{k-1} J)(i)) c. \quad (3.12)$$

Both of these vectors lie in the region delineated by the error bounds. Then, the estimate (3.10) provides a bound on the suboptimality of the policy  $\mu$  attaining the minimum in the calculation of  $T^k J$ .

The bound (3.10) can also be used to show that after a sufficiently large number of value iterations, the stationary policy  $\mu^k$  that attains the minimum in the  $k$ th value iteration [i.e.,  $(T_{\mu^k} T^{k-1})J = T^k J$ ] is optimal. Indeed, since the number of stationary policies is finite, there exists an  $\bar{\epsilon} > 0$  such that if a stationary policy  $\mu$  satisfies

$$\max_i [J_\mu(i) - J^*(i)] < \bar{\epsilon},$$

then  $\mu$  is optimal. Now let  $\bar{k}$  be such that for all  $k \geq \bar{k}$  we have

$$\frac{\alpha}{1-\alpha} \left( \max_i [(T^k J)(i) - (T^{k-1} J)(i)] - \min_i [(T^k J)(i) - (T^{k-1} J)(i)] \right) < \bar{\epsilon}.$$

Then from Eq. (3.10) we see that for all  $k \geq \bar{k}$ , the stationary policy that attains the minimum in the  $k$ th value iteration is optimal.

### Rate of Convergence

To analyze the rate of convergence of value iteration with error bounds, assume that there is a stationary policy  $\mu^*$  that attains the minimum over  $\mu$  in the relation

$$\min_\mu T_\mu T^{k-1} J = T^k J$$

for all  $k$  sufficiently large, so that eventually the method reduces to the linear iteration

$$J := g_{\mu^*} + \alpha P_{\mu^*} J.$$

In view of our preceding discussion, this is true for example if  $\mu^*$  is a unique optimal stationary policy. Generally the rate of convergence of linear iterations is governed by the maximum eigenvalue modulus of the matrix of the iteration [which is  $\alpha$  in our case, since any transition probability matrix has a unit eigenvalue with corresponding eigenvector  $c = (1, 1, \dots, 1)'$ , while all other eigenvalues lie within the unit circle of the complex plane].

It turns out, however, that when error bounds are used, the rate at which the iterates  $\hat{J}_k$  and  $\check{J}_k$  of Eqs. (3.11) and (3.12) approach the optimal cost vector  $J^*$  is governed by the modulus of the *subdominant* eigenvalue of the transition probability matrix  $P_{\mu^*}$ , that is, the eigenvalue with second largest modulus. The proof of this is outlined in Exercise 1.8. For a sketch of the ideas involved, let  $\lambda_1, \dots, \lambda_n$  be the eigenvalues of  $P_{\mu^*}$ , ordered according to decreasing modulus; that is

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|,$$

with  $\lambda_1$  equal to 1 and  $\lambda_2$  being the subdominant eigenvalue. Assume that there is a set of linearly independent eigenvectors  $c_1, c_2, \dots, c_n$  corresponding to  $\lambda_1, \lambda_2, \dots, \lambda_n$  with  $c_1 = c = (1, 1, \dots, 1)'$ . Then the initial error  $J - J_{\mu^*}$  can be expressed as a linear combination of the eigenvectors

$$J - J_{\mu^*} = \xi_1 c + \sum_{j=2}^n \xi_j c_j$$

for some scalars  $\xi_1, \xi_2, \dots, \xi_n$ . Since  $T_{\mu^*} J = g_{\mu^*} + \alpha P_{\mu^*} J$  and  $J_{\mu^*} = g_{\mu^*} + \alpha P_{\mu^*} J_{\mu^*}$ , successive errors are related by

$$T_{\mu^*} J - J_{\mu^*} = \alpha P_{\mu^*} (J - J_{\mu^*}), \quad \text{for all } J.$$

Thus the error after  $k$  iterations can be written as

$$T_{\mu^*}^k J - J_{\mu^*} = \alpha^k \xi_1 c + \alpha^k \sum_{j=2}^n \lambda_j^k \xi_j c_j.$$

Using the error bounds of Prop. 3.1 amounts to a translation of  $T_{\mu^*}^k J$  along the vector  $c$ . Thus, at best, the error bounds are tight enough to eliminate the component  $\alpha^k \xi_1 c$  of the error, but cannot affect the remaining term  $\alpha^k \sum_{j=2}^n \lambda_j^k \xi_j c_j$ , which diminishes like  $\alpha^k |\lambda_2|^k$  with  $\lambda_2$  being the subdominant eigenvalue.

### Problems where Convergence is Slow

In Example 3.1, the convergence of value iteration with the error bounds is very fast. For this example, it can be verified that  $\mu^*(1) = u^2$ ,  $\mu^*(2) = u^1$ , and that

$$P_{\mu^*} = \begin{pmatrix} 1/4 & 3/4 \\ 3/4 & 1/4 \end{pmatrix}.$$

The eigenvalues of  $P_{\mu^*}$  can be calculated to be  $\lambda_1 = 1$  and  $\lambda_2 = -\frac{1}{2}$ , which explains the fast convergence, since the modulus  $1/2$  of the subdominant eigenvalue  $\lambda_2$  is considerably smaller than one. On the other hand, there are situations where convergence of the method even with the use of error bounds is very slow. For example, suppose that  $P_{\mu^*}$  is block diagonal with two or more blocks, or more generally, that  $P_{\mu^*}$  corresponds to a system with more than one recurrent class of states (see Appendix D of Vol. I). Then it can be shown that the subdominant eigenvalue  $\lambda_2$  is equal to 1, and convergence is typically slow when  $\alpha$  is close to 1.

As an example, consider the following three simple deterministic problems, each having a single policy and more than one recurrent class of states:

*Problem 1:*  $n = 3$ ,  $P_\mu$  = three-dimensional identity,  $g(i, \mu(i)) = i$ .

*Problem 2:*  $n = 5$ ,  $P_\mu$  = five-dimensional identity,  $g(i, \mu(i)) = i$ .

*Problem 3:*  $n = 6$ ,  $g(i, \mu(i)) \approx i$  and

$$P_\mu = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Figure 1.3.3 shows the number of iterations needed by the value iteration method with and without the error bounds of Prop. 3.1 to find  $J_\mu$  within an error per coordinate of less than or equal to  $10^{-6} \max_i |J_\mu(i)|$ . The starting function in all cases was taken to be zero. The performance is rather unsatisfactory but, nonetheless, is typical of situations where the subdominant eigenvalue modulus of the optimal transition probability matrix is close to 1. One possible approach to improve the performance of value iteration for such problems is based on the adaptive aggregation method to be discussed in Section 1.3.3.

	Pr. 1 $\alpha = .9$	Pr. 1 $\alpha = .99$	Pr. 2 $\alpha = .9$	Pr. 2 $\alpha = .99$	Pr. 3 $\alpha = .9$	Pr. 3 $\alpha = .99$
W/out bounds	131	1374	131	1374	132	1392
With bounds	127	1333	129	1352	131	1374

**Figure 1.3.3** Number of iterations for the value iteration method with and without error bounds. The problems are deterministic. Because the subdominant eigenvalue of the transition probability matrix is equal to 1, the error bounds are ineffective.

### Elimination of Nonoptimal Actions in Value Iteration

We know from Prop. 2.3 that, if  $\tilde{u} \in U(i)$  is such that

$$g(i, \tilde{u}) + \alpha \sum_{j=1}^n p_{ij}(\tilde{u}) J^*(j) > J^*(i),$$

then  $\tilde{u}$  cannot be optimal at state  $i$ ; that is, for every optimal stationary policy  $\mu$ , we have  $\mu(i) \neq \tilde{u}$ . Therefore, if we are sure that the above inequality holds, we can safely eliminate  $\tilde{u}$  from the admissible set  $U(i)$ . While we cannot check this inequality, since we do not know the optimal cost function  $J^*$ , we can guarantee that it holds if

$$g(i, \tilde{u}) + \alpha \sum_{j=1}^n p_{ij}(\tilde{u}) \underline{J}(j) > \bar{J}(i), \quad (3.13)$$

where  $\bar{J}$  and  $\underline{J}$  are upper and lower bounds satisfying

$$\underline{J}(i) \leq J^*(i) \leq \bar{J}(i), \quad i = 1, \dots, n.$$

The preceding observation is the basis for a useful application of the error bounds given earlier in Prop. 3.1. As these bounds are computed in the course of the value iteration method, the inequality (3.13) can be simultaneously checked and nonoptimal actions can be eliminated from the admissible set with attendant savings in subsequent computations. Since the upper and lower bound functions  $\bar{J}$  and  $\underline{J}$  converge to  $J^*$ , it can be seen [taking into account the finiteness of the constraint set  $U(i)$ ] that eventually all nonoptimal  $\tilde{u} \in U(i)$  will be eliminated, thereby reducing the set  $U(i)$  after a finite number of iterations to the set of controls that are optimal at  $i$ . In this manner the computational requirements of value iteration can be substantially reduced. However, the amount of computer memory required to maintain the set of controls not as yet eliminated at each  $i \in S$  may be increased.

### Gauss-Seidel Version of Value Iteration

In the value iteration method described earlier, the estimate of the cost function is iterated for all states simultaneously. An alternative is to iterate one state at a time, while incorporating into the computation the interim results. This corresponds to using what is known as the *Gauss-Seidel method* for solving the nonlinear system of equations  $J = TJ$  (see [BeT89a] or [OrR70]).

For  $n$ -dimensional vectors  $J$ , define the mapping  $F$  by

$$(FJ)(1) = \min_{u \in U(1)} \left[ g(1, u) + \alpha \sum_{j=1}^n p_{1j}(u) J(j) \right] \quad (3.14)$$

and, for  $i = 2, \dots, n$ ,

$$(FJ)(i) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^{i-1} p_{ij}(u) (FJ)(j) + \alpha \sum_{j=i}^n p_{ij}(u) J(j) \right]. \quad (3.15)$$

In words,  $(FJ)(i)$  is computed by the same equation as  $(TJ)(i)$  except that the previously calculated values  $(FJ)(1), \dots, (FJ)(i-1)$  are used in place of  $J(1), \dots, J(i-1)$ . Note that the computation of  $FJ$  is as easy as the computation of  $TJ$  (unless a parallel computer is used, in which case the computation of  $TJ$  may potentially be obtained much faster than  $FJ$ ; see [Tsi89], [BeT94a] for a comparative analysis).

Consider now the value iteration method whereby we compute  $J, FJ, F^2J, \dots$ . The following propositions show that the method is valid and provide an indication of better performance over the earlier value iteration method.

**Proposition 3.2:** Let  $J, J'$  be two  $n$ -dimensional vectors. Then for any  $k = 0, 1, \dots$ ,

$$\max_{i \in S} |(F^k J)(i) - (F^k J')(i)| \leq \alpha^k \max_{i \in S} |J(i) - J'(i)|. \quad (3.16)$$

Furthermore, we have

$$(FJ^*)(i) = J^*(i), \quad i \in S, \quad (3.17)$$

$$\lim_{k \rightarrow \infty} (F^k J)(i) = J^*(i), \quad i \in S. \quad (3.18)$$

**Proof:** It is sufficient to prove Eq. (3.16) for  $k = 1$ . We have by the definition of  $F$  and Prop. 2.4,

$$|(FJ)(1) - (FJ')(1)| \leq \alpha \max_{i \in S} |J(i) - J'(i)|.$$

Also, using this inequality,

$$\begin{aligned} |(FJ)(2) - (FJ')(2)| &\leq \alpha \max \{ |(FJ)(1) - (FJ')(1)|, |J(2) - J'(2)|, \dots, \\ &\quad |J(n) - J'(n)| \} \\ &\leq \alpha \max_{i \in S} |J(i) - J'(i)|, \end{aligned}$$

Proceeding similarly, we have, for every  $i$  and  $j \leq i$ ,

$$|(FJ)(j) - (FJ')(j)| \leq \alpha \max_{i \in S} |J(i) - J'(i)|,$$

so Eq. (3.16) is proved for  $k = 1$ . The equation  $FJ^* = J^*$  follows from the definition (3.14) and (3.15) of  $F$ , and Bellman's equation  $J^* = TJ^*$ . The convergence property (3.18) follows from Eqs. (3.16) and (3.17). Q.E.D.

**Proposition 3.3:** If an  $n$ -dimensional vector  $J$  satisfies

$$J(i) \leq (TJ)(i) \leq J^*(i), \quad i = 1, \dots, n,$$

then

$$(T^k J)(i) \leq (F^k J)(i) \leq J^*(i), \quad i = 1, \dots, n, \quad k = 1, 2, \dots \quad (3.19)$$

**Proof:** The proof follows by using the definition (3.14) and (3.15) of  $F$ , and the monotonicity property of  $T$  (Lemma 1.1). Q.E.D.

The preceding proposition provides the main motivation for employing the mapping  $F$  in place of  $T$  in the value iteration method. The result indicates that the Gauss-Seidel version converges faster than the ordinary value iteration method. The faster convergence property can be substantiated by further analysis (see e.g., [BeT89a]) and has been confirmed in practice through extensive experimentation. This comparison is somewhat misleading, however, because the ordinary method will normally be used in conjunction with the error bounds of Prop. 3.4. One may also employ error bounds in the Gauss-Seidel version (see Exercise 1.9). However, there is no clear superiority of one method over the other when bounds are introduced. Furthermore, the ordinary method is better suited for parallel computation than the Gauss-Seidel version.

We note that there is a more flexible form of the Gauss-Seidel method, which selects states in arbitrary order to update their costs. This method maintains an approximation  $J$  to the optimal vector  $J^*$ , and at each iteration, it selects a state  $i$  and replaces  $J(i)$  by  $(TJ)(i)$ . The remaining values  $J(j)$ ,  $j \neq i$ , are left unchanged. The choice of the state  $i$  at each iteration is arbitrary, except for the restriction that all states are selected infinitely often. This method is an example of an *asynchronous fixed point iteration* and can be shown to converge to  $J^*$  starting from any initial  $J$ . Analyses of this type of method are given in [Ber82a], and in Chapter 6 of [BeT89a]; see also Exercise 1.15.

### Generic Rank-One Corrections

We may view value iteration coupled with the error bounds of Prop. 3.1 as a method that makes a correction to the results of value iteration along the unit vector  $e$ . It is possible to generalize the idea of correction along a fixed vector so that it works for any type of convergent linear iteration.

Let us consider the case of a single stationary policy  $\mu$  and an iteration of the form  $J := FJ$ , where

$$FJ = h_\mu + Q_\mu J.$$

Here,  $Q_\mu$  is a matrix with eigenvalues strictly within the unit circle, and  $h_\mu$  is a vector such that

$$J_\mu = FJ_\mu.$$

An example is the Gauss-Seidel iteration of Section 1.3.1, and some other examples are given in Exercises 1.4, 1.5, and 1.7, and in Section 5.3. Also, the value iteration method for stochastic shortest path problems and a single stationary policy, to be discussed in Section 2.2, is of the above form.

Consider in place of  $J := FJ$ , an iteration of the form

$$J := F\tilde{J},$$

where  $\tilde{J}$  is related to  $J$  by

$$\tilde{J} = J + \tilde{\gamma}d,$$

with  $d$  a fixed vector and  $\tilde{\gamma}$  a scalar to be selected in some optimal manner. In particular, consider choosing  $\tilde{\gamma}$  by minimizing over  $\gamma$

$$\|J + \gamma d - F(J + \gamma d)\|^2,$$

which, by denoting

$$z = Q_\mu d,$$

can be written as

$$\|J - FJ + \gamma(d - z)\|^2.$$

By setting to zero the derivative of this expression with respect to  $\gamma$ , it is straightforward to verify that the optimal solution is

$$\tilde{\gamma} = \frac{(d - z)^T(FJ - J)}{\|d - z\|^2}.$$

Thus the iteration  $J := F\tilde{J}$  can be written as

$$J := MJ,$$

where

$$MJ = FJ + \tilde{\gamma}z.$$

We note that this iteration requires only slightly more computation than the iteration  $J := FJ$ , since the vector  $z$  is computed once and the computation of  $\tilde{\gamma}$  is simple.

A key question of course is under what circumstances the iteration  $J := MJ$  converges faster than the iteration  $J := FJ$ , and whether indeed it converges at all to  $J_\mu$ . It is straightforward to verify that in the case where  $Q_\mu = \alpha P_\mu$  and  $d = e$ , the iteration  $J := MJ$  can be written as

$$J := T_\mu J + \frac{\alpha}{n(1-\alpha)} \sum_{i=1}^n ((T_\mu J)(i) - J(i))e,$$

[compare with Eq. (3.12)]. Thus in this case the iteration  $J := M(J)$  shifts the result  $T_\mu J$  of value iteration to a vector that lies somewhere in the middle of the error bound range given by Prop. 3.1. By the result of this proposition it follows that the iteration converges to  $J_\mu$ .

Generally, however, the iteration  $J := MJ$  need not converge in the case where the direction vector  $d$  is chosen arbitrarily. If on the other hand  $d$  is chosen to be an eigenvector of  $Q_\mu$ , convergence can be proved. This is shown in Exercise 1.8, where it is also proved that *if  $d$  is an eigenvector corresponding to the dominant eigenvalue of  $Q_\mu$  (the one with largest modulus), the convergence rate of the iteration  $J := MJ$  is governed by the subdominant eigenvalue of  $Q_\mu$  (the one with second largest modulus)*. One possibility for finding approximately such an eigenvector is to apply  $F$  a sufficiently large number of times to a vector  $J$ . In particular, suppose that the initial error  $J - J_\mu$  can be decomposed as

$$J - J_\mu = \sum_{j=1}^n \xi_j e_j$$

for some scalars  $\xi_1, \dots, \xi_n$ , where  $e_1, \dots, e_n$  are eigenvectors of  $Q_\mu$ , and  $\lambda_1, \dots, \lambda_n$  are corresponding eigenvalues. Suppose also that  $\lambda_1$  is the

unique dominant eigenvalue, that is,  $|\lambda_j| < |\lambda_1|$  for  $j = 2, \dots, n$ . Then the difference  $F^{k+1}J - F^k J$  is nearly equal to  $\xi_1(\lambda_1^{k+1} - \lambda_1^k)c_1$  for large  $k$  and can be used to estimate the dominant eigenvector  $c_1$ . In order to decide whether  $k$  has been chosen large enough, one can test to see if the angle between the successive differences  $F^{k+1}J - F^k J$  and  $F^k J - F^{k-1}J$  is very small; if this is so, the components of  $F^{k+1}J - F^k J$  along the eigenvectors  $c_2, \dots, c_n$  must also be very small. (For a more sophisticated version of this argument, see [Ber93], where the generic rank-one correction method is developed in more general form.)

We can thus consider a two-phase approach: in the first phase, we apply several times the regular iteration  $J := FJ$  both to improve our estimate of  $J$  and also to obtain an estimate  $d$  of an eigenvector corresponding to a dominant eigenvalue; in the second phase we use the modified iteration  $J := MJ$  that involves extrapolation along  $d$ . It can be shown that the two-phase method converges to  $J_p$ , provided the error in the estimation of  $d$  is small enough, that is, the cosine of the angle between  $d$  and  $Q_\mu d$  as measured by the ratio

$$\frac{(F^k J - F^{k-1}J)'(F^{k-1}J - F^{k-2}J)}{\|F^k J - F^{k-1}J\| \cdot \|F^{k-1}(J) - F^{k-2}J\|}$$

is sufficiently close to one.

Note that the computation of the first phase is not wasted since it uses the iteration  $J := FJ$  that we are trying to accelerate. Furthermore, since the second phase involves the calculation of  $FJ$  at the current iterate  $J$ , any error bounds or termination criteria based on  $FJ$  can be used to terminate the algorithm. As a result, the same finite termination mechanism can be used for both iterations  $J := FJ$  and  $J := MJ$ .

One difficulty of the correction method outlined above is that the appropriate vector  $d$  depends on  $Q_\mu$  and therefore also on  $\mu$ . In the case of optimization over several policies, the mapping  $F$  is defined by

$$(FJ)(i) = \min_{u \in U(i)} \left[ h_i(u) + \sum_{j=1}^n q_{ij}(u)J(j) \right], \quad i = 1, \dots, n. \quad (3.20)$$

One can then use the rank-one correction approach in two different ways:

- (1) Iteratively compute the cost vectors of the policies generated by a policy iteration scheme of the type discussed in the next subsection.
- (2) Guess at an optimal policy within the first phase, switch to the second phase, and then return to the first phase if the policy changes “substantially” during the second phase. In particular, in the first phase, the iteration  $J := FJ$  is used, where  $F$  is the nonlinear mapping of Eq. (3.20). Upon switching to the second phase, the vector  $z$  is taken

to be equal to  $Q_{\mu^*}d$ , where  $\mu^*$  is the policy that attains the minimum in Eq. (3.20) at the time of the switch. The second phase consists of the iteration

$$J := MJ = FJ + \tilde{\gamma}z,$$

where  $F$  is the nonlinear mapping of Eq. (3.20), and  $\tilde{\gamma}$  is again given by

$$\tilde{\gamma} = \frac{(d - z)'(FJ - J)}{\|d - z\|^2}.$$

To guard against subsequent changes in policy, which induce corresponding changes in the matrix  $Q_{\mu^*}$ , one should ensure that the method is working properly, for example, by recomputing  $d$  if the policy changes and/or the error  $\|FJ - J\|$  is not reduced at a satisfactory rate. This method is generally effective because the value iteration method typically finds an optimal policy much before it finds the optimal cost vector.

It should be mentioned, however, that the rank-one correction method is ineffective if there is little or no separation between the dominant and the subdominant eigenvalue moduli, both because the convergence rate of the method for obtaining  $d$  is slow, and also because the convergence rate of the modified iteration  $J := MJ$  is not much faster than the one of the regular iteration  $J := FJ$ . For such problems, one should try corrections over subspaces of dimension larger than one (see [Ber93], and the adaptive aggregation and multiple-rank correction methods given in Section 4.3.3).

### Infinite State Space – Approximate Value Iteration

The value iteration method is valid under the assumptions of Prop. 2.1, so it is guaranteed to converge to  $J^*$  for problems with infinite state and control spaces. However, for such problems, the method may be implementable only through approximations. In particular, given a function  $J$ , one may only be able to calculate a function  $\hat{J}$  such that

$$\max_{x \in S} |\hat{J}(x) - (TJ)(x)| \leq \epsilon, \quad (3.21)$$

where  $\epsilon$  is a given positive scalar. A similar situation may occur even when the state space is finite but the number of states is very large. Then instead of calculating  $(TJ)(x)$  for all states  $x$ , one may do so only for some states and estimate  $(TJ)(x)$  for the remaining states  $x$  by some form of interpolation, or by a least-squares error fit of  $(TJ)(x)$  with a function from a suitable parametric class (compare with the discussion of Section 2.3). Then the function  $\hat{J}$  thus obtained will satisfy a relation such as (3.21).

We are thus led to consider the approximate value iteration method that generates a sequence  $\{J_k\}$  satisfying

$$\max_{x \in S} |J_{k+1}(x) - (TJ_k)(x)| \leq \epsilon, \quad k = 0, 1, \dots \quad (3.22)$$

starting from an arbitrary bounded function  $J_0$ . Generally, such a sequence "converges" to  $J^*$  to within an error of  $\epsilon/(1-\alpha)$ . To see this, note that Eq. (3.22) yields

$$TJ_0 - \alpha\epsilon \leq J_1 \leq TJ_0 + \alpha\epsilon.$$

By applying  $T$  to this relation, we obtain

$$T^2J_0 - \alpha\epsilon\alpha \leq TJ_1 \leq T^2J_0 + \alpha\epsilon\alpha,$$

so by using Eq. (3.22) to write

$$TJ_1 - \alpha\epsilon \leq J_2 \leq TJ_1 + \alpha\epsilon,$$

we have

$$T^2J_0 - \epsilon(1+\alpha)\epsilon \leq J_2 \leq T^2J_0 + \epsilon(1+\alpha)\epsilon.$$

Proceeding similarly, we obtain for all  $k \geq 1$ ,

$$T^{k+1}J_0 - \epsilon(1+\alpha+\dots+\alpha^{k-1})\epsilon \leq J_k \leq T^{k+1}J_0 + \epsilon(1+\alpha+\dots+\alpha^{k-1})\epsilon.$$

By taking the limit superior and the limit inferior as  $k \rightarrow \infty$ , and by using the fact  $\lim_{k \rightarrow \infty} T^k J_0 = J^*$ , we see that

$$J^* - \frac{\epsilon}{1-\alpha}\epsilon \leq \liminf_{k \rightarrow \infty} J_k \leq \limsup_{k \rightarrow \infty} J_k \leq J^* + \frac{\epsilon}{1-\alpha}\epsilon.$$

It is also possible to obtain versions of the error bounds of Prop. 3.1 for the approximate value iteration method. We have from that proposition

$$\begin{aligned} TJ_k - \frac{\alpha}{1-\alpha} \min_{x \in S} [(TJ_k)(x) - J_k(x)]\epsilon &\leq J^* \\ &\leq TJ_k + \frac{\alpha}{1-\alpha} \max_{x \in S} [(TJ_k)(x) - J_k(x)]\epsilon. \end{aligned}$$

By using Eq. (3.22) in the above relation, we obtain

$$\begin{aligned} J_{k+1} - \epsilon\epsilon - \frac{\alpha}{1-\alpha} \min_{x \in S} [J_{k+1}(x) + \epsilon - J_k(x)]\epsilon &\leq J^* \\ &\leq J_{k+1} + \epsilon\epsilon + \frac{\alpha}{1-\alpha} \max_{x \in S} [J_{k+1}(x) + \epsilon - J_k(x)]\epsilon, \end{aligned}$$

or

$$\begin{aligned} J_{k+1} - \frac{\epsilon + \alpha \min_{x \in S} [J_{k+1}(x) - J_k(x)]}{1-\alpha}\epsilon &\leq J^* \\ &\leq J_{k+1} + \frac{\epsilon + \alpha \max_{x \in S} [J_{k+1}(x) - J_k(x)]}{1-\alpha}\epsilon. \end{aligned}$$

These bounds hold even when the state space is infinite because the bounds of Prop. 3.1 can be shown for an infinite state space as well. However, for these bounds to be useful, one should know  $\epsilon$ .

### 1.3.2 Policy Iteration

The policy iteration algorithm generates a sequence of stationary policies, each with improved cost over the preceding one. Given the stationary policy  $\mu$ , and the corresponding cost function  $J_\mu$ , an improved policy  $\{\bar{\mu}, \bar{\mu}_1, \dots\}$  is computed by minimization in the DP equation corresponding to  $J_\mu$ , that is,  $T_{\bar{\mu}}J_\mu = TJ_\mu$ , and the process is repeated.

The algorithm is based on the following proposition.

**Proposition 3.4:** Let  $\mu$  and  $\bar{\mu}$  be stationary policies such that  $T_{\bar{\mu}}J_\mu = TJ_\mu$ , or equivalently, for  $i = 1, \dots, n$ ,

$$g(i, \bar{\mu}(i)) + \alpha \sum_{j=1}^n p_{ij}(\bar{\mu}(i))J_\mu(j) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u)J_\mu(j) \right].$$

Then we have

$$J_{\bar{\mu}}(i) \leq J_\mu(i), \quad i = 1, \dots, n. \quad (3.23)$$

Furthermore, if  $\mu$  is not optimal, strict inequality holds in the above equation for at least one state  $i$ .

**Proof:** Since  $J_\mu = T_\mu J_\mu$  (Cor. 2.2.1) and, by hypothesis,  $T_{\bar{\mu}}J_\mu = TJ_\mu$ , we have for every  $i$ ,

$$\begin{aligned} J_\mu(i) &= g(i, \mu(i)) + \alpha \sum_{j=1}^n p_{ij}(\mu(i))J_\mu(j) \\ &\geq g(i, \bar{\mu}(i)) + \alpha \sum_{j=1}^n p_{ij}(\bar{\mu}(i))J_\mu(j) \\ &= (T_{\bar{\mu}}J_\mu)(i). \end{aligned}$$

Applying repeatedly  $T_{\bar{\mu}}$  on both sides of this inequality and using the monotonicity of  $T_{\bar{\mu}}$  (Lemma 1.1) and Cor. 2.1.1, we obtain

$$J_\mu \geq T_{\bar{\mu}}J_\mu \geq \dots \geq T_{\bar{\mu}}^k J_\mu \geq \dots \geq \lim_{N \rightarrow \infty} T_{\bar{\mu}}^N J_\mu = J_{\bar{\mu}},$$

proving Eq. (3.23).

If  $J_\mu = J_{\bar{\mu}}$ , then from the preceding relation it follows that  $J_\mu = T_{\bar{\mu}}J_\mu$  and since by hypothesis we have  $T_{\bar{\mu}}J_\mu = TJ_\mu$ , we obtain  $J_\mu = TJ_\mu$ , implying that  $J_\mu = J^*$  by Prop. 2.2. Thus  $\mu$  must be optimal. It follows that if  $\mu$  is not optimal, then  $J_{\bar{\mu}}(i) < J_\mu(i)$  for some state  $i$ . **Q.E.D.**

### Policy Iteration Algorithm

- Step 1: (Initialization)** Guess an initial stationary policy  $\mu^0$ .
- Step 2: (Policy Evaluation)** Given the stationary policy  $\mu^k$ , compute the corresponding cost function  $J_{\mu^k}$  from the linear system of equations
- $$(I - \alpha P_{\mu^k})J_{\mu^k} = g_{\mu^k}.$$
- Step 3: (Policy Improvement)** Obtain a new stationary policy  $\mu^{k+1}$  satisfying
- $$T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k}.$$
- If  $J_{\mu^k} = TJ_{\mu^k}$  stop; else return to Step 2 and repeat the process.

Since the collection of all stationary policies is finite (by the finiteness of  $S$  and  $C$ ) and an improved policy is generated at every iteration, it follows that the algorithm will find an optimal stationary policy in a finite number of iterations. This property is the main advantage of policy iteration over value iteration, which in general converges in an infinite number of iterations. On the other hand, finding the exact value of  $J_{\mu^k}$  in Step 2 of the algorithm requires solving the system of linear equations  $(I - \alpha P_{\mu^k})J_{\mu^k} = g_{\mu^k}$ . The dimension of this system is equal to the number of states, and thus when this number is very large, the method is not attractive.

Figure 1.3.4 provides a geometric interpretation of policy iteration and compares it with value iteration.

We note that in some cases, one can exploit the special structure of the problem at hand to accelerate policy iteration. For example, sometimes we can show that if  $\mu$  belongs to some restricted subset  $M$  of admissible control functions, then  $J_\mu$  has a form guaranteeing that  $\bar{\mu}$  will also belong to the subset  $M$ . In this case, policy iteration will be confined within the subset  $M$ , if the initial policy belongs to  $M$ . Furthermore, the policy evaluation step may be facilitated. For an example, see Exercise 1.44.

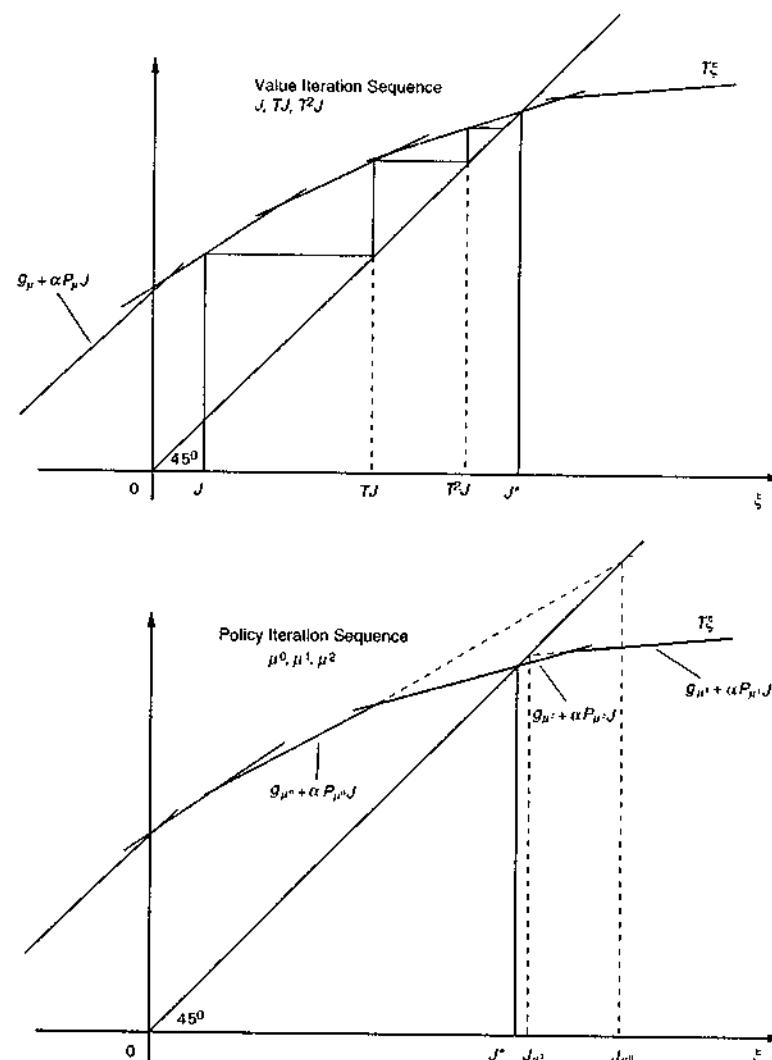
We now demonstrate policy iteration by means of the example considered earlier in this section.

### Example 3.1 (continued)

Let us go through the calculations of the policy iteration method:

**Initialization:** We select the initial stationary policy

$$\mu^0(1) = u^1, \quad \mu^0(2) = u^2.$$



**Figure 1.3.4** Geometric interpretation of policy iteration and value iteration. Each stationary policy  $\mu$  defines the linear function  $g_\mu + \alpha P_\mu J$  of the vector  $J$ , and  $TJ$  is the piecewise linear function  $\min_\mu [g_\mu + \alpha P_\mu J]$ . The optimal cost  $J^*$  satisfies  $J^* = TJ^*$ , so it is obtained from the intersection of the graph of  $TJ$  and the 45 degree line shown. The value iteration sequence is indicated in the top figure by the staircase construction, which asymptotically leads to  $J^*$ . The policy iteration sequence terminates when the correct linear segment of the graph of  $TJ$  (i.e., the optimal stationary policy) is identified, as shown in the bottom figure.

**Policy Evaluation:** We obtain  $J_{\mu^0}$  through the equation  $J_{\mu^0} = T_{\mu^0}J_{\mu^0}$ , or, equivalently, the linear system of equations

$$J_{\mu^0}(1) = g(1, u^1) + \alpha p_{11}(u^1)J_{\mu^0}(1) + \alpha p_{12}(u^1)J_{\mu^0}(2),$$

$$J_{\mu^0}(2) = g(2, u^2) + \alpha p_{21}(u^2)J_{\mu^0}(1) + \alpha p_{22}(u^2)J_{\mu^0}(2).$$

Substituting the data of the problem, we have

$$J_{\mu^0}(1) = 2 + 0.9 \cdot \frac{3}{4} \cdot J_{\mu^0}(1) + 0.9 \cdot \frac{1}{4} \cdot J_{\mu^0}(2),$$

$$J_{\mu^0}(2) = 3 + 0.9 \cdot \frac{1}{4} \cdot J_{\mu^0}(1) + 0.9 \cdot \frac{3}{4} \cdot J_{\mu^0}(2).$$

Solving this system of linear equations for  $J_{\mu^0}(1)$  and  $J_{\mu^0}(2)$ , we obtain

$$J_{\mu^0}(1) \approx 24.12, \quad J_{\mu^0}(2) \approx 25.96.$$

**Policy Improvement:** We now find  $\mu^1(1)$  and  $\mu^1(2)$  satisfying  $T_{\mu^1}J_{\mu^0} = TJ_{\mu^0}$ . We have

$$\begin{aligned} (TJ_{\mu^0})(1) &= \min \left\{ 2 + 0.9 \left( \frac{3}{4} \cdot 24.12 + \frac{1}{4} \cdot 25.96 \right), \right. \\ &\quad \left. 3 + 0.9 \left( \frac{1}{4} \cdot 24.12 + \frac{3}{4} \cdot 25.96 \right) \right\} \\ &= \min\{24.12, 23.45\} = 23.45, \end{aligned}$$

$$\begin{aligned} (TJ_{\mu^0})(1) &= \min \left\{ 1 + 0.9 \left( \frac{3}{4} \cdot 24.12 + \frac{1}{4} \cdot 25.96 \right), \right. \\ &\quad \left. 3 + 0.9 \left( \frac{1}{4} \cdot 24.12 + \frac{3}{4} \cdot 25.96 \right) \right\} \\ &= \min\{23.12, 25.95\} = 23.12. \end{aligned}$$

The minimizing controls are

$$\mu^1(1) = u^2, \quad \mu^1(2) = u^1.$$

**Policy Evaluation:** We obtain  $J_{\mu^1}$  through the equation  $J_{\mu^1} = T_{\mu^1}J_{\mu^1}$ :

$$J_{\mu^1}(1) = g(1, u^2) + \alpha p_{11}(u^2)J_{\mu^1}(1) + \alpha p_{12}(u^2)J_{\mu^1}(2),$$

$$J_{\mu^1}(2) = g(2, u^1) + \alpha p_{21}(u^1)J_{\mu^1}(1) + \alpha p_{22}(u^1)J_{\mu^1}(2).$$

Substitution of the data of the problem and solution of the system of equations yields

$$J_{\mu^1}(1) \approx 7.33, \quad J_{\mu^1}(2) \approx 7.67.$$

**Policy Improvement:** We perform the minimization required to find  $TJ_{\mu^1}$ :

$$\begin{aligned} (TJ_{\mu^1})(1) &= \min \left\{ 2 + 0.9 \left( \frac{3}{4} \cdot 7.33 + \frac{1}{4} \cdot 7.67 \right), \right. \\ &\quad \left. 3 + 0.9 \left( \frac{1}{4} \cdot 7.33 + \frac{3}{4} \cdot 7.67 \right) \right\} \\ &= \min\{8.67, 7.33\} = 7.33, \end{aligned}$$

$$\begin{aligned} (TJ_{\mu^1})(2) &= \min \left\{ 1 + 0.9 \left( \frac{3}{4} \cdot 7.33 + \frac{1}{4} \cdot 7.67 \right), \right. \\ &\quad \left. 3 + 0.9 \left( \frac{1}{4} \cdot 7.33 + \frac{3}{4} \cdot 7.67 \right) \right\} \\ &= \min\{7.67, 9.83\} = 7.67. \end{aligned}$$

Hence we have  $J_{\mu^1} = TJ_{\mu^1}$ , which implies that  $\mu^1$  is optimal and  $J_{\mu^1} = J^*$ :

$$\mu^*(1) = u^2, \quad \mu^*(2) = u^1, \quad J^*(1) \approx 7.33, \quad J^*(2) \approx 7.67.$$

### Modified Policy Iteration

When the number of states is large, solving the linear system  $(I - \alpha P_{\mu^k})J_{\mu^k} = g_{\mu^k}$  in the policy evaluation step by direct methods such as Gaussian elimination can be prohibitively time-consuming. One way to get around this difficulty is to solve the linear system iteratively by using value iteration. In fact, we may consider solving the system only approximately by executing a limited number of value iterations. This is called the *modified policy iteration algorithm*.

To formalize this method, let  $J_0$  be an arbitrary  $n$ -dimensional vector. Let  $m_0, m_1, \dots$  be positive integers, and let the vectors  $J_1, J_2, \dots$  and the stationary policies  $\mu_0, \mu_1, \dots$  be defined by

$$T_{\mu^k}J_k = TJ_k, \quad J_{k+1} = T_{\mu^{m_k}}^m J_k, \quad k = 0, 1, \dots$$

Thus, a stationary policy  $\mu^k$  is defined from  $J_k$  according to  $T_{\mu^k}J_k = TJ_k$ , and the cost  $J_{\mu^k}$  is approximately evaluated by  $m_k - 1$  additional value iterations, yielding the vector  $J_{k+1}$ , which is used in turn to define  $\mu^{k+1}$ . We have the following:

**Proposition 3.5:** Let  $\{J_k\}$  and  $\{\mu_k\}$  be the sequences generated by the modified policy iteration algorithm. Then  $\{J_k\}$  converges to  $J^*$ . Furthermore, there exists an integer  $\bar{k}$  such that for all  $k \geq \bar{k}$ ,  $\mu^k$  is optimal.

**Proof:** Let  $r$  be a scalar such that the vector  $\bar{J}_0$ , defined by  $\bar{J}_0 = J_0 + rc$ , satisfies  $T\bar{J}_0 \leq \bar{J}_0$ . [Any scalar  $r$  such that  $\max_i[(TJ_0)(i) - J_0(i)] \leq (1-\alpha)r$  has this property.] Define for all  $k$ ,  $\bar{J}_{k+1} = T_{\mu^k}\bar{J}_k$ . Then, it can be seen by induction that for all  $k$  and  $m = 0, 1, \dots, m_k$ , the vectors  $T_{\mu^k}^m J_k$  and  $T_{\mu^k}^m \bar{J}_k$  differ by the multiple of the unit vector  $r\alpha^{m_0+m_1+\dots+m_{k-1}+m} e$ . It follows that if  $J_0$  is replaced by  $\bar{J}_0$  as the starting vector in the algorithm, the same sequence of policies  $\{\mu_k\}$  will be obtained; that is, we have for all  $k$

$$T_{\mu^k} \bar{J}_k = T \bar{J}_k.$$

Now we will show that for all  $k$  we have  $\bar{J}_k \leq T^k \bar{J}_0$ . Indeed, we have  $T_{\mu^0} \bar{J}_0 = T \bar{J}_0 \leq \bar{J}_0$ , from which we obtain

$$T_{\mu^0}^m \bar{J}_0 \leq T_{\mu^0}^{m-1} \bar{J}_0, \quad m = 1, 2, \dots$$

so that

$$T_{\mu^1} \bar{J}_1 = T \bar{J}_1 \leq T_{\mu^0} \bar{J}_1 = T_{\mu^0}^{m_0+1} \bar{J}_0 \leq T_{\mu^0}^{m_0} \bar{J}_0 = \bar{J}_1 \leq T_{\mu^0} \bar{J}_0 = T \bar{J}_0.$$

This argument can be continued to show that for all  $k$ , we have  $\bar{J}_k \leq T^k \bar{J}_0$ , so that

$$\bar{J}_k \leq T^k \bar{J}_0, \quad k = 0, 1, \dots$$

On the other hand, since  $T\bar{J}_0 \leq \bar{J}_0$ , we have  $J^* \leq \bar{J}_0$ , and it follows that application of any number of mappings of the form  $T_\mu$  to  $\bar{J}_0$  produces functions that are bounded from below by  $J^*$ . Thus,

$$J^* \leq \bar{J}_k \leq T^k \bar{J}_0, \quad k = 0, 1, \dots$$

By taking the limit as  $k \rightarrow \infty$ , we obtain  $\lim_{k \rightarrow \infty} \bar{J}_k(i) = J^*(i)$  for all  $i$ , and since  $\lim_{k \rightarrow \infty} (J_k - J_k^*) = 0$ , we obtain

$$\lim_{k \rightarrow \infty} J_k(i) = J^*(i), \quad i = 1, \dots, n.$$

Since the number of stationary policies is finite, there exists an  $\bar{\epsilon} > 0$  such that if a stationary policy  $\mu$  satisfies

$$\max_i [J_\mu(i) - J^*(i)] < \bar{\epsilon},$$

then  $\mu$  is optimal. Now let  $\bar{k}$  be such that for all  $k \geq \bar{k}$  we have

$$\frac{\alpha}{1-\alpha} \left( \max_i [(TJ_k)(i) - J_k(i)] - \min_i [(TJ_k)(i) - J_k(i)] \right) < \bar{\epsilon}.$$

Then from Eq. (3.10) we see that for all  $k \geq \bar{k}$ , the stationary policy  $\mu^k$  that satisfies  $T_{\mu^k} J_k = TJ_k$  is optimal. Q.E.D.

Note that if  $m_k = 1$  for all  $k$  in the modified policy iteration algorithm, we obtain the value iteration method, while if  $m_k = \infty$  we obtain the policy iteration method, where the policy evaluation step is performed iteratively by means of value iteration. Analysis and computational experience suggest that it is usually best to take  $m_k$  larger than 1 according to some heuristic scheme. A key idea here is that a value iteration involving a single policy (evaluating  $T_\mu J$  for some  $\mu$  and  $J$ ) is much less expensive than an iteration involving all policies (evaluating  $TJ$  for some  $J$ ), when the number of controls available at each state is large. Note that error bounds such as the ones of Prop. 3.1 can be used to improve the approximation process. Furthermore, Gauss-Seidel iterations can be used in place of the usual value iterations.

### Infinite State Space – Approximate Policy Iteration

The policy iteration method can be defined for problems with infinite state and control spaces by means of the relation

$$T_{\mu^{k+1}} J_{\mu^k} = TJ_{\mu^k}, \quad k = 0, 1, \dots$$

The proof of Prop. 3.4 can then be used to show that the generated sequence of policies  $\{\mu^k\}$  is improving in the sense that  $J_{\mu^{k+1}} \leq J_{\mu^k}$  for all  $k$ . However, for infinite state space problems, the policy evaluation step and/or the policy improvement step of the method may be implementable only through approximations. A similar situation may occur even when the state space is finite but the number of states is very large.

We are thus led to consider an approximate policy iteration method that generates a sequence of stationary policies  $\{\mu^k\}$  and a corresponding sequence of approximate cost functions  $\{J_k\}$  satisfying

$$\max_{x \in S} |J_k(x) - J_{\mu^k}(x)| \leq \delta, \quad k = 0, 1, \dots \quad (3.24)$$

and

$$\max_{x \in S} |(T_{\mu^{k+1}} J_k)(x) - (TJ_k)(x)| \leq \epsilon, \quad k = 0, 1, \dots \quad (3.25)$$

where  $\delta$  and  $\epsilon$  are some positive scalars, and  $\mu^0$  is an arbitrary stationary policy. We call this the *approximate policy iteration algorithm*. The following proposition provides error bounds for this algorithm.

**Proposition 3.6:** The sequence  $\{\mu^k\}$  generated by the approximate policy iteration algorithm satisfies

$$\limsup_{k \rightarrow \infty} \max_{x \in S} (J_{\mu^k}(x) - J^*(x)) \leq \frac{\epsilon + 2\alpha\delta}{(1-\alpha)^2}. \quad (3.26)$$

**Proof:** From Eqs. (3.24) and (3.25), we have for all  $k$

$$T_{\mu^{k+1}} J_{\mu^k} - \alpha\delta c \leq T_{\mu^{k+1}} J_k \leq TJ_k + \epsilon c,$$

where  $c = (1, 1, \dots, 1)'$  is the unit vector, while from Eq. (3.24), we have for all  $k$

$$TJ_k \leq TJ_{\mu^k} + \alpha\delta c.$$

By combining these two relations, we obtain for all  $k$

$$T_{\mu^{k+1}} J_{\mu^k} \leq TJ_{\mu^k} + (\epsilon + 2\alpha\delta)c \leq T_{\mu^k} J_{\mu^k} + (\epsilon + 2\alpha\delta)c. \quad (3.27)$$

From Eq. (3.27) and the equation  $T_{\mu^k} J_{\mu^k} = J_{\mu^k}$ , we have

$$T_{\mu^{k+1}} J_{\mu^k} \leq J_{\mu^k} + (\epsilon + 2\alpha\delta)c.$$

By subtracting from this relation the equation  $T_{\mu^{k+1}} J_{\mu^{k+1}} = J_{\mu^{k+1}}$ , we obtain

$$T_{\mu^{k+1}} J_{\mu^k} - T_{\mu^{k+1}} J_{\mu^{k+1}} \leq J_{\mu^k} - J_{\mu^{k+1}} + (\epsilon + 2\alpha\delta)c,$$

which can be written as

$$J_{\mu^{k+1}} - J_{\mu^k} \leq \alpha F_k + (\epsilon + 2\alpha\delta)c, \quad (3.28)$$

where  $F_k$  is the function given by

$$\begin{aligned} F_k(x) &= \alpha^{-1}(T_{\mu^{k+1}} J_{\mu^{k+1}})(x) - \alpha^{-1}(T_{\mu^{k+1}} J_{\mu^k})(x) \\ &= E_w \{ J_{\mu^{k+1}}(f(x, \mu^{k+1}(x), w)) - J_{\mu^k}(f(x, \mu^{k+1}(x), w)) \}. \end{aligned}$$

Let

$$\xi_k = \max_{x \in S} (J_{\mu^{k+1}}(x) - J_{\mu^k}(x)).$$

Then we have  $F_k(x) \leq \xi_k$  for all  $x \in S$ , and Eq. (3.28) yields

$$\xi_k \leq \alpha\xi_k + \epsilon + 2\alpha\delta,$$

or

$$\xi_k \leq \frac{\epsilon + 2\alpha\delta}{1-\alpha}. \quad (3.29)$$

Let

$$\zeta_k = \max_{x \in S} (J_{\mu^k}(x) - J^*(x)).$$

From Eq. (3.27) and the relation

$$\max_{x \in S} ((TJ_{\mu^k})(x) - J^*(x)) \leq \alpha\zeta_k,$$

which follows from Prop. 2.4, we have

$$T_{\mu^{k+1}} J_{\mu^k} \leq TJ_{\mu^k} + (\epsilon + 2\alpha\delta)c \leq J^* + \alpha\zeta_k + (\epsilon + 2\alpha\delta)c.$$

We also have

$$T_{\mu^{k+1}} J_{\mu^k} = J_{\mu^{k+1}} + T_{\mu^{k+1}} J_{\mu^k} - T_{\mu^{k+1}} J_{\mu^{k+1}},$$

and by subtracting the last two relations, we obtain

$$J_{\mu^{k+1}} - J^* \leq \alpha\zeta_k + \alpha F_k + (\epsilon + 2\alpha\delta)c.$$

From this relation we see that

$$\zeta_{k+1} \leq \alpha\zeta_k + \alpha\xi_k + \epsilon + 2\alpha\delta.$$

By taking the limit superior as  $k \rightarrow \infty$  and by using Eq. (3.29), we obtain

$$(1-\alpha) \limsup_{k \rightarrow \infty} \zeta_k \leq \alpha \frac{\epsilon + 2\alpha\delta}{1-\alpha} + \epsilon + 2\alpha\delta.$$

This relation simplifies to

$$\limsup_{k \rightarrow \infty} \zeta_k \leq \frac{\epsilon + 2\alpha\delta}{(1-\alpha)^2},$$

which was to be proved. **Q.E.D.**

Proposition 3.6 suggests that the approximate policy iteration method makes steady progress up to a point and then the iterates  $J_{\mu^k}$  oscillate within a neighborhood of the optimum  $J^*$ . This behavior appears to be typical in practice. Note that for  $\delta = 0$  and  $\epsilon = 0$ , Prop. 3.6 shows that the cost sequence  $\{J_{\mu^k}\}$  generated by the (exact) policy iteration algorithm converges to  $J^*$ , even when the state space is infinite.

### 1.3.3 Adaptive Aggregation

Let us now consider an alternative to value iteration for performing approximate evaluation of a stationary policy  $\mu$ , that is, for solving approximately the system

$$J_\mu = T_\mu J_\mu.$$

This alternative is recommended for problems where convergence of value iteration, even with error bounds, is very slow. The idea here is to solve instead of the system  $J_\mu = T_\mu J_\mu$ , another system of smaller dimension, which is obtained by lumping together the states of the original system into subsets  $S_1, S_2, \dots, S_m$  that can be viewed as *aggregate states*. These subsets are disjoint and cover the entire state space, that is,

$$S = S_1 \cup S_2 \cup \dots \cup S_m.$$

Consider the  $n \times m$  matrix  $W$  whose  $i$ th column has unit entries at coordinates corresponding to states in  $S_i$  and all other entries equal to zero. Consider also an  $m \times n$  matrix  $Q$  such that the  $i$ th row of  $Q$  is a probability distribution  $(q_{is} | s \in S_i)$  with  $q_{is} = 0$  if  $s \notin S_i$ . The structure of  $Q$  implies two useful properties:

- (a)  $QW = I$ .
- (b) The matrix

$$R = QP_\mu W$$

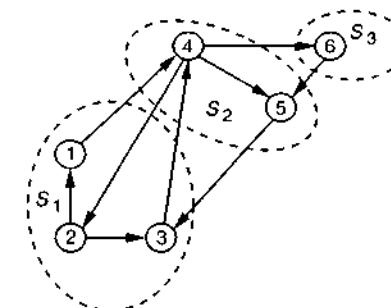
is an  $m \times m$  transition probability matrix. In particular, the  $ij$ th component of  $R$  is equal to

$$r_{ij} = \sum_{s \in S_i} q_{is} \sum_{t \in S_j} p_{st}(\mu(s)),$$

and gives the probability that the next state will belong to aggregate state  $S_j$  given that the current state is drawn from the aggregate state  $S_i$  according to the probability distribution  $\{q_{is} | s \in S_i\}$ . The transition probability matrix  $R$  defines a Markov chain, called the *aggregate Markov chain*, whose states are the  $m$  aggregate states. Figure 1.3.5 illustrates the aggregate Markov chain.

Aggregate Markov chains are most useful when their transition behavior captures the broad attributes of the behavior of the original chain. This is generally true if the states of each aggregate state are “similar” in some sense. Let us describe one such situation. In particular, suppose that we have an estimate  $J$  of  $J_\mu$  and that *we postulate that over the states  $s$  of every aggregate state  $S_i$  the variation  $J_\mu(s) - J(s)$  is constant*. This amounts to hypothesizing that for some  $m$ -dimensional vector  $y$  we have

$$J_\mu - J = Wy.$$



**Figure 1.3.5** Illustration of the aggregate Markov chain. In this example, the aggregate states are  $S_1 = \{1, 2, 3\}$ ,  $S_2 = \{4, 5\}$ , and  $S_3 = \{6\}$ . The matrix  $W$  has columns  $(1, 1, 1, 0, 0, 0)', (0, 0, 0, 1, 1, 0)',$  and  $(0, 0, 0, 0, 0, 1)'$ . In this example, the matrix  $Q$  is chosen so that each of its rows defines a uniform probability distribution over the states of the corresponding aggregate state. Thus the rows of  $Q$  are  $(1/3, 1/3, 1/3, 0, 0, 0)$ ,  $(0, 0, 0, 1/2, 1/2, 0)$ , and  $(0, 0, 0, 0, 0, 1)$ . The aggregate Markov chain has transition probabilities  $r_{11} = \frac{1}{3}(p_{21} + p_{31})$ ,  $r_{12} = \frac{1}{3}(p_{11} + p_{31})$ ,  $r_{13} = 0$ ,  $r_{21} = \frac{1}{2}(p_{41} + p_{51})$ ,  $r_{22} = \frac{1}{2}p_{45}$ ,  $r_{23} = \frac{1}{2}p_{46}$ ,  $r_{31} = 0$ ,  $r_{32} = p_{56}$ , and  $r_{33} = 0$ .

By combining the equations  $T_\mu J = g_\mu + \alpha P_\mu J$  and  $g_\mu = (I - \alpha P_\mu)J_\mu$ , we have

$$(I - \alpha P_\mu)(J_\mu - J) = T_\mu J - J.$$

This is the variational form of the equation  $J_\mu = T_\mu J_\mu$  discussed earlier in connection with error bounds in Section 1.3.1, and can be used equally well for evaluating  $J_\mu$ . Let us multiply both sides with  $Q$  and use the equation  $J_\mu - J = Wy$ . We obtain

$$Q(I - \alpha P_\mu)Wy = Q(T_\mu J - J),$$

which, by using the equations  $QW = I$  and  $R = QP_\mu W$ , is written as

$$(I - \alpha R)y = Q(T_\mu J - J).$$

This equation can be solved for  $y$ , since  $R$  is a transition probability matrix and therefore the matrix  $I - \alpha R$  is invertible. Also, by applying  $T_\mu$  to both sides of the equation  $J_\mu = J + Wy$ , we obtain

$$J_\mu = T_\mu J_\mu = T_\mu J + \alpha P_\mu Wy.$$

We thus conclude that, if the variation of  $J_\mu(s) - J(s)$  is roughly constant over the states  $s$  of each aggregate state, then the vector  $T_\mu J + \alpha P_\mu Wy$  is a good approximation for  $J_\mu$ . Starting with  $J$ , this approximation is obtained as follows.

### Aggregation Iteration

**Step 1:** Compute  $T_\mu J$ .

**Step 2:** Delineate the aggregate states (i.e., define  $W$ ) and specify the matrix  $Q$ .

**Step 3:** Solve for  $y$  the system

$$(I - \alpha R)y = Q(T_\mu J - J), \quad (3.30)$$

where  $R = QP_\mu W$ , and approximate  $J_\mu$  using

$$J := T_\mu J + \alpha P_\mu W y. \quad (3.31)$$

Note that the aggregation iteration (3.31) can be equivalently written as

$$J := T_\mu(J + Wy),$$

so it differs from a value iteration in that it operates with  $T_\mu$  on  $J + Wy$  rather than  $J$ .

Solving the system (3.30) in the aggregation iteration has an interesting interpretation. It can be seen that  $y$  is the  $\alpha$ -discounted cost vector corresponding to the transition probability matrix  $R$  and the cost-per-stage vector  $Q(T_\mu J - J)$ . Thus, calculating  $y$  can be viewed as a policy evaluation step for the aggregate Markov chain when the cost per stage for each aggregate state  $S_i$  is equal to

$$\sum_{s \in S_i} q_{is} ((T_\mu J)(s) - J(s)),$$

which is the average  $T_\mu J - J$  over the aggregate state  $S_i$  according to the distribution  $\{q_{is} \mid s \in S_i\}$ . A key attractive aspect of the aggregation iteration is that the dimension of the system (3.30) is  $m$  (the number of aggregate states), which can be much smaller than  $n$  (the dimension of the system  $J_\mu = T_\mu J_\mu$  arising in the policy evaluation step of policy iteration).

### Delineating the Aggregate States

A key issue is how to identify the aggregate states  $S_1, \dots, S_m$  in a way that the error  $J_\mu - J$  is of similar magnitude in each one. One way to do this is to view  $T_\mu J$  as an approximation to  $J_\mu$  and to group together states  $i$  with comparable magnitudes of  $(T_\mu J)(i) - J(i)$ . Thus the interval  $[c, \bar{c}]$ , where

$$c = \min_i [(T_\mu J)(i) - J(i)], \quad \bar{c} = \max_i [(T_\mu J)(i) - J(i)],$$

is divided into  $m$  segments and membership of a state  $i$  in an aggregate state is determined by the segment within which  $(T_\mu J)(i) - J(i)$  lies. By this we mean that for each state  $i$ , we set  $i \in S_k$  if  $(T_\mu J)(i) - J(i) = c + (k - 1)\delta \in (0, \delta]$ , and we set

$$i \in S_k \quad \text{if} \quad (T_\mu J)(i) - J(i) = c + (k - 1)\delta \in (0, \delta],$$

where

$$\delta = \frac{\bar{c} - c}{m}.$$

This choice is based on the conjecture that, at least near convergence,  $(T_\mu J)(i) - J(i)$  will be of comparable magnitude for states  $i$  for which  $J_\mu(i) - J(i)$  is of comparable magnitude. Analysis and experimentation given in [BeC89] has shown that the preceding scheme often works well with a small number of aggregate states  $m$  (say 3 to 6), although the properties of the method are yet fully understood.

Note that the aggregate states can change from one iteration to the next, so the aggregation scheme "adapts" to the progress of the computation. The criterion used to delineate the aggregate states does not exploit any special problem structure. In some cases, however, it is possible to take advantage of existing special structure and modify accordingly the method used to form the aggregate states.

### Adaptive Aggregation Methods

It is possible to construct a number of methods that calculate  $J_\mu$  by using aggregation iterations. One possibility is simply to perform a sequence of aggregation iterations using the preceding method to partition the state space into a few, say 3 to 10, aggregate states. This method can be greatly improved by interleaving each aggregation iteration with multiple value iterations (applications of the mapping  $T_\mu$  on the current iterate). This is recommended based on experimentation and analysis given in [BeC89], to which we refer for further discussion. An interesting empirically observed phenomenon is that the error between the iterate and  $J_\mu$  is often increased by an aggregation iteration, but then unusually large improvements are made during the next few value iterations. This suggests that the number of value iterations following an aggregation iteration should be based on algorithmic progress; that is, an aggregation iteration should be performed when the progress of the value iterations becomes relatively small. Some experimentation may be needed with a given problem to determine an appropriate criterion for switching from the value iterations to an aggregation iteration.

There is no proof of convergence of the scheme just described. On the basis of computational experimentation, it appears reliable in practice. Its convergence nonetheless can be guaranteed by introducing a feature that

enforces some irreversible progress via the value iteration method following an aggregation iteration. In particular, one may calculate the error bounds of Prop. 3.1 at the value iteration Step 1, and impose a requirement that the subsequent aggregation iteration is skipped if these error bounds do not improve by a certain factor over the bounds computed prior to the preceding aggregation iteration.

To illustrate the effectiveness of the adaptive aggregation method, consider the three deterministic problems described earlier (cf. Fig. 1.3.3), and the performance of the method with two, three, and four aggregate states, starting from the zero function. The results, given in Fig. 1.3.6, should be compared with those of Fig. 1.3.3.

It is intuitively clear that the performance of the aggregation method should improve as the number of aggregate states increases, and indeed the computational results bear this out. The two extreme cases where  $m = n$  and  $m = 1$  are of interest. When  $m = n$ , each aggregate state has a single state and we obtain the policy iteration algorithm. When  $m = 1$ , there is only one aggregate state,  $W$  is equal to the unit vector  $c = (1, \dots, 1)'$ , and a straightforward calculation shows that for the choice  $Q = (1/n, \dots, 1/n)$ , the solution of the aggregate system (3.30) is

$$y = \frac{1}{n(1-\alpha)} \sum_{i=1}^n ((T_\mu J)(i) - J(i)).$$

From this equation (using also the fact  $P_\mu c = c$ ), we obtain the iteration

$$J := T_\mu J + \frac{\alpha}{n(1-\alpha)} \sum_{i=1}^n ((T_\mu J)(i) - J(i))c, \quad (3.32)$$

which is the same as the rank-one correction formula (3.12) obtained in Section 1.3.1 and amounts to shifting the result  $T_\mu J$  of value iteration within the error bound range given by Prop. 3.1. Thus we may view the aggregation scheme as a continuum of algorithms with policy iteration and value iteration (coupled with the error bounds of Prop. 3.1) included as the two extreme special cases.

#### Adaptive Multiple-Rank Corrections

One may observe that the aggregation iteration

$$J := T_\mu(J + Wy),$$

amounts to applying  $T_\mu$  to a correction of  $J$  along the subspace spanned by the columns of  $W$ . Once the matrix  $W$  is computed based on the adaptive procedure discussed above, we may consider choosing the vector  $y$  in alternative ways. An interesting possibility, which leads to a generalization

No. of aggregate states	Pr. 1 $\alpha = .9$	Pr. 1 $\alpha = .99$	Pr. 2 $\alpha = .9$	Pr. 2 $\alpha = .99$	Pr. 3 $\alpha = .9$	Pr. 3 $\alpha = .99$
2	14	13	9	9	83	505
3	1	1	3	3	64	367
4			3	3	26	351

**Figure 1.3.6** Number of iterations of adaptive aggregation methods with two, three, and four aggregate states to solve the problems of Fig. 1.3.3. Each row of  $Q$  was chosen to define a uniform probability distribution over the states of the corresponding aggregate state.

of the rank-one correction method of the preceding subsection, is to select  $y$  so that

$$\|J + Wy - T_\mu(J + Wy)\|^2 \quad (3.33)$$

is minimized. By setting to zero the gradient with respect to  $y$  of the above expression, we can verify that the optimal vector is given by

$$\hat{y} = (Z'Z)^{-1} Z'(T_\mu J - J),$$

where  $Z = (I - \alpha P_\mu)W$ . The corresponding iteration then becomes

$$J := T_\mu(J + W\hat{y}) = T_\mu J + \alpha P_\mu W\hat{y}.$$

Much of our discussion regarding the rank-one correction method also applies to this generalized version. In particular, we can use a two-phase implementation, which allows a return from phase two to phase one whenever the progress of phase two is unsatisfactory. Furthermore, a version of the method that works in the case of multiple policies is possible.

#### 1.3.4 Linear Programming

Since  $\lim_{N \rightarrow \infty} T^N J = J^*$  for all  $J$  (cf. Prop. 2.1), we have

$$J \leq TJ \quad \Rightarrow \quad J \leq J^* = TJ^*.$$

Thus  $J^*$  is the “largest”  $J$  that satisfies the constraint  $J \leq TJ$ . This constraint can be written as a finite system of linear inequalities

$$J(i) \leq g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u)J(j), \quad i = 1, \dots, n, \quad u \in U(i),$$

and delineates a polyhedron in  $\mathbb{R}^n$ . The optimal cost vector  $J^*$  is the "northeast" corner of this polyhedron, as illustrated in Fig. 1.3.7. In particular,  $J^*(1), \dots, J^*(n)$  solve the following problem (in  $\lambda_1, \dots, \lambda_n$ ):

$$\begin{aligned} & \text{maximize} \quad \sum_{i \in S} \lambda_i \\ & \text{subject to} \quad \lambda_i \leq g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) \lambda_j, \quad i = 1, \dots, n, \quad u \in U(i), \end{aligned}$$

where  $\tilde{S}$  is any nonempty subset of the state space  $S = \{1, \dots, n\}$ . This is a linear program with  $n$  variables and as many as  $n \times q$  constraints, where  $q$  is the maximum number of elements of the sets  $U(i)$ . As  $n$  increases, its solution becomes more complex. For very large  $n$  and  $q$ , the linear programming approach can be practical only with the use of special large-scale linear programming methods.

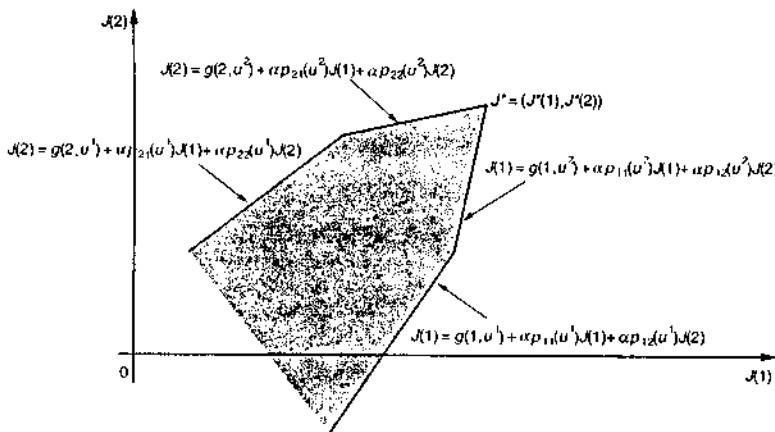


Figure 1.3.7 Linear programming problem associated with the discounted infinite horizon problem. The constraint set is shaded and the objective to maximize is  $J(1) + J(2)$ .

### Example 3.1 (continued)

For the example considered earlier in this section, the linear programming problem takes the form

$$\text{maximize } \lambda_1 + \lambda_2$$

$$\begin{aligned} & \text{subject to} \quad \lambda_1 \leq 2 + 0.9 \left( \frac{3}{4} \lambda_1 + \frac{1}{4} \lambda_2 \right), \quad \lambda_1 \leq 0.5 + 0.9 \left( \frac{1}{4} \lambda_1 + \frac{3}{4} \lambda_2 \right), \\ & \quad \lambda_2 \leq 1 + 0.9 \left( \frac{3}{4} \lambda_1 + \frac{1}{4} \lambda_2 \right), \quad \lambda_2 \leq 3 + 0.9 \left( \frac{1}{4} \lambda_1 + \frac{3}{4} \lambda_2 \right). \end{aligned}$$

### Cost Approximation Based on Linear Programming

When the number of states is very large or infinite, we may consider finding an approximation to the optimal cost function, which can be used in turn to obtain a (suboptimal) policy by minimization in Bellman's equation. One possibility is to approximate  $J^*(x)$  with the *linear* form

$$\hat{J}(x, r) = \sum_{k=1}^m r_k w_k(x), \quad (3.34)$$

where  $r = (r_1, \dots, r_m)$  is a vector of parameters, and for each state  $x$ ,  $w_k(x)$  are some fixed and known scalars. This amounts to approximating the cost function  $J^*(x)$  by a linear combination of  $m$  given functions  $w_k(x)$ , where  $k = 1, \dots, m$ . These functions play the role of a *basis* for the space of cost function approximations  $\hat{J}(x, r)$  that can be generated with different choices of  $r$  (see also the discussion of approximations in Section 2.3.3).

It is then possible to determine  $r$  by using  $\hat{J}(x, r)$  in place of  $J^*$  in the preceding linear programming approach. In particular, we compute  $r$  as the solution of the program

$$\begin{aligned} & \text{maximize} \quad \sum_{x \in \tilde{S}} \hat{J}(x, r) \\ & \text{subject to} \quad \hat{J}(x, r) \leq g(x, u) + \alpha \sum_{y \in S} p_{xy}(u) \hat{J}(y, r), \quad x \in \tilde{S}, \quad u \in \hat{U}(x), \end{aligned}$$

where  $\tilde{S}$  is either the state space  $S$  or a suitably chosen finite subset of  $S$ , and  $\hat{U}(x)$  is either  $U(x)$  or a suitably chosen finite subset of  $U(x)$ . Because  $\hat{J}(x, r)$  is linear in the parameter vector  $r$ , the above program is linear in the parameters  $r_1, \dots, r_m$ . Thus if  $m$  is small, the number of variables of the linear program is small. The number of constraints is as large as  $s \cdot q$ , where  $s$  is the number of elements of  $\tilde{S}$  and  $q$  is the maximum number of elements of the sets  $\hat{U}(x)$ . However, linear programs with a small number of variables and a large number of constraints can often be solved relatively quickly with the use of special large-scale linear programming methods known as cutting plane or column generation methods (see e.g. [Dan63], [Ber95a]). Thus, the preceding linear programming approach may be practical even for problems with a very large number of states.

## Approximate Policy Evaluation Using Linear Programming

In the case of a very large or infinite state space, it is also possible to use linear programming to evaluate approximately the cost function  $J_\mu$  of a stationary policy  $\mu$  in the context of the approximate policy iteration scheme discussed in Section 1.3.2. Suppose that we wish to approximate  $J_\mu$  by a function  $\tilde{J}(\cdot, r)$  of a given form, which is parameterized by the vector  $r = (r_1, \dots, r_m)$ . The bound of Prop. 3.6 suggests that we should try to determine the parameter vector  $r$  so as to minimize

$$\max_{x \in S} |\tilde{J}(x, r) - J_\mu(x)|.$$

From the error bounds given just prior to Prop. 3.1, it can also be seen that we have

$$\max_{x \in S} |\tilde{J}(x, r) - J_\mu(x)| \leq \frac{1}{1-\alpha} \max_{x \in S} |\tilde{J}(x, r) - (T_\mu \tilde{J})(x, r)|.$$

This motivates choosing  $r$  by solving the problem

$$\min_r \max_{x \in S} |\tilde{J}(x, r) - (T_\mu \tilde{J})(x, r)|,$$

where  $\hat{S}$  is either the state space  $S$  or a suitably chosen finite subset of  $S$ . The preceding problem is equivalent to

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \left| \tilde{J}(x, r) - g(x, \mu(x)) - \alpha \sum_{y \in S} p_{xy}(\mu(x)) \tilde{J}(y, r) \right| \leq z, \quad x \in \hat{S}. \end{aligned}$$

When  $\tilde{J}(x, r)$  has the linear form (3.34), this is a linear program in the variables  $z$  and  $r_1, \dots, r_m$ .

## 1.4 THE ROLE OF CONTRACTION MAPPINGS

Two key structural properties in DP models are responsible for most of the mathematical results one can prove about them. The first is the *monotonicity property* of the mappings  $T$  and  $T_\mu$  (cf. Lemma 1.1 in Section 1.1). This property is fundamental for total cost infinite horizon problems. For example, it forms the basis for the results on positive and negative DP models to be shown in Chapter 3.

When the cost per stage is bounded and there is discounting, however, we have another property that strengthens the effects of monotonicity: the mappings  $T$  and  $T_\mu$  are *contraction mappings*. In this section, we explain the meaning and implications of this property. The material in this section is conceptually very important, since contraction mappings are present in several additional DP models. However, the main result of this section (Prop. 4.1) will not be used explicitly in any of the proofs given later in this book.

Let  $B(S)$  denote the set of all bounded real-valued functions on  $S$ . With every function  $J : S \mapsto \mathbb{R}$  that belongs to  $B(S)$ , we associate the scalar

$$\|J\| = \max_{x \in S} |J(x)|. \quad (4.1)$$

[As an aid for the advanced reader, we mention that the function  $\|\cdot\|$  may be shown to be a norm on the linear space  $B(S)$ , and with this norm  $B(S)$  becomes a complete normed linear space [Lue69].] The following definition and proposition are specializations to  $B(S)$  of a more general notion and result that apply to such a space (see, e.g., references [LiS61] and [Lue69]).

**Definition 4.1:** A mapping  $H : B(S) \mapsto B(S)$  is said to be a *contraction mapping* if there exists a scalar  $\rho < 1$  such that

$$\|HJ - HJ'\| \leq \rho \|J - J'\|, \quad \text{for all } J, J' \in B(S),$$

where  $\|\cdot\|$  is the norm of Eq. (4.1). It is said to be an *m-stage contraction mapping* if there exists a positive integer  $m$  and some  $\rho < 1$  such that

$$\|H^m J - H^m J'\| \leq \rho \|J - J'\|, \quad \text{for all } J, J' \in B(S), \quad (4.2)$$

where  $H^m$  denotes the composition of  $H$  with itself  $m$  times.

The main result concerning contraction mappings is the following. For a proof, see references [LiS61] and [Lue69].

**Proposition 4.1: (Contraction Mapping Fixed-Point Theorem)** If  $H : B(S) \mapsto B(S)$  is a contraction mapping or an  $m$ -stage contraction mapping, then there exists a unique fixed point of  $H$ ; that is, there exists a unique function  $J^* \in B(S)$  such that

$$HJ^* = J^*.$$

Furthermore, if  $J$  is any function in  $B(S)$  and  $H^k$  is the composition of  $H$  with itself  $k$  times, then

$$\lim_{k \rightarrow \infty} \|H^k J - J^*\| = 0.$$

Now consider the mappings  $T$  and  $T_\mu$  defined by Eqs. (1.4) and (1.5). Proposition 2.1 and Cor. 2.4.1 show that  $T$  and  $T_\mu$  are contraction mappings ( $\rho = \alpha$ ). As a result, the convergence of the value iteration method to the unique fixed point of  $T$  follows directly from the contraction mapping theorem. Note also that, by Prop. 3.2, the mapping  $F$  corresponding to the Gauss-Seidel variant of the value iteration method is also a contraction mapping with  $\rho = \alpha$ , and the convergence result of Prop. 3.2 is again a special case of the contraction mapping theorem.

## 1.5 STOCHASTIC SCHEDULING AND THE MULTIARMED BANDIT

In the problem of this section there are  $n$  projects (or activities) of which only one can be worked on at any time period. Each project  $i$  is characterized at time  $k$  by its state  $x_k^i$ . If project  $i$  is worked on at time  $k$ , one receives an expected reward  $\alpha^k R^i(x_k^i)$ , where  $\alpha \in (0, 1)$  is a discount factor; the state  $x_k^i$  then evolves according to the equation

$$x_{k+1}^i = f^i(x_k^i, w_k^i), \quad \text{if } i \text{ is worked on at time } k, \quad (5.1)$$

where  $w_k^i$  is a random disturbance with probability distribution depending on  $x_k^i$  but not on prior disturbances. The states of all idle projects are unaffected; that is,

$$x_{k+1}^i = x_k^i, \quad \text{if } i \text{ is idle at time } k. \quad (5.2)$$

We assume perfect state information and that the reward functions  $R^i(\cdot)$  are uniformly bounded above and below, so the problem comes under the discounted cost framework of Section 1.2.

We assume also that at any time  $k$  there is the option of permanently retiring from all projects, in which case a reward  $\alpha^k M$  is received and no additional rewards are obtained in the future. The retirement reward  $M$  is given and provides a parameterization of the problem, which will prove very useful. Note that for  $M$  sufficiently small it is never optimal to retire, thereby allowing the possibility of modeling problems where retirement is not a real option.

The key characteristic of the problem is the independence of the projects manifested in our three basic assumptions:

1. States of idle projects remain fixed.
2. Rewards received depend only on the state of the project currently engaged.
3. Only one project can be worked on at a time.

The rich structure implied by these assumptions makes possible a powerful methodology. It turns out that optimal policies have the form of an *index rule*; that is, for each project  $i$ , there is a function  $m^i(x^i)$  such that an optimal policy at time  $k$  is to

$$\text{retire} \quad \text{if} \quad M > \max_j \{m^j(x_k^j)\}, \quad (5.3a)$$

$$\text{work on project } i \quad \text{if} \quad m^i(x_k^i) = \max_j \{m^j(x_k^j)\} \geq M. \quad (5.3b)$$

Thus  $m^i(x_k^i)$  may be viewed as an index of profitability of operating the  $i$ th project, while  $M$  represents profitability of retirement at time  $k$ . The optimal policy is to exercise the option of maximum profitability.

The problem of this section is known as a *multiarmed bandit problem*. An analogy here is drawn between project scheduling and selecting a sequence of plays on a slot machine that has several arms corresponding to different but unknown probability distributions of payoff. With each play the distribution of the selected arm is better identified, so the tradeoff here is between playing arms with high expected payoff and exploring the winning potential of other arms.

### Index of a Project

Let  $J(x, M)$  denote the optimal reward attainable when the initial state is  $x = (x^1, \dots, x^n)$  and the retirement reward is  $M$ . From Section 1.2 we know that, for each  $M$ ,  $J(\cdot, M)$  is the unique bounded solution of Bellman's equation

$$J(x, M) = \max \left[ M, \max_i L^i(x, M, J) \right], \quad \text{for all } x, \quad (5.4)$$

where  $L^i$  is defined by

$$L^i(x, M, J) = R^i(x^i) + \alpha \mathbb{E}_{w^i} \left\{ J(x^1, \dots, x^{i-1}, f^i(x^i, w^i), x^{i+1}, \dots, x^n, M) \right\}. \quad (5.5)$$

The next proposition gives some useful properties of  $J$ .

**Proposition 5.1:** Let  $B = \max_i \max_{x^i} |R^i(x^i)|$ . For fixed  $x$ , the optimal reward function  $J(x, M)$  has the following properties as a function of  $M$ :

- (a)  $J(x, M)$  is convex and monotonically nondecreasing.
- (b)  $J(x, M)$  is constant for  $M \leq -B/(1-\alpha)$ .
- (c)  $J(x, M) = M$  for all  $M \geq B/(1-\alpha)$ .

**Proof:** Consider the value iteration method starting with the function

$$J_0(x, M) = \max[0, M].$$

Successive iterates are generated by

$$J_{k+1}(x, M) = \max \left[ M, \max_i L^i(x, M, J_k) \right], \quad k = 0, 1, \dots \quad (5.6)$$

and we know from Prop. 2.1 of Section 1.2 that

$$\lim_{k \rightarrow \infty} J_k(x, M) = J(x, M), \quad \text{for all } x, M. \quad (5.7)$$

We show inductively that  $J_k(x, M)$  has the properties (a) to (c) stated in the proposition and, by taking the limit as  $k \rightarrow \infty$ , we establish the same properties for  $J$ . Clearly,  $J_0(x, M)$  satisfies properties (a) to (c). Assume that  $J_k(x, M)$  satisfies (a) to (c). Then from Eqs. (5.5) and (5.6) it follows that  $J_{k+1}(x, M)$  is convex and monotonically nondecreasing in  $M$ , since the expectation and maximization operations preserve these properties. Verification of (b) and (c) is straightforward, and is left for the reader. Q.E.D.

Consider now a problem where there is only one project that can be worked on, say project  $i$ . The optimal reward function for this problem is denoted  $J^i(x^i, M)$  and has the properties indicated in Prop. 5.1. A typical form for  $J^i(x^i, M)$ , viewed as a function of  $M$  for fixed  $x^i$ , is shown in Fig. 1.5.1. Clearly, there is a minimal value  $m^i(x^i)$  of  $M$  for which  $J^i(x^i, M) = M$ ; that is,

$$m^i(x^i) = \min \{M \mid J^i(x^i, M) = M\}, \quad \text{for all } x^i. \quad (5.8)$$

The function  $m^i(x^i)$  is called the *index function* (or simply index) of project  $i$ . It provides an indifference threshold at each state; that is,  $m^i(x^i)$  is the retirement reward for which we are indifferent between retiring and operating the project when at state  $x^i$ .

Our objective is to show the optimality of the index rule (5.3) for the index function defined by Eq. (5.8).

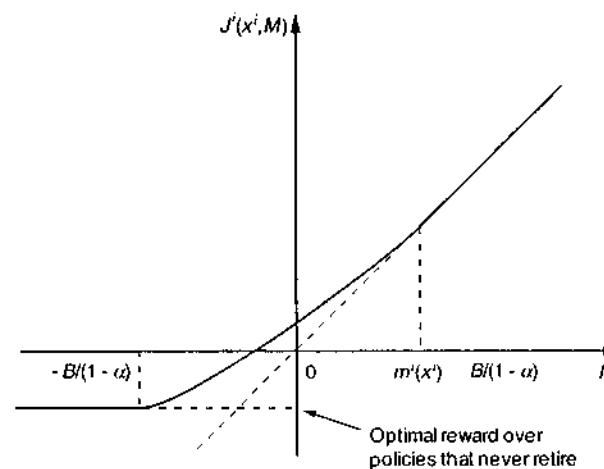


Figure 1.5.1 Form of the  $i$ -th project reward function  $J^i(x^i, M)$  for fixed  $x^i$  and definition of the index  $m^i(x^i)$ .

### Project-by-Project Retirement Policies

Consider first a problem with a single project, say project  $i$ , and a fixed retirement reward  $M$ . Then by the definition (5.8) of the index, an optimal policy is to

$$\text{retire project } i \text{ if } m^i(x^i) < M, \quad (5.9a)$$

$$\text{work on project } i \text{ if } m^i(x^i) \geq M. \quad (5.9b)$$

In other words, the project is operated continuously up to the time that its state falls into the *retirement set*

$$S^i = \{x^i \mid m^i(x^i) < M\}. \quad (5.10)$$

At that time the project is permanently retired.

Consider now the multiproject problem for fixed retirement reward  $M$ . Suppose at some time we are at state  $x = (x^1, \dots, x^n)$ . Let us ask two questions:

1. Does it make sense to retire (from all projects) when there is still a project  $i$  with state  $x^i$  such that  $m^i(x^i) > M$ ? The answer is negative. Retiring when  $m^i(x^i) > M$  cannot be optimal, since if we operate project  $i$  exclusively up to the time that its state  $x^i$  falls within the retirement set  $S^i$  of Eq. (5.10) and then retire, we will gain

a higher expected reward. [This follows from the definition (5.8) of the index and the nature of the optimal policy (5.9) for the single-project problem.]

2. Does it ever make sense to work on a project  $i$  with state in the retirement set  $S^i$  of Eq. (5.10)? Intuitively, the answer is negative; it seems unlikely that a project unattractive enough to be retired if it were the only choice would become attractive merely because of the availability of other projects that are independent in the sense assumed here.

We are led therefore to the conjecture that there is an optimal *project-by-project retirement (PPR) policy* that permanently retires projects in the same way as if they were the only project available. Thus at each time a PPR policy, when at state  $x = (x^1, \dots, x^n)$ ,

$$\text{permanently retires project } i \quad \text{if} \quad x^i \in S^i, \quad (5.11a)$$

$$\text{works on some project} \quad \text{if} \quad x^i \notin S^i \text{ for some } j, \quad (5.11b)$$

where  $S^i$  is the  $i$ th project retirement set of Eq. (5.11). Note that a PPR policy decides about retirement of projects but does not specify the project to be worked on out of those not yet retired.

The following proposition substantiates our conjecture. The proof is lengthy but quite simple.

**Proposition 5.2:** There exists an optimal PPR policy.

**Proof:** In view of Eqs. (5.4), (5.5), and (5.11), existence of a PPR policy is equivalent to having, for all  $i$ ,

$$M > L^i(x, M, J), \quad \text{for all } x \text{ with } x^i \in S^i, \quad (5.12a)$$

$$M \leq L^i(x, M, J), \quad \text{for all } x \text{ with } x^i \notin S^i, \quad (5.12b)$$

where  $L^i$  is given by

$$L^i(x, M, J) = R^i(x^i) + \alpha E_{w^i} \left\{ J(x^1, \dots, x^{i-1}, f^i(x^i, w^i), x^{i+1}, \dots, x^n, M) \right\}, \quad (5.13)$$

and  $J(x, M)$  is the optimal reward function corresponding to  $x$  and  $M$ .

The  $i$ th single-project optimal reward function  $J^i$  clearly satisfies, for all  $x^i$ ,

$$J^i(x^i, M) \leq J(x^1, \dots, x^{i-1}, x^i, x^{i+1}, \dots, x^n, M), \quad (5.14)$$

since having the option of working at projects other than  $i$  cannot decrease the optimal reward. Furthermore, from the definition of the retirement set  $S^i$  [cf. Eq. (5.10)],

$$x^i \notin S^i, \quad \text{if } M \leq R^i(x^i) + \alpha E_{w^i} \left\{ J^i(f^i(x^i, w^i), M) \right\}. \quad (5.15)$$

Using Eqs. (5.13) to (5.15), we obtain Eq. (5.12b).

It will suffice to show Eq. (5.12a) for  $i = 1$ . Denote

$\underline{x} = (x^2, \dots, x^n)$ : The state of all projects other than project 1,

$\underline{J}(x, M)$ : The optimal reward function for the problem resulting after project 1 is permanently retired,

$J(x^1, \underline{x}, M)$ : The optimal reward function for the problem involving all projects and corresponding to state  $x = (x^1, \underline{x})$ .

We will show the following inequality for all  $x = (x^1, \underline{x})$ :

$$\underline{J}(x, M) \leq J(x^1, \underline{x}, M) \leq \underline{J}(\underline{x}, M) + (J^1(x^1, M) - M). \quad (5.16)$$

In words this expresses the intuitively clear fact that at state  $(x^1, \underline{x})$  one would be happy to retire project 1 permanently if one gets in return the maximum reward that can be obtained from project 1 in excess of the retirement reward  $M$ . We claim that to show Eq. (5.12a) for  $i = 1$ , it will suffice to show Eq. (5.16). Indeed, when  $x^1 \in S^1$ , then  $J^1(x^1, M) = M$ , so from Eq. (5.16) we obtain  $J(x^1, \underline{x}, M) = \underline{J}(\underline{x}, M)$ , which is in turn equivalent to Eq. (5.12a) for  $i = 1$ .

We now turn to the proof of Eq. (5.16). Its left side is evident. To show the right side, we proceed by induction on the value iteration recursions

$$\begin{aligned} J_{k+1}(x^1, \underline{x}) &= \max \left[ M, R^1(x^1) + \alpha E \left\{ J_k(f^1(x^1, w^1), \underline{x}) \right\}, \right. \\ &\quad \left. \max_{i \neq 1} \left\{ R^i(x^i) + \alpha E \left\{ J_k(x^1, F^i(\underline{x}, w^i)) \right\} \right\} \right], \end{aligned} \quad (5.17a)$$

$$\underline{J}_{k+1}(\underline{x}) = \max \left[ M, \max_{i \neq 1} \left\{ R^i(x^i) + \alpha E \left\{ \underline{J}_k(F^i(\underline{x}, w^i)) \right\} \right\} \right], \quad (5.17b)$$

$$J_{k+1}^1(x^1) = \max \left[ M, R^1(x^1) + \alpha E \left\{ J_k^1(f^1(x^1, w^1)) \right\} \right], \quad (5.17c)$$

where, for all  $i \neq 1$  and  $\underline{x} = (x^2, \dots, x^n)$ ,

$$F^i(\underline{x}, w^i) = (x^2, \dots, x^{i-1}, f^i(x^i, w^i), x^{i+1}, \dots, x^n). \quad (5.18)$$

The initial conditions for the recursions (5.17) are

$$J_0(x^1, \underline{x}) = M, \quad \text{for all } (x^1, \underline{x}), \quad (5.19a)$$

$$\underline{J}_0(\underline{x}) = M, \quad \text{for all } \underline{x}, \quad (5.19b)$$

$$J_0^1(x^1) = M, \quad \text{for all } x^1, \quad (5.19c)$$

We know that  $J_k(x^1, \underline{x}) \rightarrow J(x^1, \underline{x}, M)$ ,  $\underline{J}_k(\underline{x}) \rightarrow \underline{J}(\underline{x}, M)$ , and  $J_k^1(x^1) \rightarrow J^1(x^1, M)$ , so to show Eq. (5.16) it will suffice to show that for all  $k$  and  $x = (x^1, \underline{x})$  we have

$$J_k(x^1, \underline{x}) \leq \underline{J}_k(\underline{x}) + (J_k^1(x^1) - M). \quad (5.20)$$

In view of the definitions (5.19), we see that Eq. (5.20) holds for  $k = 0$ . Assume that it holds for some  $k$ . We will show that it holds for  $k + 1$ . From Eq. (5.17) and the induction hypothesis (5.20), we have

$$\begin{aligned} J_{k+1}(x^1, \underline{x}) &\leq \max \left[ M, R^1(x^1) + \alpha E \{ J_k(\underline{x}) + J_k^1(f^1(x^1, w^1)) - M \}, \right. \\ &\quad \left. \max_{i \neq 1} [R^i(x^i) + \alpha E \{ J_k(F^i(\underline{x}, w^i)) + J_k^i(x^i) - M \}] \right]. \end{aligned}$$

Using the facts  $J_k(\underline{x}) \geq M$  and  $J_k^i(x^i) \geq M$  [cf. Eq. (5.17)], and the preceding equation, we see that

$$J_{k+1}(x^1, \underline{x}) \leq \max \{ \beta_1, \beta_2 \},$$

where

$$\beta_1 = \max \left[ M, R^1(x^1) + \alpha E \{ J_k^1(f^1(x^1, w^1)) \} \right] + \alpha (J_k(\underline{x}) - M).$$

$$\beta_2 = \max \left[ M, \max_{i \neq 1} [R^i(x^i) + \alpha E \{ J_k(F^i(\underline{x}, w^i)) \}] \right] + \alpha (J_k^i(x^i) - M).$$

Using Eqs. (5.17b), (5.17c), and the preceding equations, we see that

$$J_{k+1}(x^1, \underline{x}) \leq \max [J_{k+1}^1(x^1) + J_k(\underline{x}) - M, J_{k+1}^i(x^i) + J_k^i(x^i) - M]. \quad (5.21)$$

It can be seen from Eqs. (5.17) and (5.19) that  $J_k^i(x^i) \leq J_{k+1}^i(x^i)$  and  $J_k(\underline{x}) \leq J_{k+1}(\underline{x})$  for all  $k$ ,  $x^1$ , and  $\underline{x}$ , so from Eq. (5.21) we obtain that Eq. (5.20) holds for  $k + 1$ . The induction is complete. **Q.E.D.**

As a first step towards showing optimality of the index rule, we use the preceding proposition to derive an expression for the partial derivative of  $J(x, M)$  with respect of  $M$ .

**Lemma 5.1:** For fixed  $x$ , let  $K_M$  denote the retirement time under an optimal policy when the retirement reward is  $M$ . Then for all  $M$  for which  $\partial J(x, M)/\partial M$  exists we have

$$\frac{\partial J(x, M)}{\partial M} = E \{ \alpha^{K_M} \mid x_0 = x \}.$$

**Proof:** Fix  $x$  and  $M$ . Let  $\pi^*$  be an optimal policy and let  $K_M$  be the retirement time under  $\pi^*$ . If  $\pi^*$  is used for a problem with retirement reward  $M + \epsilon$ , we receive

$$E\{\text{reward prior to retirement}\} + (M + \epsilon) E\{\alpha^{K_M}\} = J(x, M) + \epsilon E\{\alpha^{K_M}\}.$$

The optimal reward  $J(x, M + \epsilon)$  when the retirement reward is  $M + \epsilon$  is no less than the preceding expression, so

$$J(x, M + \epsilon) \geq J(x, M) + \epsilon E\{\alpha^{K_M}\}.$$

Similarly, we obtain

$$J(x, M - \epsilon) \geq J(x, M) - \epsilon E\{\alpha^{K_M}\}.$$

For  $\epsilon > 0$ , these two relations yield

$$\frac{J(x, M) - J(x, M - \epsilon)}{\epsilon} \leq E\{\alpha^{K_M}\} \leq \frac{J(x, M + \epsilon) - J(x, M)}{\epsilon}.$$

The result follows by taking  $\epsilon \rightarrow 0$ . **Q.E.D.**

Note that the convexity of  $J(x, \cdot)$  with respect to  $M$  (Prop. 4.1) implies that the derivative  $\partial J(x, M)/\partial M$  exists almost everywhere with respect to Lebesgue measure [Roc70]. Furthermore, it can be shown that  $\partial J(x, M)/\partial M$  exists for all  $M$  for which the optimal policy is unique.

For a given  $M$ , initial state  $x$ , and optimal PPR policy, let  $T_i$  be the retirement time of project  $i$  if it were the only project available, let  $T$  be the retirement time for the multiproject problem. Both  $T_i$  and  $T$  take values that are either nonnegative or  $\infty$ . The existence of an optimal PPR policy implies that we must have

$$T = T_1 + \cdots + T_n$$

and in addition  $T_i$ ,  $i = 1, \dots, n$ , are independent random variables. Therefore,

$$E\{\alpha^T\} = E\{\alpha^{T_1+\cdots+T_n}\} = \prod_{i=1}^n E\{\alpha^{T_i}\}.$$

Using Lemma 5.1, we obtain

$$\frac{\partial J(x, M)}{\partial M} = \prod_{i=1}^n \frac{\partial J^i(x^i, M)}{\partial M}. \quad (5.22)$$

### Optimality of the Index Rule

We are now ready to show our main result.

**Proposition 5.3:** The index rule (5.3) is an optimal stationary policy.

**Proof:** Fix  $x = (x^1, \dots, x^n)$ , denote

$$m(x) = \max_j \{m^j(x^j)\},$$

and let  $i$  be such that

$$m^i(x^i) = \max_j \{m^j(x^j)\}.$$

If  $m(x) < M$  the optimality of the index rule (5.3a) at state  $x$  follows from the existence of an optimal PPR policy. If  $m(x) \geq M$ , we note that

$$J^i(x^i, M) = R^i(x^i) + \alpha E\{J^i(f^i(x^i, w^i), M)\}$$

and then use this relation together with Eq. (5.22) to write

$$\begin{aligned} \frac{\partial J(x, M)}{\partial M} &= \frac{\partial J(x^t, M)}{\partial M} \cdot \prod_{j \neq i} \frac{\partial J(x^j, M)}{\partial M} \\ &= \alpha \frac{\partial}{\partial M} E \left\{ J^i(f^i(x^i, w^i), M) \cdot \prod_{j \neq i} \frac{\partial J(x^j, M)}{\partial M} \right\} \\ &= \alpha E \left\{ \frac{\partial}{\partial M} J^i(f^i(x^i, w^i), M) \cdot \prod_{j \neq i} \frac{\partial J(x^j, M)}{\partial M} \right\} \\ &= \alpha E \left\{ \frac{\partial}{\partial M} J(x^1, \dots, x^{t-1}, f^i(x^i, w^i), x^{t+1}, \dots, x^n, M) \right\} \\ &= \alpha \frac{\partial}{\partial M} E\{J(x^1, \dots, x^{t-1}, f^i(x^i, w^i), x^{t+1}, \dots, x^n, M)\}, \end{aligned}$$

and finally

$$\frac{\partial J(x, M)}{\partial M} = \frac{\partial}{\partial M} L^i(x, M, J),$$

where

$$L^i(x, M, J) = R^i(x^i) + \alpha E\{J(x^1, \dots, x^{t-1}, f^i(x^i, w^i), x^{t+1}, \dots, x^n, M)\}.$$

(The interchange of differentiation and expectation can be justified for almost all  $M$ ; see [Ber73a].) By the existence of an optimal PPR policy, we also have

$$J(x, m(x)) = L^i(x, m(x), J).$$

Therefore, the convex functions  $J(x, M)$  and  $L^i(x, M, J)$  viewed as functions of  $M$  for fixed  $x$  are equal for  $M = m(x)$  and have equal derivative for almost all  $M \leq m(x)$ . It follows that for all  $M \leq m(x)$  we have

$$J(x, M) = L^i(x, M, J).$$

This implies that the index rule (5.3b) is optimal for all  $x$  with  $m(x) \geq M$ . Q.E.D.

### Deteriorating and Improving Cases

It is evident that great simplification results from the optimality of the index rule (5.3), since optimization of a multiproject problem has been reduced to  $n$  separate single-project optimization problems. Nonetheless, solution of each of these single-project problems can be complicated. Under certain circumstances, however, the situation simplifies.

Suppose that for all  $i$ ,  $x^i$ , and  $w^i$  that can occur with positive probability, we have either

$$m^i(x^i) \leq m^i(f^i(x^i, w^i)) \quad (5.23)$$

or

$$m^i(x^i) \geq m^i(f^i(x^i, w^i)). \quad (5.24)$$

Under Eq. (5.23) [or Eq. (5.24)] projects become more (less) profitable as they are worked on. We call these cases *improving* and *deteriorating*, respectively.

In the improving case the nature of the optimal policy is evident: either retire at the first period or else select a project with maximal index at the first period and continue engaging that project for all subsequent periods.

In the deteriorating case, note that Eq. (5.24) implies that if retirement is optimal when at state  $x^i$  then it is also optimal at each state  $f^i(x^i, w^i)$ . Therefore, for all  $x^i$  such that  $M = m^i(x^i)$  we have, for all  $w^i$ ,

$$J^i(x^i, M) = M, \quad J^i(f^i(x^i, w^i), M) = M.$$

From Bellman's equation

$$J^i(x^i, M) = \max [M, R^i(x^i) + \alpha E\{J^i(f^i(x^i, w^i), M)\}]$$

we obtain

$$m^i(x^i) = R^i(x^i) + \alpha m^i(x^i)$$

or

$$m^i(x^i) = \frac{R^i(x^i)}{1 - \alpha}. \quad (5.25)$$

Thus the optimal policy in the deteriorating case is

retire if  $M > \max_i \frac{R^i(x^i)}{1 - \alpha}$  and otherwise engage the project  $i$  with maximal one-step reward  $R^i(x^i)$ .

### Example 5.1 (Treasure Hunting)

Consider a search problem involving  $N$  sites. Each site  $i$  may contain a treasure with expected value  $v_i$ . A search at site  $i$  costs  $c_i$  and reveals the treasure with probability  $\beta_i$  (assuming a treasure is there). Let  $P_i$  be the probability that there is a treasure at site  $i$ . We take  $P_i$  as the state of the project corresponding to searching site  $i$ . Then the corresponding one-step reward is

$$R'(P_i) = \beta_i P_i v_i - c_i. \quad (5.26)$$

If a search at site  $i$  does not reveal the treasure, the probability  $P_i$  drops to

$$\bar{P}_i = \frac{P_i(1-\beta_i)}{P_i(1-\beta_i) + 1 - p_i},$$

as can be verified using Bayes' rule. If the search finds the treasure, the probability  $P_i$  drops to zero, since the treasure is removed from the site. Based on this and the fact that  $R'(P_i)$  is increasing with  $P_i$  [cf. Eq. (5.26)], it is seen that the deteriorating condition (5.24) holds. Therefore, it is optimal to search the site  $i$  for which the expression  $R'(P_i)$  of Eq. (5.26) is maximal, provided  $\max_i R'(P_i) > 0$ , and to retire if  $R'(P_i) \leq 0$  for all  $i$ .

## 1.6 NOTES, SOURCES, AND EXERCISES

Many authors have contributed to the analysis of the discounted problem with bounded cost per stage, most notably Shapley [Sha53], Bellman [Bel57], and Blackwell [Blc65]. For variations and extensions of the problem involving multiple criteria, weighted criteria, and constraints, see [FeS94], [Gho90], [Ros89], and [WhK80]. The mathematical issues relating to measurability concerns are analyzed extensively in [BeS78], [DyY79], and [Her89].

The error bounds given in Section 1.3 and Exercise 1.9 are improvements on results of [McQ66] (see [Por71], [Por75], [Ber76], and [PoT78]). The corresponding convergence rate was discussed in [Mor71] and [MoW77]. The Gauss-Seidel method for discounted problems was proposed in [Kus71] (see also [Has68]). An extensive discussion of the convergence aspects of the method and related background is given in Section 2.6 of [BeT89a]. The material on the generic rank-one correction, including the convergence analysis of Exercise 1.8, is new; see [Ber93], which also describes a multiple-rank correction method where the effect of several eigenvalues is nullified. Value iteration is particularly well-suited for parallel computation; see e.g., [AMT93], [BeT89a].

Policy iteration for discounted problems was proposed in [Bel57]. The modified policy iteration algorithm was suggested and analyzed in [PuS78]

and [PuS82]. The approximate policy iteration analysis and the convergence proof of policy iteration for an infinite state space (Prop. 3.6) are new and were developed in collaboration with J. Tsitsiklis. The relation between policy iteration and Newton's method (Exercise 1.10) was pointed out in [PoA69] and was further discussed in [PuB78].

The material on adaptive aggregation is due to [BeC89]. In an alternative aggregation approach [SPK89], the aggregate states are fixed. Changing adaptively the aggregate states from one iteration to the next depending on the progress of the computation has a potentially significant effect on the efficiency of the computation for difficult problems where the ordinary value iteration method is very slow.

The linear programming approach of Section 1.3.4 was proposed in [D'Ep60]. There is a relation between policy iteration and the simplex method applied to solving the linear program associated with the discounted problem. In particular, it can be shown that the simplex method for linear programming with a block pivoting rule is mathematically equivalent to the policy iteration algorithm. There are also duality connections that relate the linear programming approach with randomized policies, constraints, and multiple criteria; see e.g., [Kal83], [Put94]. Approximation methods using basis functions and linear programming were proposed in [ScS85].

A complexity analysis of finite-state infinite horizon problems is given in [PaT87]. Discretization methods that approximate infinite state space systems with finite-state Markov chains, are discussed in [Ber75], [Fox71], [HaL86], [Whi78], [Whi79], and [Whi80a]. For related multigrid approximation methods and associated complexity analysis, see [ChT89] and [ChT91]. A different approach to deal with infinite state spaces, which is based on randomization, has been introduced in [Rus94]; see also [Rus95]. Further material on computational methods may be found in [Put78].

The role of contraction mappings in discounted problems was first recognized and exploited in [Sha53], which considers two-player dynamic games. Abstract DP models and the implications of monotonicity and contraction have been explored in detail in [Den67], [Ber77], [BeS78], [VeP84], and [VeP87].

The index rule solution of the multiarmed bandit problem is due to [Git79] and [GiJ74]. Subsequent contributions include [Whi80b], [Ked81], [Whi81], and [Whi82]. The proof given here is due to [Tsi86]. Alternative proofs and analysis are given in [VWB85], [NTW89], [Tso91], [Web92], [BeN93], [Tsi93b], [BPT94a], and [BPT94b]. Much additional work on the subject is described in [Kum85] and [KuV86].

Finally, we note that even though our analysis in this chapter requires a countable disturbance space, it may still serve as the starting point of analysis of problems with uncountable disturbance space. This can be done by reducing such problems to deterministic problems with state space a set of probability measures. The basic idea of this reduction is demonstrated

in Exercise 1.13. The advanced reader may consult [BeS78] (Section 9.2), and see how such a reduction can be effected for a very broad class of finite and infinite horizon problems.

## EXERCISES

### 1.1

Write a computer problem and compute iteratively the vector  $J_\mu$  satisfying

$$J_\mu = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \alpha \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 - \epsilon & \epsilon \\ 0 & \epsilon & 1 - \epsilon \end{pmatrix} J_\mu.$$

Do your computations for all combinations of  $\alpha = 0.9$  and  $\alpha = 0.999$ , and  $\epsilon = 0.5$  and  $\epsilon = 0.001$ . Try value iteration with and without error bounds, and also adaptive aggregation with two aggregate classes of states. Discuss your results.

### 1.2

The purpose of this problem is to show that shortest path problems with a discount factor make little sense. Suppose that we have a graph with a nonnegative length  $a_{ij}$  for each arc  $(i, j)$ . The cost of a path  $(i_0, i_1, \dots, i_m)$  is  $\sum_{k=0}^{m-1} \alpha^k a_{i_k i_{k+1}}$ , where  $\alpha$  is a discount factor from  $(0, 1)$ . Consider the problem of finding a path of minimum cost that connects two given nodes. Show that this problem need not have a solution.

### 1.3

Consider a problem similar to that of Section 1.1 except that when we are at state  $x_k$ , there is a probability  $\beta$ , where  $0 < \beta < 1$ , that the next state  $x_{k+1}$  will be determined according to  $x_{k+1} = f(x_k, u_k, w_k)$  and a probability  $(1-\beta)$  that the system will move to a termination state, where it stays permanently thereafter at no cost. Show that even if  $\alpha = 1$ , the problem can be put into the discounted cost framework.

### 1.4

Consider a problem similar to that of Section 1.2 except that the discount factor  $\alpha$  depends on the current state  $x_k$ , the control  $u_k$ , and the disturbance  $w_k$ ; that is, the cost function has the form

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} E_{\substack{u_k \\ k=0,1,\dots}} \left\{ \sum_{k=0}^{N-1} \alpha_{\pi,k} g(x_k, \mu_k(x_k), w_k) \right\},$$

where

$$\alpha_{\pi,k} = \alpha(x_0, \mu_0(x_0), w_0) \alpha(x_1, \mu_1(x_1), w_1) \cdots \alpha(x_k, \mu_k(x_k), w_k),$$

with  $\alpha(x, u, w)$  a given function satisfying

$$\begin{aligned} 0 &\leq \min \{ \alpha(x, u, w) \mid x \in S, u \in C, w \in D \} \\ &\leq \max \{ \alpha(x, u, w) \mid x \in S, u \in C, w \in D \} \\ &< 1. \end{aligned}$$

Argue that the results and algorithms of Sections 1.2 and 1.3 have direct counterparts for such problems.

### 1.5 (Column Reduction [Por75])

The purpose of this problem is to provide a transformation of a certain type of discounted problem into another discounted problem with smaller discount factor. Consider the  $n$ -state discounted problem under the assumptions of Section 1.3. The cost per stage is  $g(i, u)$ , the discount factor is  $\alpha$ , and the transition probabilities are  $p_{ij}(u)$ . For each  $j = 1, \dots, n$ , let

$$m_j = \min_{i=1, \dots, n} \min_{u \in U(i)} p_{ij}(u).$$

For all  $i$ ,  $j$ , and  $u$ , let

$$\tilde{p}_{ij}(u) = \frac{p_{ij}(u) - m_j}{1 - \sum_{k=1}^n m_k},$$

assuming that  $\sum_{k=1}^n m_k < 1$ .

- (a) Show that  $\tilde{p}_{ij}(u)$  are transition probabilities.
- (b) Consider the discounted problem with cost per stage  $g(i, u)$ , discount factor  $\alpha(1 - \sum_{j=1}^n m_j)$ , and transition probabilities  $\tilde{p}_{ij}(u)$ . Show that this problem has the same optimal policies as the original, and that its optimal cost vector  $J'$  satisfies

$$J^* = J' + \frac{\alpha \sum_{j=1}^n m_j J'(j)}{1 - \alpha} e,$$

where  $J^*$  is the optimal cost vector of the original problem and  $e$  is the unit vector.

## 1.6

Let  $\bar{J} : S \mapsto \mathbb{R}$  be any bounded function on  $S$  and consider the value iteration method of Section 1.3 with a starting function  $J : S \mapsto \mathbb{R}$  of the form

$$J(x) = \bar{J}(x) + r, \quad x \in S,$$

where  $r$  is some scalar. Show that the bounds  $(T^k J)(x) + \underline{c}_k$  and  $(T^k J)(x) + \bar{c}_k$  of Prop. 3.1 are independent of the scalar  $r$  for all  $x \in S$ . Show also that if  $S$  consists of a single state  $\hat{x}$  (i.e.,  $S = \{\hat{x}\}$ ), then

$$(TJ)(\hat{x}) + \underline{c}_1 = (TJ)(\hat{x}) + \bar{c}_1 = J^*(\hat{x}).$$

## 1.7 (Jacobi Version of Value Iteration)

Consider the problem of Section 1.3 and the version of the value iteration method that starts with an arbitrary function  $J : S \mapsto \mathbb{R}$  and generates recursively  $FJ, F^2J, \dots$ , where  $F$  is the mapping given by

$$(FJ)(i) = \min_{u \in U(i)} \frac{g(i, u) + \alpha \sum_{j \neq i} p_{ij}(u)J(j)}{1 - \alpha p_{ii}(u)}.$$

Show that  $(F^k J)(i) \rightarrow J^*(i)$  as  $k \rightarrow \infty$  and provide a rate of convergence estimate that is at least as favorable as the one for the ordinary method (cf. Prop. 2.3).

## 1.8 (Convergence Properties of Rank-One Correction [Ber93])

Consider the solution of the system  $J = FJ$ , where  $F : \mathbb{R}^n \mapsto \mathbb{R}^n$  is the mapping

$$FJ = h + QJ,$$

$h$  is a given vector in  $\mathbb{R}^n$ , and  $Q$  is an  $n \times n$  matrix. Consider the generic rank-one correction iteration  $J := MJ$ , where  $M : \mathbb{R}^n \mapsto \mathbb{R}^n$  is the mapping

$$MJ = FJ + \gamma z,$$

and

$$z = Qd, \quad \gamma = \frac{(d - z)'(FJ - J)}{\|d - z\|^2},$$

- (a) Show that any solution  $J^*$  of the system  $J = FJ$  satisfies  $J^* = MJ^*$ .
- (b) Verify that the value iteration method that uses the error bounds in the manner of Eq. (3.12) is a special case of the iteration  $J := MJ$  with  $d$  equal to the unit vector.

- (c) Assume that  $d$  is an eigenvector of  $Q$ , let  $\lambda$  be the corresponding eigenvalue, and let  $\lambda_1, \dots, \lambda_{n-1}$  be the remaining eigenvalues. Show that  $MJ$  can be written as

$$MJ = h + R J,$$

where  $h$  is some vector in  $\mathbb{R}^n$  and

$$R = Q - \frac{\lambda}{(1 - \lambda)\|d\|^2} dd'(I - Q).$$

Show also that  $Rd = 0$  and that for all  $k$  and  $J$ ,

$$R^k = RQ^{k-1}, \quad M^k J = M(F^{k-1}J).$$

Furthermore, the eigenvalues of  $R$  are  $0, \lambda_1, \dots, \lambda_{n-1}$ . (This last statement requires a somewhat complicated proof; see [Ber93].)

- (d) Let  $d$  be as in part (c), and suppose that  $e_1, \dots, e_{n-1}$  are eigenvectors corresponding to  $\lambda_1, \dots, \lambda_{n-1}$ . Suppose that a vector  $J$  can be written as

$$J = J^* + \xi e + \sum_{i=1}^{n-1} \xi_i e_i,$$

where  $J^*$  is a solution of the system. Show that, for all  $k \geq 1$ ,

$$M^k J = J^* + \sum_{i=1}^{n-1} \xi_i \lambda_i^{k-1} R e_i,$$

so that if  $\lambda$  is a dominant eigenvalue and  $\lambda_1, \dots, \lambda_{n-1}$  lie within the unit circle,  $M^k J$  converges to  $J^*$  at a rate governed by the subdominant eigenvalue. Note: This result can be generalized for the case where  $Q$  does not have a full set of linearly independent eigenvectors, and for the case where  $F$  is modified through multiple-rank corrections [Ber93].

## 1.9 (Generalized Error Bounds [Ber76])

Let  $S$  be a set and  $B(S)$  be the set of all bounded real-valued functions on  $S$ . Let  $T : B(S) \mapsto B(S)$  be a mapping with the following two properties:

- (1)  $TJ \leq TJ'$  for all  $J, J' \in B(S)$  with  $J \leq J'$ .
- (2) For every scalar  $r \neq 0$  and all  $x \in S$ ,

$$\alpha_1 \leq \frac{(T(J + rc))(x) - (TJ)(x)}{r} \leq \alpha_2,$$

where  $\alpha_1, \alpha_2$  are two scalars with  $0 \leq \alpha_1 \leq \alpha_2 < 1$ .

- (a) Show that  $T$  is a contraction mapping on  $B(S)$ , and hence for every  $J \in B(S)$  we have

$$\lim_{k \rightarrow \infty} (T^k J)(x) = J^*(x), \quad x \in S,$$

where  $J^*$  is the unique fixed point of  $T$  in  $B(S)$ .

- (b) Show that for all  $J \in B(S)$ ,  $x \in S$ , and  $k = 1, 2, \dots$ ,

$$\begin{aligned} (T^k J)(x) + c_k &\leq (T^{k+1} J)(x) + c_{k+1} \leq J^*(x) \leq (T^{k+1} J)(x) + \bar{c}_{k+1} \\ &\leq (T^k J)(x) + \bar{c}_k, \end{aligned}$$

where for all  $k$

$$\begin{aligned} c_k &= \min \left\{ \frac{\alpha_1}{1 - \alpha_1} \min_{x \in S} [(T^k J)(x) - (T^{k-1} J)(x)], \right. \\ &\quad \left. \frac{\alpha_2}{1 - \alpha_2} \min_{x \in S} [(T^k J)(x) - (T^{k-1} J)(x)] \right\}, \end{aligned} \quad (6.1)$$

$$\begin{aligned} \bar{c}_k &= \max \left\{ \frac{\alpha_1}{1 - \alpha_1} \max_{x \in S} [(T^k J)(x) - (T^{k-1} J)(x)], \right. \\ &\quad \left. \frac{\alpha_2}{1 - \alpha_2} \max_{x \in S} [(T^k J)(x) - (T^{k-1} J)(x)] \right\}. \end{aligned} \quad (6.2)$$

A geometric interpretation of these relations for the case where  $S$  consists of a single element is provided in Fig. 1.6.1.

- (c) Consider the following algorithm:

$$J_k(x) = (TJ_{k-1})(x) + \gamma_k, \quad x \in S,$$

where  $J_0$  is any function in  $B(S)$ ,  $\gamma_k$  is any scalar in the range  $[c_k, \bar{c}_k]$ , and  $c_k$  and  $\bar{c}_k$  are given by Eqs. (6.1) and (6.2) with  $(T^k J)(x) - (T^{k-1} J)(x)$  replaced by  $(TJ_{k-1})(x) - J_{k-1}(x)$ . Show that for all  $k$ ,

$$\max_{x \in S} |J_k(x) - J^*(x)| \leq \alpha_2^k \max_{x \in S} |J_0(x) - J^*(x)|.$$

- (d) Let  $J \in \mathbb{R}^n$  and consider the equation  $J = TJ$ , where

$$TJ = h + MJ$$

and the vector  $h \in \mathbb{R}^n$  and the matrix  $M$  are given. Let  $s_i$  be the  $i$ th row sum of  $M$ , that is,

$$s_i = \sum_{j=1}^n m_{ij},$$

and let  $\alpha_1 = \min_i s_i$ ,  $\alpha_2 = \max_i s_i$ . Show that if the elements  $m_{ij}$  of  $M$  are all nonnegative and  $\alpha_2 < 1$ , then the conclusions of parts (a) and (b) hold.

- (e) [Por75] Consider the Gauss-Seidel method for solving the system  $J = g + \alpha PJ$ , where  $0 < \alpha < 1$  and  $P$  is a transition probability matrix. Use part (d) to obtain suitable error bounds.

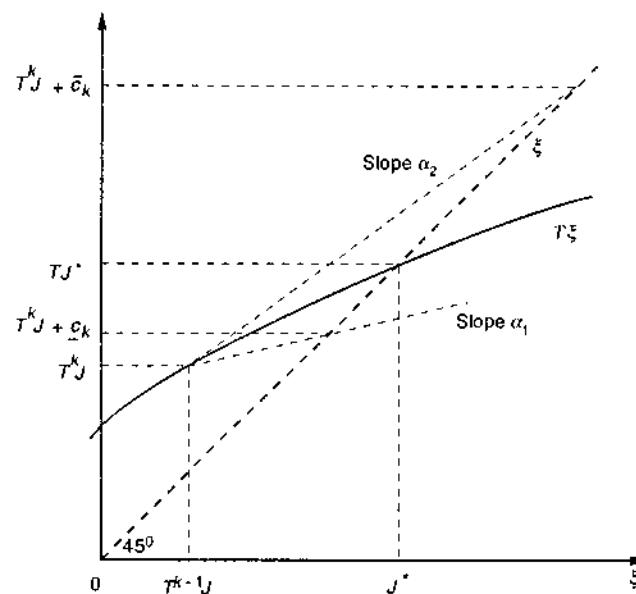


Figure 1.6.1 Graphical interpretation of the error bounds of Exercise 1.9.

### 1.10 (Policy Iteration and Newton's Method)

The purpose of this problem is to demonstrate a relation between policy iteration and Newton's method for solving nonlinear equations. Consider an equation of the form  $F(J) = 0$ , where  $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ . Given a vector  $J_k \in \mathbb{R}^n$ , Newton's method determines  $J_{k+1}$  by solving the linear system of equations

$$F(J_k) + \frac{\partial F(J_k)}{\partial J} (J_{k+1} - J_k) = 0,$$

where  $\partial F(J_k)/\partial J$  is the Jacobian matrix of  $F$  evaluated at  $J_k$ .

- (a) Consider the discounted finite-state problem of Section 1.3 and define

$$F(J) = TJ - J.$$

Show that if there is a unique  $\mu$  such that

$$T_\mu J = TJ,$$

then the Jacobian matrix of  $F$  at  $J$  is

$$\frac{\partial F(J)}{\partial J} = \alpha P_\mu - I,$$

where  $I$  is the  $n \times n$  identity.

- (b) Show that the policy iteration algorithm can be identified with Newton's method for solving  $P(J) = 0$  (assuming it gives a unique policy at each step).

### 1.11 (Minimax Problems)

Provide analogs of the results and algorithms of Sections 1.2 and 1.3 for the minimax problem where the cost is

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \max_{\substack{w_k \in W(x_k, \mu_k(x_k)) \\ k=0,1,\dots}} \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k),$$

$g$  is bounded,  $x_k$  is generated by  $x_{k+1} = f(x_k, \mu_k(x_k), w_k)$ , and  $W(x, u)$  is a given nonempty subset of  $D$  for each  $(x, u) \in S \times C$ . (Compare with Exercise 1.5 in Chapter 1 of Vol. I.)

### 1.12 (Data Transformations [Sch72])

A finite-state problem where the discount factor at each stage depends on the state can be transformed into a problem with state independent discount factors. To see this, consider the following set of equations in the variables  $J(i)$ :

$$J(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n m_{ij}(u) J(j) \right], \quad i = 1, \dots, n, \quad (6.3)$$

where we assume that for all  $i, u \in U(i)$ , and  $j$ ,  $m_{ij}(u) \geq 0$  and

$$M_i(u) = \sum_{j=1}^n m_{ij}(u) < 1,$$

Let

$$\alpha = \max_{i=1, \dots, n} \left\{ \frac{M_i(u) + m_{ii}(u)}{1 - m_{ii}(u)} \right\},$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}$$

and define, for all  $i$  and  $j$ ,

$$\bar{g}(i, u) := \frac{g(i, u)(1 - \alpha)}{1 - M_i(u)},$$

$$\bar{m}_{ij}(u) = \delta_{ij} + \frac{(1 - \alpha)(m_{ij}(u) - \delta_{ij})}{1 - M_i(u)},$$

Show that, for all  $i$  and  $j$ ,

$$\sum_{j=1}^n \bar{m}_{ij}(u) = \alpha < 1, \quad \bar{m}_{ij}(u) \geq 0,$$

and that a solution  $\{J(i) \mid i = 1, \dots, n\}$  of Eq. (6.3) is also a solution of the equations

$$J(i) = \min_{u \in U(i)} \left[ \bar{g}(i, u) + \sum_{j=1}^n \bar{m}_{ij}(u) J(j) \right], \quad i = 1, \dots, n,$$

### 1.13 (Stochastic to Deterministic Problem Transformation)

Under the assumptions and notation of Section 1.3, consider the controlled system

$$p_{k+1} = p_k P_{\mu_k}, \quad k = 0, 1, \dots,$$

where  $p_k$  is a probability distribution over  $S$  viewed as a row vector, and  $P_{\mu_k}$  is the transition probability matrix corresponding to the control function  $\mu_k$ . The state is  $p_k$  and the control is  $\mu_k$ . Consider also the cost function

$$\lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^k p_k g_{p_k},$$

Show that the optimal cost and an optimal policy for the deterministic problem involving the above system and cost function yield the optimal cost and an optimal policy for the discounted cost problem of Section 1.3.

### 1.14 (Threshold Policies and Policy Iteration)

- (a) Consider the machine replacement example of Section 1.2, and assume that the condition (2.10) holds. Let us define a *threshold* policy to be a stationary policy that replaces if and only if the state is greater than or equal to some fixed state  $i$ . Suppose that we start the policy iteration algorithm using a threshold policy. Show that all the subsequently generated policies will be threshold policies, so that the algorithm will terminate after at most  $n$  iterations.
- (b) Prove the result of part (a) for the asset selling example of Section 1.2, assuming that there is a finite number of values that the offer  $x_k$  can take. Here, a threshold policy is a stationary policy that sells the asset if the offer is higher than a certain fixed number.

### 1.15 (Distributed Asynchronous DP [Ber82a], [BeT89a])

The value iteration method is well suited for distributed (or parallel) computation since the iteration

$$J(i) := (TJ)(i)$$

can be executed in parallel for all states  $i$ . Consider the finite-state discounted problem of Section 1.3, and assume that the above iteration is executed *asynchronously* at a different processor  $i$  for each state  $i$ . By this we mean that the  $i$ th processor holds a vector  $J^i$  and updates the  $i$ th component of that vector at *arbitrary* times with an iteration of the form

$$J^i(i) := (TJ^i)(i),$$

and at *arbitrary* times transmits the results of the latest computation to other processors  $m$  who then update  $J^m(i)$  according to

$$J^m(i) := J^i(i).$$

Assume that all processors never stop computing and transmitting the results of their computation to the other processors. Show that the estimates  $J_t^i$  of the optimal cost function available at each processor  $i$  at time  $t$  converge to the optimal solution function  $J^*$  as  $t \rightarrow \infty$ . *Hint:* Let  $\bar{J}$  and  $\underline{J}$  be two functions such that  $\underline{J} \leq T\bar{J}$  and  $T\bar{J} \leq \bar{J}$ , and suppose that for all initial estimates  $J_0^i$  of the processors, we have  $\underline{J} \leq J_0^i \leq \bar{J}$ . Show that the estimates  $J_t^i$  of the processors at time  $t$  satisfy  $\underline{J} \leq J_t^i \leq \bar{J}$  for all  $t \geq 0$ , and  $T\underline{J} \leq J_t^i \leq T\bar{J}$  for  $t$  sufficiently large.

### 1.16

Assume that we have two gold mines, Anaconda and Bonanza, and a gold-mining machine. Let  $x_A$  and  $x_B$  be the current amounts of gold in Anaconda and Bonanza, respectively. When the machine is used in Anaconda (or Bonanza), there is a probability  $p_A$  (or  $p_B$ , respectively) that  $r_A x_A$  (or  $r_B x_B$ , respectively) of the gold will be mined without damaging the machine, and a probability  $1 - p_A$  (or  $1 - p_B$ , respectively) that the machine will be damaged beyond repair and no gold will be mined. We assume that  $0 < r_A < 1$  and  $0 < r_B < 1$ .

- (a) Assume that  $p_A = p_B = p$ , where  $0 < p < 1$ . Find the mine selection policy that maximizes the expected amount of gold mined before the machine breaks down. *Hint:* This problem can be viewed as a discounted multiarmed bandit problem with a discount factor  $p$ .
- (b) Assume that  $p_A < 1$  and  $p_B = 1$ . Argue that the optimal expected amount of gold mined has the form  $J^*(x_A, x_B) = \hat{J}_A(x_A) + x_B$ , where  $\hat{J}_A(x_A)$  is the optimal expected amount of gold mined if mining is restricted just to Anaconda. Show that there is no policy that attains the optimal amount  $J^*(x_A, x_B)$ .

### 1.17 (The Tax Problem [VWB85])

This problem is similar to the multiarmed bandit problem. The only difference is that, if we engage project  $i$  at period  $k$ , we pay a tax  $\alpha^k C^i(x^k)$  for every other project  $j$  [for a total of  $\alpha^k \sum_{j \neq i} C^j(x^k)$ ], instead of earning a reward  $\alpha^k R^i(x^k)$ . The objective is to find a project selection policy that minimizes the total tax paid. Show that the problem can be converted into a bandit problem with reward function for project  $i$  equal to

$$R^i(x^k) = C^i(x^k) - \alpha E\{C^i(f^i(x^k, w^k))\}.$$

### 1.18 (The Restart Problem [KaV87])

The purpose of this problem is to show that the index of a project in the multiarmed bandit context can be calculated by solving an associated infinite horizon discounted cost problem. In what follows we consider a single project with reward function  $R(x)$ , a fixed initial state  $x_0$ , and the calculation of the value of index  $m(x_0)$  for that state. Consider the problem where at state  $x_k$  and time  $k$  there are two options: (1) Continue, which brings reward  $\alpha^k R(x_k)$  and moves the project to state  $x_{k+1} = f(x_k, w)$ , or (2) restart the project, which moves the state to  $x_0$ , brings reward  $\alpha^k R(x_0)$ , and moves the project to state  $x_{k+1} = f(x_0, w)$ . Show that the optimal reward functions of this problem and of the bandit problem with  $M = m(x_0)$  are identical, and therefore the optimal reward for both problems when starting at  $x_0$  equals  $m(x_0)$ . *Hint:* Show that Bellman's equation for both problems takes the form

$$J(x) = \max[R(x_0) + \alpha E\{J(f(x_0, w))\}, R(x) + \alpha E\{J(f(x, w))\}].$$

## *Stochastic Shortest Path Problems*

### Contents

2.1. Main Results . . . . .	p. 78
2.2. Computational Methods . . . . .	p. 87
2.2.1. Value Iteration . . . . .	p. 88
2.2.2. Policy Iteration . . . . .	p. 91
2.3. Simulation-Based Methods . . . . .	p. 94
2.3.1. Policy Evaluation by Monte-Carlo Simulation . . . . .	p. 95
2.3.2. Q-Learning . . . . .	p. 99
2.3.3. Approximations . . . . .	p. 101
2.3.4. Extensions to Discounted Problems . . . . .	p. 118
2.3.5. The Role of Parallel Computation . . . . .	p. 120
2.4. Notes, Sources, and Exercises . . . . .	p. 121

In this chapter we consider a stochastic version of the shortest path problem discussed in Chapter 2 of Vol. I. An introductory analysis of this problem was given in Section 7.2 of Vol. I. The analysis of this chapter is more sophisticated and uses weaker assumptions. In particular, we make assumptions that generalize those made for deterministic shortest path problems in Chapter 2 of Vol. I.

In this chapter we also discuss another major topic of this book. In particular, in Section 2.3 we develop simulation-based methods, possibly involving approximations, which are suitable for complex problems that involve a large number of states and/or a lack of an explicit mathematical model. These methods are most economically developed in the context of stochastic shortest path problems. They can then be extended to discounted problems, and this is done in Section 2.3.1. Further extensions to average cost per stage problems are discussed in Section 4.3.4.

## 2.1 MAIN RESULTS

Suppose that we have a graph with nodes  $1, 2, \dots, n, t$ , where  $t$  is a special state called the *destination* or the *termination state*. We can view the deterministic shortest path problem of Chapter 2 of Vol. I as follows: we want to choose for each node  $i \neq t$ , a successor node  $\mu(i)$  so that  $(i, \mu(i))$  is an arc, and the path formed by a sequence of successor nodes starting at any node  $j$  terminates at  $t$  and has the minimum sum of arc lengths over all paths that start at  $j$  and terminate at  $t$ .

The stochastic shortest path problem is a generalization whereby at each node  $i$ , we must select a probability distribution over all possible successor nodes  $j$  out of a given set of probability distributions  $p_{ij}(u)$  parameterized by a control  $u \in U(i)$ . For a given selection of distributions and for a given origin node, the path traversed as well as its length are now random, but we wish that the path leads to the destination  $t$  with probability one and has minimum expected length. Note that if every feasible probability distribution assigns a probability of 1 to a single successor node, we obtain the deterministic shortest path problem.

We formulate this problem as the special case of the total cost infinite horizon problem where:

- (a) There is no discounting ( $\alpha = 1$ ).
- (b) The state space is  $S = \{1, 2, \dots, n, t\}$  with transition probabilities denoted by

$$p_{ij}(u) = P(x_{k+1} = j \mid x_k = i, u_k = u), \quad i, j \in S, u \in U(i).$$

Furthermore, the destination  $t$  is absorbing, that is, for all  $u \in U(t)$ ,

$$p_{tt}(u) = 1.$$

- (c) The control constraint set  $U(i)$  is a finite set for all  $i$ .
- (d) A cost  $g(i, u)$  is incurred when control  $u \in U(i)$  is selected. Furthermore, the destination is *cost-free*; that is,  $g(t, u) = 0$  for all  $u \in U(t)$ .

Note that as in Section 1.3, we assume that the cost per stage does not depend on  $w$ . This amounts to using expected cost per stage in all calculations. In particular, if the cost of using  $u$  at state  $i$  and moving to state  $j$  is  $\hat{g}(i, u, j)$ , we use as cost per stage the expected cost

$$g(i, u) = \sum_{j=1}^n p_{ij}(u) \hat{g}(i, u, j).$$

We are interested in problems where either reaching the destination is inevitable or else there is an incentive to reach the destination in a finite expected number of stages, so that the essence of the problem is to reach the destination with minimum expected cost. We will be more specific about this shortly.

Note that since the destination is a cost-free and absorbing state, the cost starting from  $t$  is zero for every policy. Accordingly, for all cost functions, we ignore the component that corresponds to  $t$  and define the mappings  $T$  and  $T_\mu$  on functions  $J$  with components  $J(1), \dots, J(n)$ . We will also view the functions  $J$  as  $n$ -dimensional vectors. Thus

$$(TJ)(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J(j) \right], \quad i = 1, \dots, n,$$

$$(T_\mu J)(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J(j), \quad i = 1, \dots, n.$$

As in Section 1.3, for any stationary policy  $\mu$ , we use the compact notation

$$P_\mu = \begin{pmatrix} p_{11}(\mu(1)) & \cdots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \cdots & p_{nn}(\mu(n)) \end{pmatrix},$$

and

$$g_\mu = \begin{pmatrix} g(1, \mu(1)) \\ \vdots \\ g(n, \mu(n)) \end{pmatrix}.$$

We can then write in vector notation

$$T_\mu J = g_\mu + P_\mu J.$$

In terms of this notation, the cost function of a policy  $\pi = \{\mu_0, \mu_1, \dots\}$  can be written as

$$J_\pi = \limsup_{N \rightarrow \infty} T_{\mu_0} \cdots T_{\mu_{N-1}} J_0 = \limsup_{N \rightarrow \infty} \left( g_{\mu_0} + \sum_{k=1}^{N-1} P_{\mu_0} \cdots P_{\mu_{k-1}} g_{\mu_k} \right),$$

where  $J_0$  denotes the zero vector. The cost function of a stationary policy  $\mu$  can be written as

$$J_\mu = \limsup_{N \rightarrow \infty} T_\mu^{N-1} J_0 = \limsup_{N \rightarrow \infty} \sum_{k=0}^{N-1} P_\mu^k g_\mu.$$

The stochastic shortest path problem was discussed in Section 7.2 of Vol. I, under the assumption that all policies lead to the destination with probability 1, regardless of the initial state. In order to analyze the problem under weaker conditions, we introduce the notion of a proper policy.

**Definition 1.1:** A stationary policy  $\mu$  is said to be *proper* if, when using this policy, there is positive probability that the destination will be reached after at most  $n$  stages, regardless of the initial state; that is, if

$$\rho_\mu = \max_{i \in \mathcal{S}, t \in \mathcal{U}(i)} P\{x_n \neq t \mid x_0 = i, \mu\} < 1. \quad (1.1)$$

A stationary policy that is not proper is said to be *improper*.

With a little thought, it can be seen that  $\mu$  is proper if and only if in the Markov chain corresponding to  $\mu$ , each state  $i$  is connected to the destination with a path of positive probability transitions. Note from the definition (1.1) that

$$\begin{aligned} P\{x_{2n} \neq t \mid x_0 = i, \mu\} &= P\{x_{2n} \neq t \mid x_n \neq t, x_0 = i, \mu\} \\ &\quad \times P\{x_n \neq t \mid x_0 = i, \mu\} \\ &\leq \rho_\mu^2. \end{aligned}$$

More generally, for a proper policy  $\mu$ , the probability of not reaching the destination after  $k$  stages diminishes as  $\rho_\mu^{[k/n]}$  regardless of the initial state; that is,

$$P\{x_k \neq t \mid x_0 = i, \mu\} \leq \rho_\mu^{[k/n]}, \quad i = 1, \dots, n. \quad (1.2)$$

Thus the destination will eventually be reached with probability one under a proper policy. Furthermore, the limit defining the associated total cost

vector  $J_\mu$  will exist and be finite, since the expected cost incurred in the  $k$ th period is bounded in absolute value by

$$\rho_\mu^{[k/n]} \max_{i \in \mathcal{S}, u \in \mathcal{U}(i)} |g(i, \mu(i))|. \quad (1.3)$$

Note that, under a proper policy, the cost structure is similar to the one for discounted problems, the main difference being that the effective discount factor depends on the current state and stage, but builds up to at least  $\rho_\mu$  per  $n$  stages.

Throughout this section, we assume the following:

**Assumption 1.1:** There exists at least one proper policy.

**Assumption 1.2:** For every improper policy  $\mu$ , the corresponding cost  $J_\mu(i)$  is  $\infty$  for at least one state  $i$ ; that is, some component of the sum  $\sum_{k=0}^{N-1} P_\mu^k g_\mu$  diverges to  $\infty$  as  $N \rightarrow \infty$ .

In the case of a deterministic shortest path problem, Assumption 1.1 is satisfied if and only if every node is connected to the destination with a path, while Assumption 1.2 is satisfied if and only if each cycle that does not contain the destination has positive length. A simple condition that implies Assumption 1.2 is that the cost  $g(i, u)$  is strictly positive for all  $i \neq t$  and  $u \in U(i)$ . Another important case where Assumptions 1.1 and 1.2 are satisfied is when *all* policies are proper, that is, when termination is inevitable under all stationary policies (this was assumed in Section 7.2 of Vol. I). Actually, for this case, it is possible to show that mappings  $T$  and  $T_\mu$  are contraction mappings with respect to some norm [not necessarily the maximum norm of Eq. (1.1) in Chapter 1]; see Section 4.3 of [BeT89a], or [Tse90]. As a result of this contraction property, the results shown for discounted problems can also be shown for stochastic shortest path problems where termination is inevitable under all stationary policies. It turns out, however, that similar results can be shown even when some improper policies exist: the results that we prove under Assumptions 1.1 and 1.2 are almost as strong as those for discounted problems with bounded cost per stage. In particular, we show that:

- (a) The optimal cost vector is the unique solution of Bellman's equation  

$$J^* = TJ^*.$$
- (b) The value iteration method converges to the optimal cost vector  $J^*$  for an arbitrary starting vector.

- (c) A stationary policy  $\mu$  is optimal if and only if  $T_\mu J^* = TJ^*$ .  
 (d) The policy iteration algorithm yields an optimal proper policy starting from an arbitrary proper policy.

The following proposition provides some basic preliminary results:

**Proposition 1.1:**

- (a) For a proper policy  $\mu$ , the associated cost vector  $J_\mu$  satisfies

$$\lim_{k \rightarrow \infty} (T_\mu^k J)(i) = J_\mu(i), \quad i = 1, \dots, n, \quad (1.4)$$

for every vector  $J$ . Furthermore,

$$J_\mu = T_\mu J_\mu,$$

and  $J_\mu$  is the unique solution of this equation.

- (b) A stationary policy  $\mu$  satisfying for some vector  $J$ ,

$$J(i) \geq (T_\mu J)(i), \quad i = 1, \dots, n,$$

is proper.

**Proof:** (a) Using an induction argument, we have for all  $J \in \mathbb{R}^n$  and  $k \geq 1$

$$T_\mu^k J = P_\mu^k J + \sum_{m=0}^{k-1} P_\mu^m g_\mu. \quad (1.5)$$

Equation (1.2) implies that for all  $J \in \mathbb{R}^n$ , we have

$$\lim_{k \rightarrow \infty} P_\mu^k J = 0,$$

so that

$$\lim_{k \rightarrow \infty} T_\mu^k J = \lim_{k \rightarrow \infty} \sum_{m=0}^{k-1} P_\mu^m g_\mu = J_\mu,$$

where the limit above can be shown to exist using Eq. (1.2).

Also we have by definition

$$T_\mu^{k+1} J = g_\mu + P_\mu T_\mu^k J,$$

and by taking the limit as  $k \rightarrow \infty$ , we obtain

$$J_\mu = g_\mu + P_\mu J_\mu,$$

which is equivalent to  $J_\mu = T_\mu J_\mu$ .

Finally, to show uniqueness, note that if  $J = T_\mu J$ , then we have  $J = T_\mu^k J$  for all  $k$ , so that  $J = \lim_{k \rightarrow \infty} T_\mu^k J = J_\mu$ .

- (b) The hypothesis  $J \geq T_\mu J$ , the monotonicity of  $T_\mu$ , and Eq. (1.5) imply that

$$J \geq T_\mu^k J = P_\mu^k J + \sum_{m=0}^{k-1} P_\mu^m g_\mu, \quad k = 1, 2, \dots$$

If  $\mu$  were not proper, by Assumption 1.2, some component of the sum in the right-hand side of the above relation would diverge to  $\infty$  as  $k \rightarrow \infty$ , which is a contradiction. Q.E.D.

The following proposition is the main result of this section, and provides analogs to the main results for discounted cost problems (Props. 2.1-2.3 in Section 1.2).

**Proposition 1.2:**

- (a) The optimal cost vector  $J^*$  satisfies Bellman's equation

$$J^* = TJ^*.$$

Furthermore,  $J^*$  is the unique solution of this equation.

- (b) We have

$$\lim_{k \rightarrow \infty} (T^k J^*)(i) = J^*(i), \quad i = 1, \dots, n,$$

for every vector  $J$ .

- (c) A stationary policy  $\mu$  is optimal if and only if

$$T_\mu J^* = TJ^*.$$

**Proof:** (a), (b) We first show that  $T$  has at most one fixed point. Indeed, if  $J$  and  $J'$  are two fixed points, then we select  $\mu$  and  $\mu'$  such that  $J = TJ = T_\mu J$  and  $J' = TJ' = T_{\mu'} J'$ ; this is possible because the control constraint set is finite. By Prop. 1.1(b), we have that  $\mu$  and  $\mu'$  are proper, and Prop. 1.1(a) implies that  $J = J_\mu$  and  $J' = J_{\mu'}$ . We have  $J = T^k J \leq T_\mu^k J$  for all  $k \geq 1$ , and by Prop. 1.1(a), we obtain  $J \leq \lim_{k \rightarrow \infty} T_\mu^k J = J_\mu = J'$ . Similarly,  $J' \leq J$ , showing that  $J = J'$  and that  $T$  has at most one fixed point.

We next show that  $T$  has at least one fixed point. Let  $\mu$  be a proper policy (there exists one by Assumption 1.1). Choose  $\mu'$  such that

$$T_{\mu'} J_{\mu} = T J_{\mu}.$$

Then we have  $J_{\mu} = T_{\mu} J_{\mu} \geq T_{\mu'} J_{\mu}$ . By Prop. 1.1(b),  $\mu'$  is proper, and using the monotonicity of  $T_{\mu'}$  and Prop. 1.1(a), we obtain

$$J_{\mu} \geq \lim_{k \rightarrow \infty} T_{\mu'}^k J_{\mu} = J_{\mu}'.$$
 (1.6)

Continuing in the same manner, we construct a sequence  $\{\mu^k\}$  such that each  $\mu^k$  is proper and

$$J_{\mu^k} \geq T J_{\mu^k} \geq J_{\mu^{k+1}}, \quad k = 0, 1, \dots$$
 (1.7)

Since the set of proper policies is finite, some policy  $\mu$  must be repeated within the sequence  $\{\mu^k\}$ , and by Eq. (1.7), we have

$$J_{\mu} = T J_{\mu}.$$

Thus  $J_{\mu}$  is a fixed point of  $T$ , and in view of the uniqueness property shown earlier,  $J_{\mu}$  is the unique fixed point of  $T$ .

Next we show that the unique fixed point of  $T$  is equal to the optimal cost vector  $J^*$ , and that  $T^k J \rightarrow J^*$  for all  $J$ . The construction of the preceding paragraph provides a proper  $\mu$  such that  $T J_{\mu} = J_{\mu}$ . We will show that  $T^k J \rightarrow J_{\mu}$  for all  $J$  and that  $J_{\mu} = J^*$ . Let  $c = (1, 1, \dots, 1)$ , let  $\delta > 0$  be some scalar, and let  $\hat{J}$  be the vector satisfying

$$T_{\mu} \hat{J} = \hat{J} - \delta c,$$

There is a unique such vector because the equation  $\hat{J} = T_{\mu} \hat{J} + \delta c$  can be written as  $\hat{J} = g_{\mu} + \delta c + P_{\mu} \hat{J}$ , so  $\hat{J}$  is the cost vector corresponding to  $\mu$  for  $g_{\mu}$  replaced by  $g_{\mu} + \delta c$ . Since  $\mu$  is proper, by Prop. 1.1(a),  $\hat{J}$  is unique. Furthermore, we have  $J_{\mu} \leq \hat{J}$ , which implies that

$$J_{\mu} = T J_{\mu} \leq T \hat{J} \leq T_{\mu} \hat{J} = \hat{J} - \delta c \leq \hat{J}.$$

Using the monotonicity of  $T$  and the preceding relation, we obtain

$$J_{\mu} = T^k J_{\mu} \leq T^k \hat{J} \leq T^{k-1} \hat{J} \leq \hat{J}, \quad k \geq 1.$$

Hence,  $T^k \hat{J}$  converges to some vector  $\tilde{J}$ , and we have

$$T \tilde{J} = T \left( \lim_{k \rightarrow \infty} T^k \hat{J} \right),$$

The mapping  $T$  can be seen to be continuous, so we can interchange  $T$  with the limit in the preceding relation, thereby obtaining  $\tilde{J} = T \hat{J}$ . By the uniqueness of the fixed point of  $T$  shown earlier, we must have  $\tilde{J} = J_{\mu}$ . It is also seen that

$$J_{\mu} - \delta c = T J_{\mu} - \delta c \leq T(J_{\mu} - \delta c) \leq T J_{\mu} = J_{\mu}.$$

Thus,  $T^k(J_{\mu} - \delta c)$  is monotonically increasing and bounded above. As earlier, it follows that  $\lim_{k \rightarrow \infty} T^k(J_{\mu} - \delta c) = J_{\mu}$ . For any  $J$ , we can find  $\delta > 0$  such that

$$J_{\mu} - \delta c \leq J \leq J.$$

By the monotonicity of  $T$ , we then have

$$T^k(J_{\mu} - \delta c) \leq T^k J \leq T^k J, \quad k \geq 1,$$

and since  $\lim_{k \rightarrow \infty} T^k(J_{\mu} - \delta c) = \lim_{k \rightarrow \infty} T^k J = J_{\mu}$ , it follows that

$$\lim_{k \rightarrow \infty} T^k J = J_{\mu}.$$

To show that  $J_{\mu} = J^*$ , take any policy  $\pi = \{\mu_0, \mu_1, \dots\}$ . We have

$$T_{\mu_0} \cdots T_{\mu_{k-1}} J_0 \geq T^k J_0,$$

where  $J_0$  is the zero vector. Taking the limsup of both sides as  $k \rightarrow \infty$  in the preceding inequality, we obtain

$$J_{\pi} \geq J_{\mu},$$

so  $\mu$  is an optimal stationary policy and  $J_{\mu} = J^*$ .

(c) If  $\mu$  is optimal, then  $J_{\mu} = J^*$  and, by Assumptions 1.1 and 1.2,  $\mu$  is proper, so by Prop. 1.1(a),  $T_{\mu} J^* = T_{\mu} J_{\mu} = J_{\mu} = J^* = T J^*$ . Conversely, if  $J^* = T J^* = T_{\mu} J^*$ , it follows from Prop. 1.1(b) that  $\mu$  is proper, and by using Prop. 1.1(a), we obtain  $J^* = J_{\mu}$ . Therefore,  $\mu$  is optimal. Q.E.D.

The results of Prop. 1.2 can also be proved (with minor changes) assuming, in place of Assumption 1.2, that  $g(i, u) \geq 0$  for all  $i$  and  $u \in U(i)$ , and that there exists an optimal proper policy; see Exercise 2.12.

### Compact Control Constraint Sets

It turns out that the finiteness assumption on the control constraint  $U(i)$  can be weakened. It is sufficient that, for each  $i$ ,  $U(i)$  be a compact subset of a Euclidean space, and that  $p_{ij}(u)$  and  $g(i, u)$  be continuous in  $u$  over  $U(i)$ , for all  $i$  and  $j$ . Under these compactness and continuity assumptions, and also Assumptions 1.1 and 1.2, Prop. 1.2 holds as stated. The proof is similar to the one given above, but is technically much more complex. It can be found in [BeT91b].

## Underlying Contractions

We mentioned in Section 1.4 that the strong results we derived for discounted problems in Chapter 1 owe their validity to the contraction property of the mapping  $T$ . Despite the similarity of Prop. 1.2 with the corresponding discounted cost results of Section 1.2, under Assumptions 1.1 and 1.2, the mapping  $T$  of this section need not be a contraction mapping with respect to any norm; see Exercise 2.13 for a counterexample. On the other hand there is an important special case where  $T$  is a contraction mapping with respect to a *weighted sup norm*. In particular, it can be shown that if all stationary policies are proper, then there exist positive constants  $v_1, \dots, v_n$  and some  $\gamma$  with  $0 \leq \gamma < 1$ , such that we have for all vectors  $J_1$  and  $J_2$ ,

$$\max_{i=1,\dots,n} \frac{1}{v_i} |(TJ_1)(i) - (TJ_2)(i)| \leq \gamma \max_{i=1,\dots,n} \frac{1}{v_i} |J_1(i) - J_2(i)|.$$

A proof of this fact is outlined in Exercise 2.14.

## Pathologies of Stochastic Shortest Path Problems

We now give two examples that illustrate the sensitivity of our results to seemingly minor changes in our assumptions.

### Example 1.1 (The Blackmailer's Dilemma [Whi82])

This example shows that the assumption of a finite or compact control constraint set cannot be easily relaxed. Here, there are two states, state 1 and the destination state  $t$ . At state 1, we can choose a control  $u$  with  $0 < u \leq 1$ ; we then move to state  $t$  at no cost with probability  $u^2$ , and stay in state 1 at a cost  $-u$  with probability  $1 - u^2$ . Note that every stationary policy is proper in the sense that it leads to the destination with probability one.

We may regard  $u$  as a demand made by a blackmailer, and state 1 as the situation where the victim complies. State  $t$  is the situation where the victim refuses to yield to the blackmailer's demand. The problem then can be seen as one whereby the blackmailer tries to maximize his total gain by balancing his desire for increased demands with keeping his victim compliant.

If controls were chosen from a *finite* subset of the interval  $(0, 1]$ , the problem would come under the framework of this section. The optimal cost would then be finite, and there would exist an optimal stationary policy. It turns out, however, that without the finiteness restriction the optimal cost starting at state 1 is  $-\infty$  and there exists no optimal stationary policy. Indeed, for any stationary policy  $\mu$  with  $\mu(1) = u$ , we have

$$J_\mu(1) = -u + (1 - u^2)J_\mu(1)$$

from which

$$J_\mu(1) = -\frac{1}{u},$$

Therefore,  $\min_\mu J_\mu(1) = -\infty$  and  $J^*(1) = -\infty$ , but there is no stationary policy that achieves the optimal cost. Note also that this situation would not change if the constraint set were  $u \in [0, 1]$  (i.e.,  $u = 0$  were an allowable control), although in this case the stationary policy that applies  $\mu(1) = 0$  is improper and its corresponding cost vector is zero, thus violating Assumption 1.2.

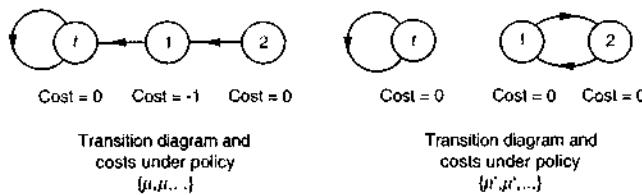
An interesting fact about this problem is that there is an optimal *non-stationary* policy  $\pi$  for which  $J_\pi(1) = -\infty$ . This is the policy  $\pi = \{\mu_0, \mu_1, \dots\}$  that applies  $\mu_k(1) = \gamma/(k+1)$  at time  $k$  and state 1, where  $\gamma$  is a scalar in the interval  $(0, 1/2)$ . We leave the verification of this fact to the reader. What happens with the policy  $\pi$  is that the blackmailer requests diminishing amounts over time, which nonetheless add to  $\infty$ . However, the probability of the victim's refusal diminishes at a much faster rate over time, and as a result, the probability of the victim remaining compliant forever is strictly positive, leading to an infinite total expected payoff to the blackmailer.

## Example 1.2 (Pure Stopping Problems)

This example illustrates why we need to assume that all improper policies have infinite cost for at least some initial state (Assumption 1.2). Consider an optimal stopping problem where a state-dependent cost is incurred only when invoking a stopping action that drives the system to the destination; all costs are zero prior to stopping. Eventual stopping is a requirement here, so to properly formulate such a stopping problem as a total cost infinite horizon problem, it is essential to make the stopping costs negative (by adding a negative constant to all stopping costs if necessary), providing an incentive to stop. We then come under the framework of this section but with Assumption 1.2 violated because the improper policy that never stops does not yield infinite cost for any starting state. Unfortunately, this seemingly small relaxation of our assumptions invalidates our results as shown by the example of Fig. 2.1.1. This example is in effect a deterministic shortest path problem involving a cycle with zero length. In particular, in the example there is a (nonoptimal) improper policy that yields finite cost for all initial states (rather than infinite cost for some initial state), and  $T$  has multiple fixed points.

## 2.2 COMPUTATIONAL METHODS

All the methods developed in connection with the discounted cost problem in Section 1.3, have stochastic shortest path analogs. For example, value iteration works as shown by Prop. 1.2(b). Furthermore, the (exact and approximate) linear programming approach also has a straightforward extension (cf. Section 1.3.4), since  $J^*$  is the largest solution of the system of inequalities  $J \leq TJ$ . In this section, we will discuss in more detail some



**Figure 2.1.1** Example where Prop. 1.2 fails to hold when Assumption 1.2 is violated. There are two stationary policies,  $\mu$  and  $\mu'$ , with transition probabilities and costs as shown. The equation  $J = T J$  is given by

$$J(1) = \min\{-1, J(2)\},$$

$$J(2) = J(1),$$

and is satisfied by any  $J$  of the form

$$J(1) = \delta, \quad J(2) = \delta,$$

with  $\delta \leq -1$ . Here the proper policy  $\mu$  is optimal and the corresponding optimal cost vector is

$$J(1) = -1, \quad J(2) = -1.$$

The difficulty is that the improper policy  $\mu'$  has finite (zero) cost for all initial states.

of the major methods, and we will also focus on some stochastic shortest path problems with special structure. It turns out that by exploiting this special structure, we can improve the convergence properties of some of the methods. For example, in deterministic shortest path problems, value iteration terminates finitely (Section 2.1 of Vol. I), whereas this does not happen for any significant class of discounted cost problems.

### 2.2.1 Value Iteration

As shown by Prop. 1.2(b), value iteration works for stochastic shortest path problems. Furthermore, several of the enhancements and variations of value iteration for discounted problems have stochastic shortest path analogs. In particular, there are error bounds similar to the ones of Prop. 3.1 in Section 1.3 (although not quite as powerful; see Section 7.2 of Vol. I). It can also be shown that the Gauss-Seidel version of the method works and that its rate of convergence is typically faster than that of the ordinary method (Exercise 2.6). Furthermore, the rank-one correction method described in Section 1.3.1 is straightforward and effective, as long as there is some separation between the dominant and the subdominant eigenvalue moduli.

### Finite Termination of Value Iteration

Generally, the value iteration method requires an infinite number of iterations in stochastic shortest path problems. However, under special circumstances, the method can terminate finitely. A prominent example is the case of a deterministic shortest path problem, but there are other more general circumstances where termination occurs. In particular, let us assume that the transition probability graph corresponding to some optimal stationary policy  $\mu^*$  is acyclic. By this we mean that there are no cycles in the graph that has as nodes the states  $1, \dots, n, t$ , and has an arc  $(i, j)$  for each pair of states  $i$  and  $j$  such that  $p_{ij}(\mu^*(i)) > 0$ . We assume in particular that there are no positive self-transition probabilities  $p_{ii}(\mu^*(i))$  for  $i \neq t$ , but it turns out that under Assumptions 1.1 and 1.2, a stochastic shortest path problem with such self-transitions can be converted into another stochastic shortest path problem where  $p_{ii}(u) = 0$  for all  $i \neq t$  and  $u \in U(i)$ . In particular, it can be shown (Exercise 2.8) that the modified stochastic shortest path problem that has costs

$$\tilde{g}(i, u) = g(i, u) + \frac{g(i, u)p_{ii}(u)}{1 - p_{ii}(u)}, \quad i = 1, \dots, n,$$

in place of  $g(i, u)$ , and transition probabilities

$$\tilde{p}_{ij}(u) = \begin{cases} 0 & \text{if } j = i, \\ \frac{p_{ij}(u)}{1 - p_{ii}(u)} & \text{if } j \neq i, \end{cases} \quad i = 1, \dots, n,$$

instead of  $p_{ij}(u)$  is equivalent to the original in the sense that it has the same optimal costs and policies.

We claim that, under the preceding acyclicity assumption, the value iteration method will yield  $J^*$  after at most  $n$  iterations when started from the vector  $J$  given by

$$J(i) = \infty, \quad i = 1, \dots, n. \quad (2.1)$$

To show this, consider the sets of states  $S_0, S_1, \dots$  defined by

$$S_0 = \{t\}, \quad (2.2)$$

$$S_{k+1} = \{i \mid p_{ij}(\mu^*(i)) = 0 \text{ for all } j \notin \cup_{m=0}^k S_m\}, \quad k = 0, 1, \dots, \quad (2.3)$$

and let  $S_{\bar{k}}$  be the last of these sets that is nonempty. Then in view of our acyclicity assumption, we have

$$\cup_{m=0}^{\bar{k}} S_m = \{1, \dots, n, t\}. \quad (2.4)$$

Let us show by induction that, starting from the vector  $J$  of Eq. (2.1), the value iteration method will yield for  $k = 0, 1, \dots, \bar{k}$ ,

$$(T^k J)(i) = J^*(i), \quad \text{for all } i \in \cup_{m=0}^{\bar{k}} S_m, i \neq t.$$

Indeed, this is so for  $k = 0$ . Assume that  $(T^k J)(i) = J^*(i)$  if  $i \in \cup_{m=0}^k S_m$ . Then, by the monotonicity of  $T$ , we have for all  $i$ ,

$$J^*(i) \leq (T^{k+1} J)(i),$$

while we have by the induction hypothesis, the definition of the sets  $S_k$ , and the optimality of  $\mu^*$ ,

$$\begin{aligned} (T^{k+1} J)(i) &\leq g(i, \mu^*(i)) + \sum_{j \in \cup_{m=0}^k S_m} p_{ij}(\mu^*(i)) J^*(j) \\ &= J^*(i), \quad \text{for all } i \in \cup_{m=0}^{k+1} S_m, i \neq t. \end{aligned}$$

The last two relations complete the induction.

Thus, we have shown that under the acyclicity assumption, at the  $k$ th iteration, the value iteration method, will set to the optimal values the costs of states in the set  $S_k$ . In particular, all optimal costs will be found after  $k$  iterations.

### Consistently Improving Policies

The properties of value iteration can be further improved if there is an optimal policy  $\mu^*$  under which from a given state, we can only go to a state of lower cost; that is, for all  $i$ , we have

$$p_{ij}(\mu^*(i)) > 0 \quad \Rightarrow \quad J^*(i) > J^*(j).$$

We call such a policy *consistently improving*.

A case where a consistently improving policy exists arises in deterministic shortest-path problems when all the arc lengths are positive. Another important case arises in continuous-space shortest-path problems; see [TsI93a] and Exercise 2.10.

The transition probability graph corresponding to a consistently improving policy is seen to be acyclic, so when such a policy exists, by the preceding discussion, the value iteration method terminates finitely. However, a stronger property can be proved. As discussed in Chapter 2 of Vol. I, for shortest-path problems with positive arc lengths, one can use Dijkstra's algorithm. This is the label correcting method, which removes from the OPEN list a node with minimum label at each iteration and requires just one iteration per node. A similar property holds for stochastic shortest path problems if there is a consistently improving policy: if one removes from the OPEN list a state  $j$  with minimum cost estimate  $J(j)$ , the Gauss-Seidel version of the value iteration method requires just one iteration per state; see Exercise 2.11.

For problems where a consistently improving policy exists, it is also appropriate to use straightforward adaptations of the label correcting shortest path methods discussed in Section 2.3.1 of Vol. I. In particular, one may approximate the policy of removing from the OPEN list a minimum cost state by using the SLF and LLL strategies (see [PBT95]).

### 2.2.2 Policy Iteration

The policy iteration algorithm is based on the construction used in the proof of Prop. 1.2 to show that  $T$  has a fixed point. In the typical iteration, given a proper policy  $\mu$  and the corresponding cost vector  $J_\mu$ , one obtains a new proper policy  $\bar{\mu}$  satisfying  $T_{\bar{\mu}} J_\mu = TJ_\mu$ . It was shown in Eq. (1.6) that  $J_{\bar{\mu}} \leq J_\mu$ . It can be seen also that strict inequality  $J_{\bar{\mu}}(i) < J_\mu(i)$  holds for at least one state  $i$ , if  $\mu$  is nonoptimal; otherwise we would have  $J_\mu = TJ_\mu$  and by Prop. 1.2(c),  $\mu$  would be optimal. Therefore, the new policy is strictly better if the current policy is nonoptimal. Since the number of proper policies is finite, the policy iteration algorithm terminates after a finite number of iterations with an optimal proper policy.

It is possible to execute approximately the policy evaluation step of policy iteration, using a finite number of value iterations, as in the discounted case. Here we start with some vector  $J_0$ . For all  $k$ , a stationary policy  $\mu^k$  is defined from  $J_k$  according to  $T_{\mu^k} J_k = TJ_k$ , the cost  $J_{\mu^k}$  is approximately evaluated by  $m_k - 1$  additional value iterations, yielding the vector  $J_{k+1}$ , which is used in turn to define  $\mu^{k+1}$ . The proof of Prop. 3.5 in Section 1.3 can be essentially repeated to show that  $J_k \rightarrow J^*$ , assuming that the initial vector  $J_0$  satisfies  $TJ_0 \leq J_0$ . Unfortunately, the requirement  $TJ_0 \leq J_0$  is essential for the convergence proof, unless all stationary policies are proper, in which case  $T$  is a contraction mapping (cf. Exercise 2.14).

As in Section 1.3.3, it is possible to use adaptive aggregation in conjunction with approximate policy evaluation. However, it is important that the destination  $t$  forms by itself an aggregate state, which will play the role of the destination in the aggregate Markov chain.

### Approximate Policy Iteration

Let us consider an approximate policy iteration algorithm that generates a sequence of stationary policies  $\{\mu^k\}$  and a corresponding sequence of approximate cost vectors  $\{J_k\}$  satisfying

$$\max_{i=1,\dots,n} |J_k(i) - J_{\mu^k}(i)| \leq \delta, \quad k = 0, 1, \dots \quad (2.5)$$

and

$$\max_{i=1,\dots,n} |(T_{\mu^{k+1}} J_k)(i) - (T J_k)(i)| \leq \epsilon, \quad k = 0, 1, \dots \quad (2.6)$$

where  $\delta$  and  $\epsilon$  are some positive scalars, and  $\mu^0$  is some proper policy. One difficulty with such an algorithm is that, even if the current policy  $\mu^k$  is proper, the next policy  $\mu^{k+1}$  may not be proper. In this case, we have  $J_{\mu^{k+1}}(i) = \infty$  for some  $i$ , and the method breaks down. Note, however, that for a sufficiently small  $\epsilon$ , Eq. (2.6) implies that  $T_{\mu^{k+1}} J_k = T J_k$ , so by Prop. 1.1(b),  $\mu^{k+1}$  will be proper. In any case, we will analyze the method

under the assumption that all generated policies are proper. The following proposition parallels Prop. 3.6 in Section 1.3. It provides an estimate of the difference  $J_{\mu^k} - J^*$  in terms of the scalar  $\rho$ .

$$\rho = \max_{\substack{i=1,\dots,n \\ \mu \text{ proper}}} P\{x_n \neq t \mid x_0 = i, \mu\}.$$

Note that for every proper policy  $\mu$  and state  $i$ , we have  $P\{x_n \neq t \mid x_0 = i, \mu\} < 1$  by the definition of a proper policy, and since the number of proper policies is finite, we have  $\rho < 1$ .

**Proposition 2.1:** Assume that the stationary policies  $\mu^k$  generated by the approximate policy iteration algorithm are all proper. Then

$$\limsup_{k \rightarrow \infty} \max_{i=1,\dots,n} (J_{\mu^k}(i) - J^*(i)) \leq \frac{n(1-\rho+n)(\epsilon+2\delta)}{(1-\rho)^2}. \quad (2.7)$$

**Proof:** The proof is similar to the one of Prop. 3.6 in Section 1.3. We modify the arguments in order to use the relations  $T_\mu(J + rc) \leq T_\mu J + rc$  and  $P_\mu^n c \leq \rho c$ , which hold for all proper policies  $\mu$  and positive scalars  $r$ . We use Eqs. (2.5) and (2.6) to obtain for all  $k$

$$T_{\mu^{k+1}} J_{\mu^k} \leq T J_{\mu^k} + (\epsilon+2\delta)c \leq T_{\mu^k} J_{\mu^k} + (\epsilon+2\delta)c. \quad (2.8)$$

From Eq. (2.8) and the equation  $T_{\mu^k} J_{\mu^k} = J_{\mu^k}$ , we have

$$T_{\mu^{k+1}} J_{\mu^k} \leq J_{\mu^k} + (\epsilon+2\delta)c.$$

By subtracting from this relation the equation  $T_{\mu^{k+1}} J_{\mu^{k+1}} = J_{\mu^{k+1}}$ , we obtain

$$T_{\mu^{k+1}} J_{\mu^k} - T_{\mu^{k+1}} J_{\mu^{k+1}} \leq J_{\mu^k} - J_{\mu^{k+1}} + (\epsilon+2\delta)c.$$

This relation can be written as

$$J_{\mu^{k+1}} - J_{\mu^k} \leq P_{\mu^{k+1}}(J_{\mu^{k+1}} - J_{\mu^k}) + (\epsilon+2\delta)c, \quad (2.9)$$

where  $P_{\mu^{k+1}}$  is the transition probability matrix corresponding to  $\mu^{k+1}$ . Let

$$\xi_k = \max_{i=1,\dots,n} (J_{\mu^{k+1}}(i) - J_{\mu^k}(i)).$$

Then Eq. (2.9) yields

$$\xi_k c \leq \xi_k P_{\mu^{k+1}} c + (\epsilon+2\delta)c.$$

By multiplying this relation with  $P_{\mu^{k+1}}$  and by adding  $(\epsilon+2\delta)c$ , we obtain

$$\xi_k c \leq \xi_k P_{\mu^{k+1}}^2 c + (\epsilon+2\delta)c \leq \xi_k P_{\mu^{k+1}}^2 c + 2(\epsilon+2\delta)c.$$

By repeating this process for a total of  $n-1$  times, we have

$$\xi_k c \leq \xi_k P_{\mu^{k+1}}^n c + n(\epsilon+2\delta)c \leq \rho \xi_k c + n(\epsilon+2\delta)c.$$

Thus,

$$\xi_k \leq \frac{n(\epsilon+2\delta)}{1-\rho}. \quad (2.10)$$

Let  $\mu^*$  be an optimal stationary policy. From Eq. (2.8), we have

$$\begin{aligned} T_{\mu^{k+1}} J_{\mu^k} &\leq T_{\mu^*} J_{\mu^k} + (\epsilon+2\delta)c \\ &= T_{\mu^*} J_{\mu^k} - T_{\mu^*} J^* + J^* + (\epsilon+2\delta)c \\ &= P_{\mu^*}(J_{\mu^k} - J^*) + J^* + (\epsilon+2\delta)c. \end{aligned}$$

We also have

$$T_{\mu^{k+1}} J_{\mu^k} = J_{\mu^{k+1}} + T_{\mu^{k+1}} J_{\mu^k} - T_{\mu^{k+1}} J_{\mu^{k+1}} = J_{\mu^{k+1}} + P_{\mu^{k+1}}(J_{\mu^k} - J_{\mu^{k+1}}).$$

By subtracting the last two relations, and by using the definition of  $\xi_k$  and Eq. (2.10), we obtain

$$\begin{aligned} J_{\mu^{k+1}} - J^* &\leq P_{\mu^*}(J_{\mu^k} - J^*) + P_{\mu^{k+1}}(J_{\mu^{k+1}} - J_{\mu^k}) + (\epsilon+2\delta)c \\ &\leq P_{\mu^*}(J_{\mu^k} - J^*) + \xi_k P_{\mu^{k+1}} c + (\epsilon+2\delta)c \\ &\leq P_{\mu^*}(J_{\mu^k} - J^*) + \xi_k c + (\epsilon+2\delta)c \\ &\leq P_{\mu^*}(J_{\mu^k} - J^*) + \frac{(1-\rho+n)(\epsilon+2\delta)}{1-\rho}c. \end{aligned} \quad (2.11)$$

Let

$$\zeta_k = \max_{i=1,\dots,n} (J_{\mu^k}(i) - J^*(i)).$$

Then Eq. (2.11) yields, for all  $k$ ,

$$\zeta_{k+1} c \leq \zeta_k P_{\mu^*} c + \frac{(1-\rho+n)(\epsilon+2\delta)}{1-\rho}c.$$

By multiplying this relation with  $P_{\mu^*}$  and by adding  $(1-\rho+n)(\epsilon+2\delta)c/(1-\rho)$ , we obtain

$$\zeta_{k+2} c \leq \zeta_{k+1} P_{\mu^*} c + \frac{(1-\rho+n)(\epsilon+2\delta)}{1-\rho}c \leq \zeta_k P_{\mu^*}^2 c + \frac{2(1-\rho+n)(\epsilon+2\delta)}{1-\rho}c.$$

By repeating this process for a total of  $n - 1$  times, we have

$$\zeta_{k+n} c \leq \zeta_k P_\mu^n c + \frac{n(1-\rho+n)(\epsilon+2\delta)}{1-\rho} c \leq \rho \zeta_k c + \frac{n(1-\rho+n)(\epsilon+2\delta)}{1-\rho} c.$$

By taking the limit superior as  $k \rightarrow \infty$ , we obtain

$$(1-\rho) \limsup_{k \rightarrow \infty} \zeta_k \leq \frac{n(1-\rho+n)(\epsilon+2\delta)}{1-\rho},$$

which was to be proved. **Q.E.D.**

The error bound (2.7) uses the worst-case estimate of the number of stages required to reach  $t$  with positive probability, which is  $n$ . We can strengthen the error bound if we have a better estimate. In particular, for all  $m \geq 1$ , let

$$\rho_m = \max_{\substack{i=1, \dots, n \\ \mu \text{ proper}}} P\{x_m \neq t \mid x_0 = i, \mu\},$$

and let  $\bar{m}$  be the minimal  $m$  for which  $\rho_m < 1$ . Then the proof of Prop. 2.1 can be adapted to show that

$$\limsup_{k \rightarrow \infty} \max_{i=1, \dots, n} (J_{\mu^k}(i) - J^*(i)) \leq \frac{\bar{m}(1-\rho_{\bar{m}}+\bar{m})(\epsilon+2\delta)}{(1-\rho_{\bar{m}})^2}.$$

### 2.3 SIMULATION-BASED METHODS

The computational methods described so far apply when there is a mathematical model of the cost structure and the transition probabilities of the system. In many problems, however, such a model is not available, but instead, the system and cost structure can be simulated. By this we mean that the state space and the control space are known, and there is a computer program that simulates, for a given control  $u$ , the probabilistic transitions from any given state  $i$  to a successor state  $j$  according to the transition probabilities  $p_{ij}(u)$ , and also generates a corresponding transition cost  $g(i, u, j)$ . It is then of course possible to use repeated simulation to calculate (at least approximately) the transition probabilities of the system and the expected stage costs by averaging, and then to apply the methods discussed earlier.

The methodology discussed in this section, however, is geared towards an alternative possibility, which is much more attractive when one is faced with a large and complex system, and one contemplates approximations: rather than estimate explicitly the transition probabilities and costs, we

estimate the cost function of a given policy by generating a number of simulated system trajectories and associated costs, and by using some form of "least-squares fit."

Within this context, there are a number of possible approximation techniques, which for the most part are patterned after the value and the policy iteration methods. We focus first on exact methods where estimates of various cost functions are maintained in a "look-up table" that contains one entry per state. We later develop approximate methods where cost functions are maintained in a "compact" form; that is, they are represented by a function chosen from a parametric class, perhaps involving a feature extraction mapping or a neural network. We first consider these methods for the stochastic shortest path problem, and we later adapt them for the discounted cost problem in Section 2.3.4.

To make the notation better suited for the simulation context, we make a slight change in the problem definition. In particular, instead of considering the expected cost  $g(i, u)$  at state  $i$  under control  $u$ , we allow the cost  $g$  to depend on the next state  $j$ . Thus our notation for the cost per stage is now  $g(i, u, j)$ . All the results and the entire analysis of the preceding sections can be rewritten in terms of the new notation by replacing  $g(i, u)$  with  $\sum_{j=1}^n p_{ij}(u)g(i, u, j)$ .

#### 2.3.1 Policy Evaluation by Monte-Carlo Simulation

Consider the stochastic shortest path problem of Section 2.1. Suppose that we are given a *fixed* stationary policy  $\mu$  and we want to calculate by simulation the corresponding cost vector  $J_\mu$ . One possibility is of course to generate, starting from each  $i$ , many sample state trajectories and average the corresponding costs to obtain an approximation to  $J_\mu(i)$ . We can do this separately for each possible initial state, but a more efficient method is to use each trajectory to obtain a cost sample for many states by considering the costs of the trajectory portions that start at these states. If a state is encountered multiple times within the same trajectory, the corresponding cost samples can be treated as multiple independent samples for that state.<sup>†</sup>

To simplify notation, in what follows we do not show the dependence of various quantities on the given policy. In particular, the transition probability from  $i$  to  $j$ , and the corresponding stage cost are denoted by  $p_{ij}$  and  $g(i, j)$ , in place of  $p_{ij}(\mu(i))$  and  $g(i, \mu(i), j)$ , respectively.

To formalize the process, suppose that we perform an infinite number of simulation runs, each ending at the termination state  $t$ . Assume also that within the total number of runs, each state is encountered an infinite

<sup>†</sup> The validity of doing so is not quite obvious because in the case of multiple visits to the same state within the same trajectory, the corresponding multiple cost samples are correlated, since portions of the corresponding cost sequence are shared by these cost samples. For a justifying analysis, see Exercise 2.15.

number of times. Consider the  $m$ th time a given state  $i$  is encountered, and let  $(i, i_1, i_2, \dots, i_N, t)$  be the remainder of the corresponding trajectory. Let  $c(i, m)$  be the corresponding cost of reaching state  $t$ .

$$c(i, m) = g(i, i_1) + g(i_1, i_2) + \dots + g(i_N, t).$$

We assume that the simulations correctly average the desired quantities; that is, for all states  $i$ , we have

$$J_\mu(i) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M c(i, m). \quad (3.1)$$

We can iteratively calculate the sums appearing in the above equations by using the update formulas

$$J_\mu(i) := J_\mu(i) + \gamma_m (c(i, m) - J_\mu(i)), \quad m = 1, 2, \dots,$$

where

$$\gamma_m = \frac{1}{m}, \quad m = 1, 2, \dots,$$

and the initial conditions are, for all  $i$ ,

$$J_\mu(i) = 0.$$

The normal way to implement the preceding algorithm is to update the costs  $J_\mu(i)$  at the end of each simulation run that generates the state trajectory  $(i_1, i_2, \dots, i_N, t)$ , by using for each  $k = 1, \dots, N$ , the formula

$$J_\mu(i_k) := J_\mu(i_k) + \gamma_{m_k} (g(i_k, i_{k+1}) + g(i_{k+1}, i_{k+2}) + \dots + g(i_N, t) - J_\mu(i_k)), \quad (3.2)$$

where  $m_k$  is the number of visits thus far to state  $i_k$  and  $\gamma_{m_k} = 1/m_k$ . There are also forms of the law of large numbers, which allow the use of a different stepsize  $\gamma_{m_k}$  in the above equation. It can be shown that for convergence of iteration (3.2) to the correct cost value  $J_\mu(i_k)$ , it is sufficient that  $\gamma_{m_k}$  be diminishing at the rate of one over the number of visits to state  $i_k$ .

### Monte-Carlo Simulation Using Temporal Differences

An alternative (and essentially equivalent) method to implement the Monte-Carlo simulation update (3.2), is to update  $J_\mu(i_1)$  immediately after  $g(i_1, i_2)$  and  $i_2$  are generated, then update  $J_\mu(i_1)$  and  $J_\mu(i_2)$  immediately after  $g(i_2, i_3)$  and  $i_3$  are generated, and so on. The method uses the quantities

$$d_k = g(i_k, i_{k+1}) + J_\mu(i_{k+1}) - J_\mu(i_k), \quad k = 1, \dots, N, \quad (3.3)$$

with  $i_{N+1} = t$ , which are called *temporal differences*. They represent the difference between the current estimate  $J_\mu(i_k)$  of *expected cost* to go to the termination state and the *predicted cost* to go to the termination state,

$$g(i_k, i_{k+1}) + J_\mu(i_{k+1}),$$

based on the simulated outcome of the current stage. Given a sample state trajectory  $(i_1, i_2, \dots, i_N, t)$ , the cost update formula (3.2) can be rewritten in terms of the temporal differences  $d_k$  as follows [to see this, just add the formulas below and use the fact  $J_\mu(i_{N+1}) = J_\mu(t) = 0$ ]:

Following the state transition  $(i_1, i_2)$ , set

$$J_\mu(i_1) := J_\mu(i_1) + \gamma_{m_1} d_1 = J_\mu(i_1) + \gamma_{m_1} (g(i_1, i_2) + J_\mu(i_2) - J_\mu(i_1)).$$

Following the state transition  $(i_2, i_3)$ , set

$$J_\mu(i_2) := J_\mu(i_2) + \gamma_{m_2} d_2 = J_\mu(i_2) + \gamma_{m_2} (g(i_2, i_3) + J_\mu(i_3) - J_\mu(i_2)).$$

$$J_\mu(i_3) := J_\mu(i_3) + \gamma_{m_3} d_3 = J_\mu(i_3) + \gamma_{m_3} (g(i_3, i_4) + J_\mu(i_4) - J_\mu(i_3)),$$

⋮ ⋮ ⋮

Following the state transition  $(i_N, t)$ , set

$$J_\mu(i_1) := J_\mu(i_1) + \gamma_{m_1} d_N = J_\mu(i_1) + \gamma_{m_1} (g(i_N, t) - J_\mu(i_N)),$$

$$J_\mu(i_2) := J_\mu(i_2) + \gamma_{m_2} d_N = J_\mu(i_2) + \gamma_{m_2} (g(i_N, t) - J_\mu(i_N)),$$

⋮ ⋮ ⋮

$$J_\mu(i_N) := J_\mu(i_N) + \gamma_{m_N} d_N = J_\mu(i_N) + \gamma_{m_N} (g(i_N, t) - J_\mu(i_N)).$$

The stepsizes  $\gamma_{m_k}$ ,  $k = 1, \dots, N$ , are given by  $\gamma_{m_k} = 1/m_k$ , where  $m_k$  is the number of visits already made to state  $i_k$ . In the case where the sample trajectory involves at most one visit to each state, the preceding updates are equivalent to the update (3.2). If there are multiple visits to some state during a sample trajectory, there is a difference between the preceding updates and the update (3.2), because the updates corresponding to each visit to the given state affect the updates corresponding to subsequent visits to the same state. However, this is an effect which is of second order in the stepsize  $\gamma$ , so once  $\gamma$  becomes small, the difference is negligible.

### TD( $\lambda$ )

The preceding implementation of the Monte-Carlo simulation method for evaluating the cost of a policy  $\mu$  is known as TD(1) (here TD stands for Temporal Differences). A generalization of TD(1) is TD( $\lambda$ ), where  $\lambda$  is a parameter with

$$0 \leq \lambda \leq 1.$$

Given a sample trajectory  $(i_1, \dots, i_N, t)$  with a corresponding cost sequence  $g(i_1, i_2), \dots, g(i_N, t)$ , TD( $\lambda$ ) updates the cost estimates  $J_\mu(i_1), \dots, J_\mu(i_N)$  using the temporal differences

$$d_k = g(i_k, i_{k+1}) + J_\mu(i_{k+1}) - J_\mu(i_k), \quad k = 1, \dots, N,$$

and the equations

$$J_\mu(i_1) := J_\mu(i_1) + \gamma_{m_1} d_1, \quad \text{following the transition } (i_1, i_2).$$

$$\begin{cases} J_\mu(i_1) := J_\mu(i_1) + \gamma_{m_1} \lambda d_2, \\ J_\mu(i_2) := J_\mu(i_2) + \gamma_{m_2} d_2, \end{cases} \quad \text{following the transition } (i_2, i_3),$$

and more generally for  $k = 1, \dots, N$ ,

$$\begin{cases} J_\mu(i_1) := J_\mu(i_1) + \gamma_{m_1} \lambda^{k-1} d_k, \\ J_\mu(i_2) := J_\mu(i_2) + \gamma_{m_2} \lambda^{k-2} d_k, \\ \dots \\ J_\mu(i_k) := J_\mu(i_k) + \gamma_{m_k} d_k, \end{cases} \quad \text{following the transition } (i_k, i_{k+1}).$$

The use of a value of  $\lambda$  less than 1 tends to discount the effect of the temporal differences of state transitions far into the future on the cost estimate of the current state. In the case where  $\lambda = 0$ , we obtain the TD(0) algorithm, which following the transition  $(i_k, i_{k+1})$  updates  $J_\mu(i_k)$  by

$$J_\mu(i_k) := J_\mu(i_k) + \gamma_{m_k} (g(i_k, i_{k+1}) + J_\mu(i_{k+1}) - J_\mu(i_k)). \quad (3.4)$$

This algorithm is a special case of an important stochastic iterative algorithm known as the *stochastic approximation* (or *Robbins-Monro*) method (see e.g., [BMP90], [BeT89a], [LiS83]) for solving Bellman's equations

$$E\{g(i_k, i_{k+1}) + J_\mu(i_{k+1})\} - J_\mu(i_k) = 0.$$

In this algorithm, the expected value above is approximated by using a single sample at each iteration [cf. Eq. (3.4)].

The stepsizes  $\gamma_{m_k}$  need not be equal to  $1/m_k$ , where  $m_k$  is the number of visits thus far to state  $i_k$ , but they should diminish to zero with the number of visits to each state. For example one may use the same stepsize  $\gamma_m = 1/m$  for all states within the  $m$ th simulation trajectory. With such

a stepsize and under some technical conditions, chief of which is that each state  $i = 1, \dots, n$  is visited infinitely often in the course of the simulation, it can be shown that for all  $\lambda \in [0, 1]$ , the cost estimates  $J_\mu(i)$  generated by TD( $\lambda$ ) converge to the correct values with probability 1.

While TD( $\lambda$ ) yields the correct values of  $J_\mu(i)$  in the limit regardless of the value of  $\lambda$ , the choice of  $\lambda$  may have a substantial effect on the rate of convergence. Some experience suggests that using  $\lambda < 1$  (rather than  $\lambda = 1$ ), often reduces the number of sample trajectories needed to attain the same variance of error between  $J_\mu(i)$  and its estimate. However, at present there is no analysis relating to this phenomenon.

### Simulation-Based Policy Iteration

The policy evaluation procedures discussed above can be embedded within a simulation-based policy iteration approach. Let us introduce the notion of the *Q-factor* of a state-control pair  $(i, u)$  and a stationary policy  $\mu$ , defined as

$$Q_\mu(i, u) = \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + J_\mu(j)). \quad (3.5)$$

It is the expected cost corresponding to starting at state  $i$ , using control  $u$  at the first stage, and using the stationary policy  $\mu$  at the second and subsequent stages.

The *Q*-factors can be evaluated by first evaluating  $J_\mu$  as above, and then using further simulation and averaging (if necessary) to compute the right-hand side of Eq. (3.5) for all pairs  $(i, u)$ . Once this is done, one can execute a policy improvement step using the equation

$$\bar{\mu}(i) = \arg \min_{u \in U(i)} Q_\mu(i, u), \quad i = 1, \dots, n. \quad (3.6)$$

We thus obtain a version of the policy iteration algorithm that combines policy evaluation using simulation, and policy improvement using Eq. (3.6) and further simulation, if necessary. In particular, given a policy  $\mu$  and its associated cost vector  $J_\mu$ , the cost of the improved policy  $J_{\bar{\mu}}$  is computed by simulation, with  $\bar{\mu}(i)$  determined using Eq. (3.6) on-line.

#### 2.3.2 Q-Learning

We now introduce an alternative method for cases where there is no explicit model of the system and the cost structure. This method is analogous to value iteration and has the advantage that it can be used directly in the case of multiple policies. Instead of approximating the cost function of a particular policy, it updates directly the *Q*-factors associated with an *optimal* policy, thereby avoiding the multiple policy evaluation

steps of the policy iteration method. These  $Q$ -factors are defined, for all pairs  $(i, u)$  by

$$Q(i, u) := \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + J^*(j)).$$

From this definition and Bellman's equation, we see that the  $Q$ -factors satisfy for all pairs  $(i, u)$ ,

$$Q(i, u) = \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right), \quad (3.7)$$

and it can be shown that the  $Q$ -factors are the unique solution of the above system of equations. The proof is essentially the same as the proof of existence and uniqueness of solution of Bellman's equation; see Prop. 1.2 of Section 2.1. In fact, by introducing a system whose states are the original states  $1, \dots, n, t$  together with all the pairs  $(i, u)$ , the above system of equations can be seen to be a special case of Bellman's equation (see Exercise 2.17). Furthermore, the  $Q$ -factors can be obtained by the iteration

$$Q(i, u) := \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right), \quad \text{for all } (i, u),$$

which is analogous to value iteration. A more general version of this is

$$Q(i, u) := (1-\gamma)Q(i, u) + \gamma \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right), \quad (3.8)$$

where  $\gamma$  is a stepsize parameter with  $\gamma \in (0, 1]$ , that may change from one iteration to the next. The  $Q$ -learning method is an approximate version of this iteration, whereby the expected value is replaced by a single sample, i.e.,

$$Q(i, u) := Q(i, u) + \gamma \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') - Q(i, u) \right).$$

Here  $j$  and  $g(i, u, j)$  are generated from the pair  $(i, u)$  by simulation, that is, according to the transition probabilities  $p_{ij}(u)$ . Thus  $Q$ -learning can be viewed as a combination of value iteration and simulation.

Because  $Q$ -learning works using a single sample per iteration, it is well suited for a simulation context. By contrast, there is no single sample version of the value iteration method, except in special cases [see Exercise 2.9(d)]. The reason is that, while it is possible to use a single-sample approximation of a term of the form  $E\{\min[\cdot]\}$ , such as the one appearing

in the  $Q$ -factor equation (3.8), it is not possible to do so for a term of the form  $\min\{E\{\cdot\}\}$ , such as the one appearing in Bellman's equation.

To guarantee the convergence of the  $Q$ -learning algorithm to the optimal  $Q$ -factors, all state-control pairs  $(i, u)$  must be visited infinitely often, and the stepsize  $\gamma$  should be chosen in some special way. In particular, if the iteration corresponds to the  $m$ th visit of the pair  $(i, u)$ , one may use in the  $Q$ -learning iteration the stepsize  $\gamma_k = c/m$ , where  $c$  is a positive constant. We refer to [Tsi9-1] for a proof of convergence of  $Q$ -learning under very general conditions.

### 2.3.3 Approximations

We now consider approximation/suboptimal control schemes that are suitable for problems with a large number of states. The discounted versions of these schemes, which are discussed in Section 2.3.4, can be adapted for the case of an infinite state space. Generally there are two types of approximations to consider:

- (a) Approximation of the optimal cost function  $J^*$ . This is done by using a function that, given a state  $i$ , produces an approximation  $\hat{J}(i, r)$  of  $J^*(i)$  where  $r$  is a parameter/weight vector that is typically determined by some form of optimization; for example, by using some type of least squares framework. Once  $\hat{J}(i, r)$  is known, it can be used in real-time to generate a suboptimal control at any state  $i$  according to

$$\hat{\mu}(i) = \arg \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + \hat{J}(j, r)).$$

An alternative possibility, which does not require the real-time calculation of the expected value in the above formula, is to obtain approximations  $\hat{Q}(i, u, r)$  of the  $Q$ -factors  $Q(i, u)$ , and then to generate a suboptimal control at any state  $i$  according to

$$\hat{\mu}(i) = \arg \min_{u \in U(i)} \hat{Q}(i, u, r).$$

It is also possible to use approximations  $\hat{J}_\mu(i, r)$  of the cost functions  $J_\mu$  of policies  $\mu$  in an approximate policy iteration scheme. Note that the cost approximation approach can be enhanced if we have additional information on the true functions  $J^*(i)$ ,  $Q(i, u)$ , or  $J_\mu(i)$ . For example, if we know that  $J^*(i) \geq 0$  for all  $i$ , we may first compute the approximation  $\hat{J}(i, r)$  by using some method, and then replace this approximation by  $\max\{0, \hat{J}(i, r)\}$ . This idea applies to all the approximation procedures of this section.

- (b) Approximation of a policy  $\mu$ , or the optimal policy  $\mu^*$ . Again this approximation will be done by some function parameterized by a

parameter/weight vector  $r$ , which given a state  $i$ , produces an approximation  $\hat{\mu}(i, r)$  of  $\mu(i)$  or an approximation  $\hat{\mu}^*(i, r)$  of  $\mu^*(i)$ . The parameter/weight vector  $r$  can be determined by some type of least squares optimization framework.

In this section we discuss several possibilities, emphasizing primarily the case of cost approximation. The choice of the structure of the approximating functions is very significant for the success of the approximation approach. One possibility is to use the *linear* form

$$\hat{J}(i, r) = \sum_{k=1}^m r_k w_k(i), \quad (3.9)$$

where  $r = (r_1, \dots, r_m)$  is the parameter vector and  $w_k(i)$  are some fixed and known scalars. This amounts to approximating the cost function  $J^*$  by a linear combination of  $m$  given basis functions  $(w_k(1), \dots, w_k(n))$ , where  $k = 1, \dots, m$ .

### Example 3.1: (Polynomial Approximations)

An important example of linear cost approximation is based on polynomial basis functions. Suppose that the state consists of  $q$  integer components  $x_1, \dots, x_q$ , each taking values within some limited range of the nonnegative integers. For example, in a queueing system,  $x_k$  may represent the number of customers in the  $k$ th queue, where  $k = 1, \dots, q$ . Suppose that we want to use an approximating function that is quadratic in the components  $x_k$ . Then we can define a total of  $1 + q + q^2$  basis functions that depend on the state  $x = (x_1, \dots, x_q)$  via

$$w_0(x) = 1, \quad w_k(x) = x_k, \quad w_{ks}(x) = x_k x_s, \quad k, s = 1, \dots, q.$$

An approximating function that is a linear combination of these functions is given by

$$\hat{J}(x, r) = r_0 + \sum_{k=1}^q r_k x_k + \sum_{k=1}^q \sum_{s=1}^q r_{ks} x_k x_s,$$

where the parameter vector  $r$  has components  $r_0$ ,  $r_k$ , and  $r_{ks}$ , with  $k, s = 1, \dots, q$ . In fact, any kind of approximating function that is polynomial in the components  $x_1, \dots, x_q$  can be constructed in this way.

### Example 3.2: (Feature Extraction)

Suppose that through intuition or analysis we can identify a number of characteristics of the state that affect the optimal cost function in a substantial way. We assume that these characteristics can be numerically quantified, and that they form a  $q$ -dimensional vector  $f(i) = (f_1(i), \dots, f_q(i))$ , called the *feature vector* of state  $i$ . For example, in computer chess (Section 6.3.2 of

Vol. 1) where the state is the current board position, appropriate features are material balance, piece mobility, king safety, and other positional factors. Features, when well-chosen, can capture the dominant nonlinearities of the optimal cost function  $J^*$ , and can be used to approximate  $J^*$  through the linear combination

$$\hat{J}(i, r) = r_0 + \sum_{k=1}^q r_k f_k(i),$$

where  $r_0, r_1, \dots, r_q$  are appropriately chosen weights.

It is also possible to combine feature extraction with more general polynomial approximations of the type discussed in Example 3.1. For example, a feature extraction mapping  $f$  followed by a quadratic polynomial mapping, yields an approximating function of the form

$$\hat{J}(i, r) = r_0 + \sum_{k=1}^q r_k f_k(i) + \sum_{k=1}^q \sum_{s=1}^q r_{ks} f_k(i) f_s(i),$$

where the parameter vector  $r$  has components  $r_0$ ,  $r_k$ , and  $r_{ks}$ , with  $k, s = 1, \dots, q$ . This function can be viewed as a linear cost approximation that uses the basis functions

$$w_0(i) = 1, \quad w_k(i) = f_k(i), \quad w_{ks}(i) = f_k(i) f_s(i), \quad k, s = 1, \dots, q.$$

Note that more than one state may map into the same feature vector, so that each distinct value of feature vector corresponds to a subset of states. This subset may be viewed as an "aggregate state." The optimal cost function  $J^*$  is approximated by a function that is constant over each aggregate state. We will discuss this viewpoint shortly.

It can be seen from the preceding examples that linear approximating functions of the form (3.9) are well suited for a broad variety of situations. There are also interesting nonlinear approximating functions  $\hat{J}(i, r)$ , including those defined by neural networks, perhaps in combination with feature extraction mappings. In our discussion, we will not address the choice of the structure of  $\hat{J}(i, r)$ , but rather focus on various methods for obtaining a suitable parameter vector  $r$ . We will primarily discuss three approaches:

- (a) *Feature-based aggregation*, where  $r$  is determined as the cost vector of an "aggregate stochastic shortest path problem."
- (b) *Minimizing the Bellman equation error*, where  $r$  is determined so that the approximate cost function  $\hat{J}(i, r)$  nearly satisfies Bellman's equation.
- (c) *Approximate policy iteration*, where the cost functions  $J_\mu$  of the generated policies  $\mu$  are approximated by  $\hat{J}_\mu(i, r)$ , with  $r$  chosen according to a least-squares error criterion.

We note, however, that the methods described in this subsection are not fully understood. We have chosen to present them because of their potential to deal with problems that are too complex to be handled in any other way.

### Feature-Based Aggregation

We mentioned earlier in Example 3.2 that a feature extraction mapping divides the state space into subsets. The states of each subset are mapped into the same feature vector, and are “similar” in that they “share the same features.” With this context in mind, let the set of states  $\{1, \dots, n\}$  of the given stochastic shortest path problem be partitioned in  $m$  disjoint subsets  $S_k$ ,  $k = 1, \dots, m$ . We approximate the optimal cost  $J^*(i)$  by a function that is constant over each set  $S_k$ , that is,

$$\hat{J}(i, r) = \sum_{k=1}^m r_k w_k(i),$$

where  $r = (r_1, \dots, r_m)'$  is a vector of parameters and

$$w_k(i) = \begin{cases} 1 & \text{if } i \in S_k, \\ 0 & \text{if } i \notin S_k. \end{cases}$$

Equivalently, the approximate cost function  $(\hat{J}(1, r), \dots, \hat{J}(n, r))'$  is represented as  $W r$ , where  $W$  is the  $n \times m$  matrix whose entry in the  $i$ th row and  $k$ th column is  $w_k(i)$ . The  $i$ th row of  $W$  may be viewed as the feature vector corresponding to state  $i$  (cf. Example 3.2).

In the aggregation approach, the parameters  $r_k$  are obtained as the optimal costs of an “aggregate stochastic shortest path problem” whose states are the subsets  $S_k$ . Thus  $r_k$  is chosen to be the optimal cost of the aggregate state  $S_k$  in an aggregate problem, which is formulated similar to the aggregation method of Section 1.3.3. In particular, let  $Q$  be an  $m \times n$  matrix such that the  $k$ th row of  $Q$  is a probability distribution  $(q_{k1}, \dots, q_{kn})$  with  $q_{ki} = 0$  if  $i \notin S_k$ . As in Section 1.3.3, the structure of  $Q$  implies that for each stationary policy  $\mu$ , the matrix

$$R_\mu = Q P_\mu W$$

is an  $m \times m$  transition probability matrix. The states of the aggregate stochastic shortest path problem are the sets  $S_1, \dots, S_m$  together with the termination state  $t$ ; the stationary policies select at aggregate state  $S_k$  a control  $u \in U(i)$  for each  $i \in S_k$  and thus can be identified with stationary policies of the original stochastic shortest path problem; finally the transition probability matrix corresponding to  $\mu$  in the aggregate stochastic shortest path problem is  $R_\mu$ . Given a stationary policy  $\mu$ , the state transition mechanism in the aggregate stochastic shortest path problem can be described as follows: at aggregate state  $S_k$ , we move to state  $i$  with probability  $q_{ki}$ , then we move to state  $j$  with probability  $p_{ij}(\mu(i))$ , and finally, if  $j$  is not the termination state  $t$ , we move to the aggregate state  $S_l$  corresponding to  $j$  ( $j \in S_l$ ).

Suppose now that  $r = (r_1, \dots, r_m)'$  is the optimal cost function of the aggregate stochastic shortest path problem. Then  $r$  solves the corresponding Bellman equation, which has the form

$$r_k = \sum_{i=1}^n q_{ki} \min_{u \in U(i)} \sum_{j=1}^m p_{ij}(u) \left( g(i, u, j) + \sum_{s=1}^m r_s w_s(j) \right), \quad k = 1, \dots, m.$$

One way to obtain  $r$  is policy iteration based on Monte-Carlo simulation, as described in Section 2.3.1. An alternative, due to [TsV94], is to use a simulation-based form of value iteration for the aggregate problem. Here, at each iteration we choose a subset  $S_k$ , we randomly select a state  $i \in S_k$  according to the probabilities  $q_{ki}$ , and we update  $r_k$  according to

$$r_k := (1 - \gamma)r_k + \gamma \min_{u \in U(i)} \sum_{j=1}^m p_{ij}(u) \left( g(i, u, j) + \sum_{s=1}^m r_s w_s(j) \right), \quad (3.10)$$

where  $\gamma$  is a positive stepsize that diminishes to zero as the algorithm progresses. The following example illustrates the method. We refer to [TsV94] for experimental results relating to this example as well for convergence analysis of the method.

#### Example 3.3: (Tetris [TsV94])

Tetris is a popular video game played on a two-dimensional grid. Each square in the grid can be full or empty, making up a “wall of bricks” with “holes” and a “jagged top”. The squares fill up as blocks of different shapes fall at a constant rate from the top of the grid and are added to the top of the wall. As a given block falls, the player can move horizontally and rotate the block in all possible ways, subject to the constraints imposed by the sides of the grid and the top of the wall. There is a finite set of standard shapes for the falling blocks. The game starts with an empty grid and ends when a square in the top row becomes full and the top of the wall reaches the top of the grid. However, when a row of full squares is created, this row is removed, the bricks lying above this row move one row downward, and the player scores a point. The player’s objective is to maximize the score attained (total number of rows removed) up to termination of the game.

Assuming that, for every policy, the game terminates with probability one (something that is not really known at present), we can model the problem of finding an optimal tetris playing strategy as a stochastic shortest path problem. The control, denoted by  $u$ , is the horizontal positioning and rotation applied to the falling block. The state consists of two components:

- (1) The board position, that is, a binary description of the full/empty status of each square, denoted by  $x$ .
- (2) The shape of the current falling block, denoted by  $y$ .

The component  $y$  is generated according to a probability distribution  $p(y)$ , independently of the control. Exercise 2.9 shows that under these circumstances, it is possible to derive a reduced form of Bellman’s equation

involving a cost function  $\hat{J}$  that depends only on the component  $x$  of the state (see also Exercise 1.22 of Vol. I). This equation has the intuitive form

$$\hat{J}(x) = \sum_y p(y) \max_u [g(x, y, u) + \hat{J}(f(x, y, u))], \quad \text{for all } x,$$

where  $g(x, y, u)$  and  $f(x, y, u)$  are the number of points scored (rows removed), and the board position when the state is  $(x, y)$  and control  $u$  is applied, respectively.

Unfortunately, the number of states is extremely large. It is equal to  $m2^{hw}$ , where  $m$  is the number of different shapes of falling blocks, and  $h$  and  $w$  are the height and width of the grid, respectively. In particular, for the reasonable numbers  $m = 7$ ,  $h = 20$ , and  $w = 7$  we have over  $10^{12}$  states. Thus it is essential to use approximations.

An approximating function that involves feature extraction is particularly attractive here, since the quality of a given position can be described quite well by a few features that are easily recognizable by experienced players. These features include the current height of the wall, and the presence of "holes" and "glitches" (severe irregularities) in the first few rows. Suppose that, based on human experience and trial and error, we obtain a method to map each board position  $x$  into a vector of features. Suppose that there is a finite number of possible feature vectors, say  $m$ , and define

$$w_k(x) = \begin{cases} 1 & \text{if board position } x \text{ maps into the } k\text{th feature vector,} \\ 0 & \text{otherwise.} \end{cases}$$

The approximating function  $\hat{J}(x, r)$  is given by

$$\hat{J}(x, r) = \sum_{k=1}^m r_k w_k(x),$$

where  $r = (r_1, \dots, r_m)$  is the parameter vector. The simulation-based value iteration (3.10) takes the form

$$r_k := (1 - \gamma)r_k + \gamma \max_u [g(x, y, u) + \hat{J}(f(x, y, u), r)],$$

where the positive stepsize  $\gamma$  diminishes with the number of visits to position  $x$ .

One way to implement the method is as follows: The game is simulated many times from start to finish, starting from a variety of "representative" board positions. At each iteration, we have the current board position  $x$  and we determine the feature vector  $k$  to which  $x$  maps. Then we randomly generate a falling block  $y$  according to a known and fixed probabilistic mechanism, and we update  $r_k$  using the above iteration. Let  $u^*$  be the choice of  $u$  that attains the maximum in the iteration,

$$u^* = \arg \max_u [g(x, y, u) + \hat{J}(f(x, y, u), r)].$$

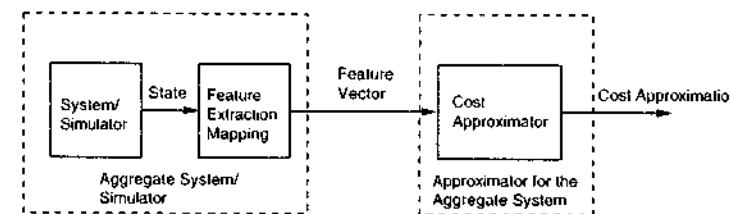
Then the board position subsequent to  $x$  in the simulation is  $f(x, y, u^*)$ , and this position is used as the current state for the next iteration.

In the aggregate stochastic shortest path problem formulated above, policies consist of a different control choice for each state. A somewhat different aggregate stochastic shortest path problem is obtained by requiring that, for each  $k$ , the same control is used at all states of  $S_k$ . This control must be chosen from a suitable set  $\bar{U}(k)$  of admissible controls for the states in  $S_k$ . The optimal cost function  $r = (r_1, \dots, r_m)'$  corresponding to this aggregate stochastic shortest path problem solves the following Bellman equation

$$r_k = \min_{u \in \bar{U}(k)} \sum_{i=1}^n q_{ki} \sum_{j=1}^m p_{ij}(u) \left( g(i, u, j) + \sum_{s=1}^m r_s w_s(j) \right), \quad k = 1, \dots, m.$$

This equation can be solved by  $Q$ -learning, particularly when  $m$  is relatively small and the number of controls in the sets  $\bar{U}(k)$  is also small.

Note also that the aggregate problem need not be solved exactly, but can itself be solved approximately by any of the simulation-based methods to be discussed subsequently in this section. In this context, aggregation is used as a feature extraction mapping that maps each state  $i$  to the corresponding feature vector  $w(i) = (w_1(i), \dots, w_m(i))$ . This feature vector becomes the input to some other approximating function (see Fig. 2.3.1).



**Figure 2.3.1** View of a cost function approximation scheme that consists of a feature extraction mapping followed by an approximator. The scheme conceptually separates into an aggregate system and a cost approximator for the aggregate system.

We finally mention an extension of the aggregation approach whereby we represent the approximate cost function  $(\hat{J}(1, r), \dots, \hat{J}(n, r))'$  as  $Wr$ , where each row of the  $n \times m$  matrix  $W$  is a probability distribution. Thus

$$\hat{J}(i, r) = \sum_{k=1}^m r_k w_k(i),$$

where  $w_k(i)$  is the  $(i, k)$ th entry of the matrix  $W$ , and we have

$$\sum_{k=1}^n w_k(i) = 1, \quad w_k(i) \geq 0, \quad i = 1, \dots, n, \quad k = 1, \dots, m.$$

The transition probability matrix of the aggregate stochastic shortest path problem corresponding to  $\mu$  is still  $R_\mu = QP_\mu W$ , and we may use as parameter vector  $r$  the optimal cost vector of this aggregate problem.

### Approximation Based on Bellman's Equation

Another possibility for approximation of the optimal cost by a function  $\hat{J}(i, r)$ , where  $r$  is a vector of unknown parameters, is based on minimizing the error in Bellman's equation; for example by solving the problem

$$\min_r \sum_{i \in S} \left| \hat{J}(i, r) - \min_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \hat{J}(j, r)) \right|^2, \quad (3.11)$$

where  $S$  is a suitably chosen subset of "representative" states. This minimization may be attempted by using some type of gradient or Gauss-Newton method.

A gradient-like method that can be used to solve this problem is obtained by making a correction to  $r$  that is proportional to the gradient of the squared error term in Eq. (3.11). This method is given by

$$\begin{aligned} r &:= r - \gamma D(i, r) \nabla D(i, r) \\ &:= r - \gamma D(i, r) \left( \sum_j p_{ij}(\bar{u}) \nabla \hat{J}(j, r) - \nabla \hat{J}(i, r) \right), \end{aligned} \quad (3.12)$$

where  $\nabla$  denotes the gradient with respect to  $r$ ,  $D(i, r)$  is the error in Bellman's equation, given by

$$D(i, r) := \min_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \hat{J}(j, r)) - \hat{J}(i, r),$$

$\bar{u}$  is given by

$$\bar{u} = \arg \min_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \hat{J}(j, r)),$$

and  $\gamma$  is a stepsize, which may change from one iteration to the next. The method should perform many such iterations at each of the representative states. Typically one should cycle through the set of representative states  $S$  in some order, which may change (perhaps randomly) from one cycle to the next.

Note that in iteration (3.12) we approximate the gradient of the term

$$\min_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \hat{J}(j, r)) \quad (3.13)$$

by

$$\sum_j p_{ij}(\bar{u}) \nabla \hat{J}(j, r),$$

which can be shown to be correct only when the above minimum is attained at a unique  $\bar{u} \in U(i)$  [otherwise the function (3.13) is nondifferentiable with respect to  $r$ ]. Thus the convergence of iteration (3.12) should be analyzed using the theory of nondifferentiable optimization. One possibility to avoid this complication is to replace the nondifferentiable term (3.13) by a smooth approximation, which can be arbitrarily accurate (see [Ber82b], Ch. 3).

An interesting special case arises when we want to approximate the cost function of a given policy  $\mu$  by a function  $\hat{J}_\mu(i, r)$ , where  $r$  is a parameter vector. The iteration (3.12) then takes the form

$$\begin{aligned} r &:= r - \gamma E_J \{ d_\mu(i, j, r) \mid i, \mu \} E_J \{ \nabla d_\mu(i, j, r) \mid i, \mu \} \\ &= r - \gamma E_J \{ d_\mu(i, j, r) \mid i, \mu \} (E_J \{ \nabla \hat{J}_\mu(j, r) \mid i, \mu \} - \nabla \hat{J}_\mu(i, r)), \end{aligned} \quad (3.14)$$

where

$$d_\mu(i, j, r) = g(i, \mu(i), j) + \hat{J}_\mu(j, r) - \hat{J}_\mu(i, r),$$

and  $E_J \{ \cdot \mid i, \mu \}$  denotes expected value over  $j$  using the transition probabilities  $p_{ij}(\mu(i))$ . There is a simpler version of iteration (3.14) that does not require averaging over the successor states  $j$ . In this version, the two expected values in iteration (3.14) are replaced by two independent single sample values. In particular,  $r$  is updated by

$$r := r - \gamma d_\mu(i, j, r) (\nabla \hat{J}_\mu(\bar{j}, r) - \nabla \hat{J}_\mu(i, r)), \quad (3.15)$$

where  $j$  and  $\bar{j}$  correspond to two independent transitions starting from  $i$ . It is necessary to use two independently generated states  $j$  and  $\bar{j}$  in order that the expected value (over  $j$  and  $\bar{j}$ ) of the product

$$d_\mu(i, j, r) (\nabla \hat{J}_\mu(\bar{j}, r) - \nabla \hat{J}_\mu(i, r)),$$

given  $i$ , is equal to the term

$$E_J \{ d_\mu(i, j, r) \mid i, \mu \} (E_J \{ \nabla \hat{J}_\mu(j, r) \mid i, \mu \} - \nabla \hat{J}_\mu(i, r))$$

appearing in the right-hand side of Eq. (3.14).

There are also versions of the above iterations that update  $Q$ -factor approximations rather than cost approximations. In particular, let us introduce an approximation  $\tilde{Q}(i, u, r)$  to the  $Q$ -factor  $Q(i, u)$ , where  $r$  is an

unknown parameter vector. Bellman's equation for the  $Q$ -factors is given by [cf. Eq. (3.7)]

$$Q(i, u) = \sum_j p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right),$$

so in analogy with problem (3.11), we determine the parameter vector  $r$  by solving the least squares problem

$$\min_r \sum_{(i, u) \in \tilde{V}} \left| \hat{Q}(i, u, r) - \sum_j p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} \hat{Q}(j, u', r) \right) \right|^2, \quad (3.16)$$

where  $\tilde{V}$  is a suitably chosen subset of "representative" state-control pairs. The analog of the gradient-like methods (3.12) and (3.14) is given by

$$\begin{aligned} r &:= r - \gamma E\{d_u(i, j, r) \mid i, u\} E\{\nabla d_u(i, j, r) \mid i, u\} \\ &= r - \gamma E\{d_u(i, j, r) \mid i, u\} \left( \sum_j p_{ij}(u) \nabla \hat{Q}(j, \bar{u}, r) - \nabla \hat{Q}(i, u, r) \right), \end{aligned}$$

where  $d_u(i, j, r)$  is given by

$$d_u(i, j, r) = g(i, u, j) + \min_{u' \in U(j)} \hat{Q}(j, u', r) - \hat{Q}(i, u, r),$$

$\bar{u}$  is obtained by

$$\bar{u} = \arg \min_{u' \in U(j)} \hat{Q}(j, u', r),$$

and  $\gamma$  is a stepsize parameter. In analogy with Eq. (3.15), the two-sample version of this iteration is given by

$$r := r - \gamma d_u(i, j, r) (\nabla \hat{Q}(\bar{j}, \bar{u}, r) - \nabla \hat{Q}(i, u, r)),$$

where  $j$  and  $\bar{j}$  are two states independently generated from  $i$  according to the transition probabilities corresponding to  $u$ , and

$$\bar{u} = \arg \min_{u' \in U(j)} \hat{Q}(\bar{j}, u', r).$$

Note that there is no two-sample version of iteration (3.12), which is based on optimal cost approximation. This is the advantage of using  $Q$ -factor approximations rather than optimal cost approximations. The point is that it is possible to use single-sample or two-sample approximations in gradient-like methods for terms of the form  $E\{\min[\cdot]\}$ , such as the one appearing in Eq. (3.16), but not for terms of the form  $\min[E\{\cdot\}]$ , such as the one appearing in Eq. (3.11). The following example illustrates the use of the two-sample approximation idea.

### Example 3.4: (Tetris Continued)

Consider the game of tetris described in Example 3.3, and suppose that an approximation of a given form  $\hat{J}(x, r)$  is desired, where the parameter vector  $r$  is obtained by solving the problem

$$\min_r \sum_{i \in S} \left| \hat{J}(x, r) - \sum_y p(y) \max_u [g(x, y, u) + \hat{J}(f(x, y, u), r)] \right|^2,$$

where  $\hat{S}$  is a suitably chosen set of "representative" states. Because this problem involves a term of the form  $E\{\max[\cdot]\}$ , a two-sample gradient-like method is possible. It has the form

$$r := r - \gamma d(x, y, r) (\nabla \hat{J}(f(x, \bar{y}, \bar{u}), r) - \nabla \hat{J}(x, r)),$$

where  $y$  and  $\bar{y}$  are two falling blocks that are randomly and independently generated.

$$d(x, y, r) = \max_u [g(x, y, u) + \hat{J}(f(x, y, u), r)] - \hat{J}(x, r),$$

and

$$\bar{u} = \arg \max_{u'} [g(x, \bar{y}, u') + \hat{J}(f(x, \bar{y}, u'), r)].$$

Similar to Example 3.3, consider a feature-based approximating function  $\hat{J}(x, r)$  given by

$$\hat{J}(x, r) = \sum_{k=1}^m r_k w_k(x),$$

where  $r = (r_1, \dots, r_m)$  is the parameter vector and

$$w_k(x) = \begin{cases} 1 & \text{if board position } x \text{ maps into the } k\text{th feature vector,} \\ 0 & \text{otherwise.} \end{cases}$$

For this approximating function, the preceding two-sample gradient iteration takes the relatively simple form

$$r_k := r_k - \gamma d(x, \bar{y}, r) (w_k(f(x, \bar{y}, \bar{u})) - w_k(x)), \quad k = 1, \dots, m.$$

Note that this iteration updates at most two parameters [the ones corresponding to the feature vectors to which the board positions  $x$  and  $f(x, \bar{y}, \bar{u})$  map, assuming that these feature vectors are different]. To implement the method, a set  $\hat{S}$  containing a large number of states  $x$  is selected and at each  $x \in \hat{S}$ , two falling blocks  $y$  and  $\bar{y}$  are independently generated. The controls  $u$  and  $\bar{u}$  that are optimal for  $(x, y)$  and  $(x, \bar{y})$ , based on the current parameter vector  $r$ , are calculated, and the parameters of the feature vectors associated with  $x$  and  $f(x, \bar{y}, \bar{u})$  are adjusted according to the preceding formula.

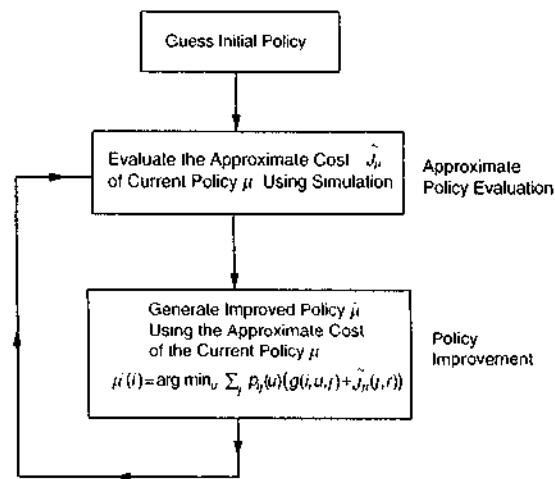


Figure 2.3.2 Block diagram of approximate policy iteration.

### Approximate Policy Iteration Using Monte-Carlo Simulation

We now discuss an approximate form of the policy iteration method, where we use approximations  $\tilde{J}_\mu(i, r)$  to the cost  $J_\mu$  of stationary policies  $\mu$ , and/or approximations  $\tilde{Q}_\mu(i, u, r)$  to the corresponding  $Q$ -factors. The theoretical basis of the method was discussed in Section 2.2.2 (cf. Prop. 2.1).

Similar to our earlier discussion on simulation, suppose that for a fixed stationary policy  $\mu$ , we have a subset of “representative” states  $\hat{S}$  (perhaps chosen in the course of the simulation), and that for each  $i \in \hat{S}$ , we have  $M(i)$  samples of the cost  $J_\mu(i)$ . The  $m$ th such sample is denoted by  $c(i, m)$ . Then, we can introduce approximate costs  $\tilde{J}_\mu(i, r)$ , where  $r$  is a parameter/weight vector obtained by solving the following least-squares optimization problem

$$\min_r \sum_{i \in \hat{S}} \sum_{m=1}^{M(i)} |\tilde{J}_\mu(i, r) - c(i, m)|^2.$$

Once the optimal value of  $r$  has been determined, we can approximate the costs  $J_\mu(i)$  of the policy  $\mu$  by  $\tilde{J}_\mu(i, r)$ . Then, we can evaluate approximate  $Q$ -factors using the formula

$$\tilde{Q}_\mu(i, u, r) = \sum_j p_{ij}(u)(g(i, u, j) + \tilde{J}_\mu(j, r)). \quad (3.17)$$

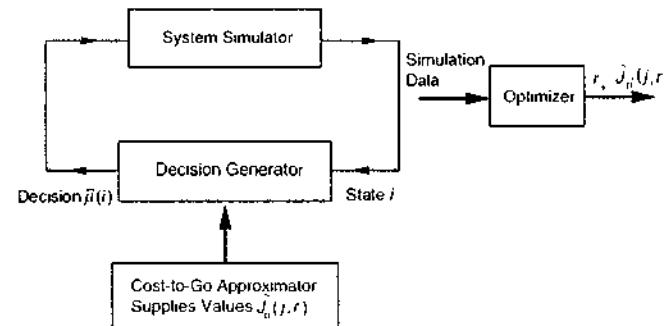


Figure 2.3.3 Structure of approximate policy iteration algorithm.

and we can obtain an improved policy  $\bar{\mu}$  using the formula

$$\begin{aligned} \bar{\mu}(i) &= \arg \min_{u \in U(i)} \tilde{Q}_\mu(i, u, r) \\ &= \arg \min_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \tilde{J}_\mu(j, r)), \quad \text{for all } i. \end{aligned} \quad (3.18)$$

We thus obtain an algorithm that alternates between approximate policy evaluation steps and policy improvement steps, as illustrated in Fig. 2.3.2. The algorithm requires a single approximation per policy iteration, namely the approximation  $\tilde{J}_\mu(i, r)$  associated with the current policy  $\mu$ . The parameter vector  $r$  determines the  $Q$ -factors via Eq. (3.17) and the next policy  $\bar{\mu}$  via Eq. (3.18).

For another view of the approximate policy iteration algorithm, note that it consists of four modules (see Fig. 2.3.3):

- The *simulator*, which given a state-decision pair  $(i, u)$ , generates the next state  $j$  according to the correct transition probabilities.
- The *decision generator*, which generates the decision  $\bar{\mu}(i)$  of the improved policy at the current state  $i$  [cf. Eq. (3.18)] for use in the simulator.
- The *cost-to-go approximator*, which is the function  $\tilde{J}_\mu(j, r)$  that is consulted by the decision generator for approximate cost-to-go values to use in the minimization of Eq. (3.18).
- The *optimizer*, which accepts as input the sample trajectories produced by the simulator and solves the problem

$$\min_r \sum_{i \in \hat{S}} \sum_{m=1}^{M(i)} |\tilde{J}_\mu(i, r) - c(i, m)|^2 \quad (3.19)$$

to obtain the approximation  $\tilde{J}_{\bar{\mu}}(i, \bar{r})$  of the cost of  $\bar{\mu}$ .

Note that in very large problems, the policy  $\bar{\mu}$  cannot be evaluated and stored in explicit form, and thus the optimization in Eq. (3.18) must be evaluated “on the fly” during the simulation. When this is the case, the parameter vector  $\bar{r}$  associated with  $\mu$  remains unchanged as we evaluate the cost of the improved policy  $\bar{\mu}$  by generating the simulation data and by solving the least squares problem (3.19).

One way to solve this latter problem is to use gradient-like methods. Given a sample state trajectory  $(i_1, i_2, \dots, i_N, t)$  generated using the policy  $\bar{\mu}$ , which is defined by Eq. (3.18), the parameter vector  $\bar{r}$  associated with  $\bar{\mu}$  is updated by

$$\bar{r} := \bar{r} - \gamma \sum_{k=1}^N \nabla \tilde{J}_{\bar{\mu}}(i_k, \bar{r}) \left( \tilde{J}_{\bar{\mu}}(i_k, \bar{r}) - \sum_{m=k}^N g(i_m, \bar{\mu}(i_m), i_{m+1}) \right), \quad (3.20)$$

where  $\gamma$  is a stepsize. The summation in the right-hand side above is a sample gradient corresponding to a term in the least squares summation of problem (3.19).

We finally mention two variants of the approximate policy iteration algorithm, both of which require additional approximations per policy iteration. In the first variant, instead of calculating the approximate  $Q$ -factors via Eq. (3.17), we form an approximation  $\hat{Q}_{\mu}(i, u, y)$ , where the parameter vector  $y$  is determined by solving the least squares problem

$$\min_y \sum_{(i,u) \in \hat{V}} |\hat{Q}_{\mu}(i, u, y) - \hat{Q}_{\mu}(i, u, r)|^2, \quad (3.21)$$

where  $\hat{V}$  is a “representative” set of state-control pairs  $(i, u)$ , and  $\hat{Q}_{\mu}(i, u, r)$  is evaluated using Eq. (3.17) and either exact calculation or simulation. This variant is useful if it speeds up the calculations of the policy improvement step [cf. Eq. (3.18)].

In the second variant of the algorithm, we first perform the approximate policy evaluation step to obtain  $\tilde{J}_{\mu}(i, r)$ . Then we compute the improved policy  $\bar{\mu}(i)$  by the formula (3.18) only for states  $i$  in a “representative” subset  $S$ . We then obtain an “improved” policy  $\hat{\mu}(i, v)$ , which is defined over all states, by introducing a parameter vector  $v$  and by solving the least squares problem

$$\min_v \sum_{i \in S} \|\bar{\mu}(i) - \hat{\mu}(i, v)\|^2. \quad (3.22)$$

Here, we assume that the controls are elements of some Euclidean space and  $\|\cdot\|$  denotes the norm on that space. This approach accelerates the policy improvement step [cf. Eq. (3.18)] at the expense of solving an additional least squares problem per policy iteration.

### Approximate Policy Iteration Using TD(1)

Just as there is a temporal differences implementation of Monte-Carlo simulation, there is also a temporal differences implementation of the gradient iteration (3.20). The temporal differences  $d_k$  are given by

$$d_k = g(i_k, \bar{\mu}(i_k), i_{k+1}) + \tilde{J}_{\bar{\mu}}(i_{k+1}, \bar{r}) - \tilde{J}_{\bar{\mu}}(i_k, \bar{r}), \quad k = 1, \dots, N, \quad (3.23)$$

and the iteration (3.20) can be alternatively written as follows [just add the equations below using the temporal difference expression (3.23) to obtain the iteration (3.20)]:

Following the state transition  $(i_1, i_2)$ , set

$$\bar{r} := \bar{r} + \gamma d_1 \nabla \tilde{J}_{\bar{\mu}}(i_1, \bar{r}). \quad (3.24)$$

Following the state transition  $(i_2, i_3)$ , set

$$\bar{r} := \bar{r} + \gamma d_2 (\nabla \tilde{J}_{\bar{\mu}}(i_1, \bar{r}) + \nabla \tilde{J}_{\bar{\mu}}(i_2, \bar{r})). \quad (3.25)$$

Following the state transition  $(i_N, t)$ , set

$$\bar{r} := \bar{r} + \gamma d_N (\nabla \tilde{J}_{\bar{\mu}}(i_1, \bar{r}) + \nabla \tilde{J}_{\bar{\mu}}(i_2, \bar{r}) + \dots + \nabla \tilde{J}_{\bar{\mu}}(i_N, \bar{r})). \quad (3.26)$$

The vector  $\bar{r}$  may be updated at each transition, although the gradients  $\nabla \tilde{J}_{\bar{\mu}}(i_k, \bar{r})$  are evaluated for the value of  $\bar{r}$  that prevails at the time  $i_k$  is generated. Also, for convergence, the stepsize  $\gamma$  should diminish over time. A popular choice is to use during the  $m$ th trajectory  $\gamma = c/m$ , where  $c$  is a constant.

A variant of this method that has been proposed under the name TD( $\lambda$ ) uses a parameter  $\lambda \in [0, 1]$  in the formulas (3.23)-(3.26). It has the following form:

For  $k = 1, \dots, N$ , following the state transition  $(i_k, i_{k+1})$ , set

$$\bar{r} := \bar{r} + \gamma d_k \sum_{m=1}^k \lambda^{k-m} \nabla \tilde{J}_{\bar{\mu}}(i_m, \bar{r}).$$

While this method has received wide attention, its validity has been questioned. Examples have been constructed [Ber95b] where the approximating function  $\tilde{J}_{\bar{\mu}}(i, \bar{r})$  obtained in the limit by TD( $\lambda$ ) is an increasingly poor approximation to  $J_{\mu}(i)$  as  $\lambda$  decreases towards 0, and the approximation obtained by TD(0) is very poor. It is possible, however, to use the two-sample gradient iteration (3.15) for a simulation-based, approximate evaluation of the cost functions of various policies in an approximate policy iteration scheme. This iteration resembles the TD(0) formula but aims at minimizing the error in Bellman’s equation.

### Optimistic Policy Iteration

In the approximate policy iteration approach discussed so far, the least squares problem that evaluates the cost of the improved policy  $\bar{\mu}$  must be solved completely for the vector  $\bar{r}$ . An alternative is to solve this problem approximately and replace the policy  $\mu$  with the policy  $\bar{\mu}$  after a single or a few simulation runs. An extreme possibility is to replace  $\mu$  with  $\bar{\mu}$  at the end of each state transition, as in the next algorithm:

Following the state transition  $(i_k, i_{k+1})$ , set

$$\bar{r} := \bar{r} + \gamma d_k \sum_{m=1}^k \nabla \hat{J}_{\bar{\mu}}(i_m, \bar{r}),$$

and generate the next transition  $(i_{k+1}, i_{k+2})$  by simulation using the control

$$\bar{\mu}(i_{k+1}) = \arg \min_{u \in U(i)} \sum_j p_{i_{k+1}j}(u) (g(i_{k+1}, u, j) + \hat{J}_{\bar{\mu}}(j, \bar{r})).$$

The theoretical convergence properties of this method have not been investigated so far, although its TD( $\lambda$ ) version has been used with success in solving some challenging problems [Tes92].

### Variations Involving Multistage Lookahead

To reduce the effect of the approximation error

$$J_{\mu}(i) - \hat{J}_{\mu}(i, r)$$

between the true and approximate costs of a policy  $\mu$ , one can consider a lookahead of several stages in computing the improved policy  $\bar{\mu}$ . The method adopted earlier for generating the decisions  $\bar{\mu}(i)$  of the improved policy,

$$\bar{\mu}(i) = \arg \min_{u \in U(i)} \sum_j p_{ij}(u) (g(i, u, j) + \hat{J}_{\mu}(j, r)), \quad \text{for all } i,$$

corresponds to a single stage lookahead. At a given state  $i$ , it finds the optimal decision for a one-stage problem with stage cost  $g(i, u, j)$  and terminal cost (after the first stage)  $\hat{J}_{\mu}(j, r)$ .

An  $m$ -stage lookahead version finds the optimal policy for an  $m$ -stage problem, whereby we start at the current state  $i$ , make the  $m$  subsequent decisions with perfect state information, incur the corresponding costs of the  $m$  stages, and pay a terminal cost  $\hat{J}_{\mu}(j, r)$ , where  $j$  is the state after  $m$  stages. This is a finite horizon stochastic optimal control problem that may be tractable, depending on the horizon  $m$  and the number of possible

successor states from each state. If  $u_1(i)$  is the first decision of the  $m$ -stage lookahead optimal policy starting at state  $i$ , the improved policy is defined by

$$\bar{\mu}(i) = u_1(i).$$

Note that if  $\hat{J}_{\mu}(j, r)$  is equal to the exact cost  $J_{\mu}(j)$  for all states  $j$ , that is, there is no approximation, the multistage version of policy iteration can be shown to terminate with an optimal policy under the same conditions as ordinary policy iteration (see Exercise 2.16).

Multistage lookahead can also be used in the real-time calculation of a suboptimal control policy, once an approximation  $\hat{J}(i, r)$  of the optimal cost has been obtained by any one of the methods of this subsection. An example is the computer chess programs discussed in Section 6.3 of Vol. I. In that case, the approximation of the cost-to-go function (the position evaluator discussed in Section 6.3 of Vol. I) is relatively primitive. It is derived from the features of the position (material balance, piece mobility, king safety, etc.), appropriately weighted with factors that are either heuristically determined by trial and error, or (in the case of a champion program, IBM's Deep Thought) by training on examples from grandmaster play. It is well-known that the quality of play of computer chess programs crucially depends on the size of the lookahead. This indicates that in many types of problems, the multistage lookahead versions of the methods of this subsection should be much more effective than their single stage lookahead counterparts. This improvement in performance must of course be weighed against the considerable increase in computation required to optimally solve the associated multistage problems.

### Approximation in Policy Space

We finally mention a conceptually different approximation possibility that aims at direct optimization over policies of a given type. Here we hypothesize a stationary policy of a certain structural form, say  $\tilde{\mu}(i, r)$ , where  $r$  is a vector of unknown parameters/weights that is subject to optimization. We also assume that for a fixed  $r$ , the cost of starting at  $i$  and using the stationary policy  $\tilde{\mu}(\cdot, r)$ , call it  $\hat{J}(i, r)$ , can be evaluated by simulation. We may then minimize over  $r$

$$E_i \{ \hat{J}(i, r) \}, \quad (3.27)$$

where the expectation is taken with respect to some probability distribution over the set of initial states. This minimization will typically be carried out by some method that does not require the use of gradients if the gradient of  $\hat{J}(i, r)$  with respect to  $r$  cannot be easily calculated. If the simulation can produce the value of the gradient  $\nabla \hat{J}(i, r)$  together with  $\hat{J}(i, r)$ , then a gradient-based method can be used. Generally, the minimization of the cost

function (3.27) tends to be quite difficult if the dimension of the parameter vector  $r$  is large (say over 10). As a result, the method is most likely effective only when adequate optimal policy approximation is possible with very few parameters.

### 2.3.4 Extension to Discounted Problems

We now discuss adaptations of the simulation-based methods for the case of a discounted problem. Consider first the evaluation of policies by simulation. One difficulty here is that trajectories do not terminate, so we cannot obtain sample costs corresponding to different states. One way to get around this difficulty is to approximate a discounted cost by a finite horizon cost of sufficiently large horizon. Another possibility is to convert the  $\alpha$ -discounted problem to an equivalent stochastic shortest path problem by introducing an artificial termination state  $t$  and a transition probability  $1 - \alpha$  from each state  $i \neq t$  to the termination state  $t$ . The remaining transition probabilities are scaled by multiplication with  $\alpha$  (see Vol. 1, Section 7.3). Bellman's equation for this stochastic shortest path problem is identical with Bellman's equation for the original  $\alpha$ -discounted problem, so the optimal cost functions and the optimal policies of the two problems are identical.

The preceding approaches may lead to long simulation runs, involving many transitions. An alternative possibility that is useful in some cases is based on identifying a special state, called the *reference state*, that is assumed to be reachable from all other states under the given policy. Suppose that such a state can be identified and for concreteness assume that it is state 1. Thus, we assume that the Markov chain corresponding to the given policy has a single recurrent class and state 1 belongs to that class (see Appendix D of Vol. 1). If there are multiple recurrent classes, the procedure described in what follows can be modified so that there is a reference state for each class.

To simplify notation, we do not show the dependence of various quantities on the given policy. In particular, the transition probability from  $i$  to  $j$  and the corresponding stage cost are denoted by  $p_{ij}$  and  $g(i, j)$ , in place of  $p_{ij}(\mu(i))$  and  $g(i, \mu(i), j)$ , respectively. For each initial state  $i$ , let  $C(i)$  denote the average discounted cost incurred up to reaching the reference state 1. Let also  $m_i$  denote the *first passage time* from state  $i$  to state 1, that is, the number of transitions required to reach state 1 starting from state  $i$ . Note that  $m_i$  is a random variable. We denote

$$D(i) = E\{\alpha^{m_i}\}.$$

By dividing the cost  $J_\mu(i)$  into the portion up to reaching state 1 and the

remaining portion starting from state 1, we have

$$\begin{aligned} J_\mu(i) &= E \left\{ \sum_{k=0}^{m_i-1} \alpha^k g(x_k, x_{k+1}) \mid x_0 = i \right\} \\ &\quad + E \left\{ \sum_{k=m_i}^{\infty} \alpha^k g(x_k, x_{k+1}) \mid x_{m_i} = 1 \right\} \\ &= C(i) + D(i)J_\mu(1). \end{aligned} \quad (3.28)$$

Applying this equation for  $i = 1$ , we have  $J_\mu(1) = C(1) + D(1)J_\mu(1)$ , so that

$$J_\mu(1) = \frac{C(1)}{1 - D(1)}. \quad (3.29)$$

Combining Eqs. (3.28) and (3.29), we obtain

$$J_\mu(i) = C(i) + \frac{D(i)C(1)}{1 - D(1)}, \quad i = 1, \dots, n.$$

Therefore, to calculate the cost vector  $J_\mu$ , it is sufficient to calculate the costs  $C(i)$ , and in addition to calculate the expected discount terms  $D(i)$ . Both of these can be computed, similar to the stochastic shortest path problem, by generating many sample system trajectories, and averaging the corresponding sample costs and discount terms up to reaching the reference state 1.

Note here that because  $C(1)$  and  $D(1)$  crucially affect the calculated values  $J_\mu(i)$ , it may be worth doing some extra simulations starting from the reference state 1 to ensure that  $C(1)$  and  $D(1)$  are accurately calculated.

Once a simulation method is available to evaluate (perhaps approximately) the cost of various policies, it can be embedded within a (perhaps approximate) policy iteration algorithm along the lines discussed for the stochastic shortest path problem.

We note also that there is a straightforward extension of the  $Q$ -learning algorithm to discounted problems. The optimal  $Q$ -factors are the unique solution of the equation

$$\begin{aligned} Q(i, u) &= \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + \alpha J^\star(j)) \\ &= \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \alpha \min_{u' \in U(j)} Q(j, u') \right). \end{aligned}$$

This is again proved by introducing a system whose states are pairs  $(i, u)$ , so that the above system of equations becomes a special case of Bellman's

equation. With similar observations, it follows that the vector of  $Q$ -factors can be obtained by the value iteration

$$Q(i, u) := \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \alpha \min_{u' \in U(j)} Q(j, u') \right).$$

The  $Q$ -learning method is an approximate version of this iteration, whereby the expected value is replaced by a single sample, i.e.,

$$Q(i, u) := Q(i, u) + \gamma \left( g(i, u, j) + \alpha \min_{u' \in U(j)} Q(j, u') - Q(i, u) \right).$$

Here  $j$  and  $g(i, u, j)$  are generated from the pair  $(i, u)$  by simulation, that is, according to the transition probabilities  $p_{ij}(u)$ .

We finally note that approximation based on minimization of the error in Bellman's equation can also be used in the case of a discounted cost. One simply needs to introduce the discount factor at the appropriate places in the various iterations given above. For example, the variant of iteration (3.15) for evaluating the discounted cost of a policy  $\mu$  is

$$r := r - \gamma d_\mu(i, j, r) (\alpha \nabla \tilde{J}_\mu(\bar{j}, r) - \nabla \tilde{J}_\mu(i, r)),$$

where  $\alpha$  is the discount factor and

$$d_\mu(i, j, r) = g(i, \mu(i), j) + \alpha \tilde{J}_\mu(j, r) - \tilde{J}_\mu(i, r).$$

### 2.3.5 The Role of Parallel Computation

It is well-known that Monte-Carlo simulation is very well-suited for parallelization; one can simply carry out multiple simulation runs in parallel and occasionally merge the results. Also several DP-related methods are well-suited for parallelization; for example, each value iteration can be parallelized by executing the cost updates of different states in different parallel processors (see e.g., [AMT93]). In fact the parallel updates can be asynchronous. By this we mean that different processors may execute cost updates as fast as they can, without waiting to acquire the most recent updates from other processors; these latter updates may be late in coming because some of the other processors may be slow or because some of the communication channels connecting the processors may be slow. Asynchronous parallel value iteration can be shown to have the same convergence properties as its synchronous counterpart, and is often substantially faster. We refer to [Ber82a] and [BeT89a] for an extensive discussion.

There are similar parallelization possibilities in approximate DP. Indeed, approximate policy iteration may be viewed as a combination of two operations:

- (a) *Simulation*, which produces many pairs  $(i, c(i))$  of states  $i$  and sample costs  $c(i)$  associated with the improved policy  $\bar{\mu}$ .
- (b) *Training*, which obtains the state-sample cost pairs produced by the simulator and uses them in the least-squares optimization of the parameter vector  $\bar{r}$  of the approximate cost function  $\tilde{J}_{\bar{\mu}}(\cdot, \bar{r})$ .

The simulation operation can be parallelized in the usual way by executing multiple independent simulations in multiple processors. The training operation can also be parallelized to a great extent. For example, one may parallelize the gradient iteration

$$\bar{r} := \bar{r} - \gamma \sum_{k=1}^N \nabla \tilde{J}_{\bar{\mu}}(i_k, \bar{r}) (\tilde{J}_{\bar{\mu}}(i_k, \bar{r}) - c(i_k)),$$

that is used for training [cf. Eq. (3.20)]. There are two possibilities here:

- (1) To assign different components of  $\bar{r}$  to different processors and to execute the component updates in parallel.
- (2) To parallelize the computation of the sample gradient

$$\sum_{k=1}^N \nabla \tilde{J}_{\bar{\mu}}(i_k, \bar{r}) (\tilde{J}_{\bar{\mu}}(i_k, \bar{r}) - c(i_k))$$

in the gradient iteration, by assigning different blocks of state-sample cost pairs to different processors.

There are several straightforward versions of these parallelization methods, and it is also valid to use asynchronous versions of them ([BeT89a], Ch. 7).

There is still another parallelization approach for the training process. It is possible to divide the state space  $S$  into several subsets  $S_m$ ,  $m = 1, \dots, M$ , and to calculate a different approximation  $\tilde{J}_{\bar{\mu}}(i, \bar{r}_m)$  for each subset  $S_m$ . In other words, the parameter vector  $\bar{r}_m$  that is used to calculate the approximate cost  $\tilde{J}_{\bar{\mu}}(i, \bar{r}_m)$  depends on the subset  $S_m$  to which state  $i$  belongs. The parameters  $\bar{r}_m$  can be obtained by a parallel training process using the applicable simulation data, that is, the state-sample cost pairs  $(i, c(i))$  with  $i \in S_m$ . Note that the extreme case where each set  $S_m$  corresponds to a single state, corresponds to the case where there is no approximation.

## 2.4 NOTES, SOURCES, AND EXERCISES

The analysis of the stochastic shortest path problems of Section 2.1 is taken from [BeT89a] and [BeT91b]. The latter reference proves the results

shown here under a more general compactness assumption on  $U(i)$  and continuity assumption on  $g(i, u)$  and  $p_{ij}(u)$ . Stochastic shortest path problems were first formulated and analyzed in [EaZ62] under the assumption  $q(i, u) > 0$  for all  $i = 1, \dots, n$  and  $u \in U(i)$ . Finitely terminating value iteration algorithms have been developed for several types of stochastic shortest path problems (see [NgP88], [PoF92], [PsT93], [Psi93a]). The use of a Dijkstra-like algorithm for continuous space shortest path problems involving a consistently improving policy was proposed in [Psi93a] (see Exercise 2.10). A Dijkstra-like algorithm was also proposed for another class of problems involving a consistently improving policy in [NgP88]. The algorithm of Exercise 2.11 is new in the general form given here. The error bound on the performance of approximate policy iteration (Prop. 2.1), which was developed in collaboration with J. Tsitsiklis, is also new. Two-player dynamic game versions of the stochastic shortest path problem have been discussed in [PoA69] (see also the survey [RaF91]).

Several approximation methods that are not based on simulation were given in [SeS85]. The interest in simulation-based methods is relatively recent. In the artificial intelligence community, these methods are collectively referred to as *reinforcement learning*. In the engineering community, these methods are also referred to as *neuro-dynamic programming*. The method of temporal differences was proposed in an influential paper by Sutton [Sut88]. *Q*-learning was proposed by Watkins [Wat89]. A convergence proof of *Q*-learning under fairly weak assumptions was given in [Psi94]; see also [JJS93], which discusses the convergence of  $\text{TD}(\lambda)$ . For a nice survey of related methods, which also includes historical references, see [BBS93]. A variant of *Q*-learning is the method of advantage updating developed in [Bai93], [Bai94], [Bai95], and [HBK94] (see Exercise 2.18). The material on feature-based aggregation has been adapted from [TsV94]. The two-sample simulation-based gradient method for minimizing the error in Bellman's equation was proposed in [Ber95b]; see also [HBK94]. The optimistic policy iteration method was used in an application to backgammon described in [Tes92].

---

## EXERCISES

---

### 2.1

Suppose that you want to travel from a start point  $S$  to a destination point  $D$  in minimum average time. There are two options:

- (1) Use a direct route that requires  $a$  time units.

- (2) Take a potential shortcut that requires  $b$  time units to go to an intermediate point  $I$ . From  $I$  you can either go to the destination  $D$  in  $c$  time units or return to the start (this will take an additional  $b$  time units). You will find out the value of  $c$  once you reach the intermediate point  $I$ . What you know a priori is that  $c$  has one of the  $m$  values  $c_1, \dots, c_m$  with corresponding probabilities  $p_1, \dots, p_m$ . Consider two cases: (i) The value of  $c$  is constant over time, and (ii) The value of  $c$  changes each time you return to the start independently of the value at the previous time periods.
  - (a) Formulate the problem as a stochastic shortest path problem. Write Bellman's equation and characterize the optimal stationary policies as best as you can in terms of the given problem data. Solve the problem for the case  $a = 2$ ,  $b = 1$ ,  $c_1 = 0$ ,  $c_2 = 5$ ,  $p_1 = 0.5$ ,  $p_2 = 0.5$ .
  - (b) Formulate as a stochastic shortest path problem the variation where once you reach the intermediate point  $I$ , you can wait there. Each  $d$  time units the value of  $c$  changes to one of the values  $c_1, \dots, c_m$  with probabilities  $p_1, \dots, p_m$ , independently of its earlier values. Each time the value of  $c$  changes, you have the option of waiting for an extra  $d$  units, returning to the start, or going to the destination. Characterize the optimal stationary policies as best as you can.

### 2.2

A gambler engages in a game of successive coin flipping over an infinite horizon. He wins one dollar each time heads comes up, and loses  $m > 0$  dollars each time two successive tails come up (so the sequence TTTT loses  $3m$  dollars). The gambler at each time period either flips a fair coin or else cheats by flipping a two-headed coin. In the latter case, however, he gets caught with probability  $p > 0$  before he flips the coin, the game terminates, and the gambler keeps his earnings thus far. The gambler wishes to maximize his expected earnings.

- (a) View this as a stochastic shortest path problem and identify all proper and all improper policies.
- (b) Identify a critical value  $\bar{m}$  such that if  $m > \bar{m}$ , then all improper policies give an infinite cost for some initial state.
- (c) Assume that  $m > \bar{m}$ , and show that it is then optimal to try to cheat if the last flip was tails and to play fair otherwise.
- (d) Show that if  $m < \bar{m}$  it is optimal to always play fair.

### 2.3

Consider a stochastic shortest path problem where all stationary policies are proper. Show that for every policy  $\pi$  there exists an  $m > 0$  such that

$$P(x_m = t \mid x_0 = i, \pi) > 0$$

for all  $i = 1, \dots, n$ . *Abbreviated Proof:* Assume the contrary; that is, there exists a nonstationary  $\pi = \{\mu_0, \mu_1, \dots\}$  and an initial state  $i$  such that  $P(x_m = t \mid x_0 = i, \pi) = 0$  for all  $m$ . For each state  $j$ , let  $m(j)$  be the minimum integer  $m$  such that state  $j$  is reachable from  $i$  with positive probability under policy  $\pi$ ; that is,

$$m(j) = \min\{m \mid P(x_m = j \mid x_0 = i, \pi) > 0\},$$

where we adopt the convention that  $m(j) = \infty$  if  $j$  is not reachable from  $i$  under  $\pi$ , i.e.,  $P(x_m = j \mid x_0 = i, \pi) = 0$  for all  $m$ . In particular, we have  $m(i) = 0$  and  $m(t) = \infty$ . Consider any stationary policy  $\mu$  such that  $\mu(j) = \mu_{m(j)}(j)$  for all  $j$  with  $m(j) < \infty$ . Argue that for any two states  $j$  and  $j'$  with  $m(j) < \infty$  and  $m(j') = \infty$ , we have  $p_{j,j'}(\mu(j)) = 0$ . Thus, states  $j'$  with  $m(j') = \infty$  (including  $t$ ) are not reachable under the stationary policy  $\mu$  from states  $j$  with  $m(j) < \infty$  (including  $i$ ), thereby contradicting the hypothesis.

## 2.4

Consider the stochastic shortest path problem, and assume that  $g(i, u) \leq 0$  for all  $i$  and  $u \in U(i)$ . Show that either the optimal cost is  $-\infty$  for some initial state, or else, under every policy, the system eventually enters with probability one a set of cost-free states and never leaves that set thereafter.

## 2.5

Consider the stochastic shortest path problem, and assume that there exists at least one proper policy. Proposition 1.2 implies that if, for each improper policy  $\mu$ , we have  $J_\mu(i) = \infty$  for at least one state  $i$ , then there is no improper policy  $\mu'$  such that  $J_{\mu'}(j) = -\infty$  for at least one state  $j$ . Give an alternative proof of this fact that does not use Prop. 1.2. *Hint:* Suppose that there exists an improper policy  $\mu'$  such that  $J_{\mu'}(j) = -\infty$  for at least one state  $j$ . Combine this policy with a proper policy to produce another improper policy  $\mu''$  for which  $J_{\mu''}(i) < \infty$  for all  $i$ .

## 2.6 (Gauss-Seidel Method for Stochastic Shortest Paths)

Show that the Gauss-Seidel version of the value iteration method for stochastic shortest paths converges under the same assumptions as the ordinary method (Assumptions 1.1 and 1.2). *Hint:* Consider two functions  $\underline{J}$  and  $\bar{J}$  that differ by a constant from  $J^*$  at all states except the destination, and are such that  $\underline{J} \leq T\underline{J}$  and  $T\bar{J} \leq \bar{J}$ .

## 2.7 (Sequential Space Decomposition)

Consider the stochastic shortest path problem, and suppose that there is a finite sequence of subsets of states  $S_1, S_2, \dots, S_M$  such that each of the states  $i = 1, \dots, n$  belongs to one and only one of these subsets, and the following property holds:

For all  $m = 1, \dots, M$  and states  $i \in S_m$ , the successor state  $j$  is either the termination state  $t$  or else belongs to one of the subsets  $S_m, S_{m-1}, \dots, S_1$  for all choices of the control  $u \in U(i)$ .

- (a) Show that the solution of this problem decomposes into the solution of  $M$  stochastic shortest path problems, each involving the states in a subset  $S_m$  plus a termination state.
- (b) Show also that a finite horizon problem with  $N$  stages can be viewed as a stochastic shortest path problem with the property given above.

## 2.8

Consider a stochastic shortest path problem under Assumptions 1.1 and 1.2. Assuming  $p_{ii}(u) < 1$  for all  $i \neq t$  and  $u \in U(i)$ , consider another stochastic shortest path problem that has transition probabilities  $p_{ij}(u)/(1-p_{ii}(u))$  for all  $i \neq t$  and  $j \neq i$ , and costs

$$\tilde{g}(i, u) = g(i, u) + \frac{g(i, u)p_{ii}(u)}{1-p_{ii}(u)}.$$

- (a) Show that the two problems are equivalent in that they have the same optimal costs and policies. How would you deal with the case where  $p_{ii}(u) = 1$  for some  $i \neq t$  and  $u \in U(i)$ ?
- (b) Interpret  $\tilde{g}(i, u)$  as an average cost incurred between arrival to state  $i$  and transition to a state  $j \neq i$ .

## 2.9 (Simplifications for Uncontrollable State Components)

Consider a stochastic shortest path problem under Assumptions 1.1 and 1.2, where the state is a composite  $(i, y)$  of two components  $i$  and  $y$ , and the evolution of the main component  $i$  can be directly affected by the control  $u$ , but the evolution of the other component  $y$  cannot (cf. Section 1.4 and Exercise 1.22 of Vol. I). In particular, we assume that given the state  $(i, y)$  and the control  $u$ , the next state  $(j, z)$  is determined as follows: first  $j$  is generated according to transition probabilities  $p_{ij}(u, y)$ , and then  $z$  is generated according to conditional probabilities  $p(z \mid j)$  that depend on the main component  $j$  of the new state. We also assume that the cost per stage is  $g(i, y, u, j)$  and does not depend on the second component  $z$  of the next state  $(j, z)$ . For functions  $\hat{J}(i)$ ,  $i = 1, \dots, n$ , consider the mapping

$$(\hat{T}\hat{J})(i) = \sum_y p(y \mid i) \left( \min_{u \in U(i, y)} \sum_{j=1}^n p_{ij}(u, y) (g(i, y, u, j) + \hat{J}(j)) \right).$$

and the corresponding mapping of a stationary policy  $\mu$ ,

$$(\hat{T}_\mu J)(i) = \sum_y p(y | i) \sum_{j=1}^n p_{ij}(\mu(i, y), y) (g(i, y, \mu(i, y), j) + J(j)).$$

- (a) Show that  $\hat{J} := \hat{T}\hat{J}$  is a form of Bellman's equation and can be used to characterize the optimal stationary policies. *Hint.* Given  $J(i, y)$ , define  $\hat{J}(i) = \sum_y p(y | i) J(i, y)$ .
- (b) Show the validity of a modified value iteration algorithm that starts with an arbitrary function  $\hat{J}$  and sequentially produces  $\hat{T}\hat{J}$ ,  $\hat{T}^2\hat{J}$ , ...
- (c) Show the validity of a modified policy iteration algorithm whose typical iteration, given the current policy  $\mu^k(i, y)$ , consists of two steps: (1) The policy evaluation step, which computes the unique function  $\hat{J}_{\mu^k}$  that solves the linear system of equations  $\hat{J}_{\mu^k} = \hat{T}_{\mu^k} \hat{J}_{\mu^k}$ . (2) The policy improvement step, which computes the improved policy  $\mu^{k+1}(i, y)$  from the equation  $\hat{T}_{\mu^{k+1}} \hat{J}_{\mu^k} = \hat{T} \hat{J}_{\mu^k}$ .
- (d) Suppose that  $y$  is the only source of randomness in the problem; that is, for each  $(i, y, u)$ , there is a state  $j$  such that  $p_{ij}(u, y) = 1$ . Justify the use of the following single sample version of value iteration (cf. the  $Q$ -learning algorithm of Section 7.6.2)

$$\hat{J}(i) := \hat{J}(i) + \gamma \left( \min_{u \in U(i, y)} [g(i, y, u, j) + \hat{J}(j)] - \hat{J}(i) \right).$$

Here, given  $i$ , we generate  $y$  according to the probability distribution  $p(y | i)$ , and  $j$  is the unique state corresponding to  $(i, y, u)$ .

## 2.10 (Discretized Shortest Path Problems [Tsi93a])

Suppose that the states are the grid points of a grid on the plane. The set of neighbors of each grid point  $x$  is denoted  $U(x)$  and includes between two and four grid points. At each grid point  $x$ , we have two options:

- (1) Choose two neighbors  $x^+, x^- \in U(x)$  and a probability  $p \in [0, 1]$ , pay a cost  $g(x) \sqrt{p^2 + (1-p)^2}$ , and move to  $x^+$  or to  $x^-$  with probability  $p$  or  $1-p$ , respectively. Here  $g$  is a function such that  $g(x) > 0$  for all  $x$ .
- (2) Stop and pay a cost  $t(x)$ .

Show that there exists a consistently improving optimal policy for this problem. *Note:* This problem can be used to model discretized versions of deterministic continuous space 2-dimensional shortest path problems. (Compare also with Exercise 6.11 in Chapter 6 of Vol. I.)

## 2.11 (Dijkstra's Algorithm and Consistently Improving Policies)

Consider the stochastic shortest path problem under Assumptions 1.1 and 1.2, and assume that there exists a consistently improving optimal stationary policy.

- (a) Show that the transition probability graph of this policy is acyclic.
- (b) Consider the following algorithm, which maintains two subsets of states  $P$  and  $L$ , and a function  $J$  defined on the state space. (To relate the algorithm with Dijkstra's method of Section 2.3.1 of Vol. I, associate  $J$  with the node labels,  $L$  with the OPEN list, and  $P$  with the subset of nodes that have already exited the OPEN list.) Initially,  $P = \emptyset$ ,  $L = \{i\}$ , and

$$J(i) = \begin{cases} \infty & \text{if } i = 1, \dots, n, \\ 0 & \text{if } i = i. \end{cases}$$

At the typical iteration, select a state  $j^*$  from  $L$  such that

$$j^* = \arg \min_{j \in L} J(j).$$

(If  $L$  is empty the algorithm terminates.) Remove  $j^*$  from  $L$  and place it in  $P$ . In addition, for all  $i \notin P$  such that there exists a  $u \in U(i)$  with  $p_{ij^*}(u) > 0$ , and

$$p_{ij}(u) = 0 \quad \text{for all } j \notin P,$$

define

$$\hat{U}(i) = \{u \in U(i) \mid p_{ij^*}(u) > 0 \text{ and } p_{ij}(u) = 0 \text{ for all } j \notin P\},$$

set

$$J(i) := \min \left[ J(i), \min_{u \in \hat{U}(i)} \left[ g(i, u) + \sum_{j \in P} p_{ij}(u) J(j) \right] \right],$$

and place  $i$  in  $L$  if it is not already there. Show that the algorithm is well defined in the sense that  $\hat{U}(i)$  is nonempty and the set  $L$  does not become empty until all states are in  $P$ . Furthermore, each state  $j$  is removed from  $L$  once, and at the time it is removed, we have  $J(j) = J^*(j)$ .

## 2.12 (Alternative Assumptions for Prop. 1.2)

Consider a variation of Assumption 1.2, whereby we assume that  $g(i, u) \geq 0$  for all  $i$  and  $u \in U(i)$ , and that there exists an optimal proper policy. Prove the assertions of Prop. 1.2, except that, in part (a), uniqueness of the solution of Bellman's equation should be shown within the set  $\mathfrak{R}^+ = \{J \mid J \geq 0\}$  (rather than within  $\mathfrak{R}^0$ ), and the vector  $J$  in part (b) must belong to  $\mathfrak{R}^+$ .

*Hint:* Proposition 1.1 is not valid, so a somewhat different proof is needed. Complete the details of the following argument. The assumptions guarantee that  $J^*$  is finite and  $J^* \in \mathfrak{R}^t$ . [We have  $J^* \geq 0$  because  $g(i, u) \geq 0$ , and  $J^*(i) < \infty$  because a proper policy exists.] The idea now is to show that  $J^* \geq T J^*$ , and then to choose  $\mu$  such that  $T_\mu J^* = T J^*$  and show that  $\mu$  is optimal and proper. Let  $\pi = \{\mu_0, \mu_1, \dots\}$  be a policy. We have for all  $i$ ,

$$J_\pi(i) = g(i, \mu_0(i)) + \sum_{j=1}^n p_{ij}(\mu_0(i)) J_{\pi_1}(j)$$

where  $\pi_1$  is the policy  $\{\mu_1, \mu_2, \dots\}$ . Since  $J_{\pi_1} \geq J^*$ , we obtain

$$J_\pi(i) \geq g(i, \mu_0(i)) + \sum_{j=1}^n p_{ij}(\mu_0(i)) J^*(j) = (T_{\mu_0} J^*)(i) \geq (T J^*)(i).$$

Taking the infimum over  $\pi$  in the preceding equation, we obtain

$$J^* \geq T J^*. \quad (4.1)$$

Let  $\mu$  be such that  $T_\mu J^* = T J^*$ . From Eq. (4.1), we have  $J^* \geq T_\mu J^*$ , and using the monotonicity of  $T_\mu$ , we obtain

$$J^* \geq T_\mu^N J^* = P_\mu^N J^* + \sum_{k=0}^{N-1} P_\mu^k y_\mu \geq \sum_{k=0}^{N-1} P_\mu^k y_\mu, \quad N \geq 1. \quad (4.2)$$

By taking limit superior as  $N \rightarrow \infty$ , we obtain  $J^* \geq J_\mu$ . Therefore,  $\mu$  is an optimal proper policy, and  $J^* = J_\mu$ . Since  $\mu$  was selected so that  $T_\mu J^* = T J^*$ , we obtain, using  $J^* = J_\mu$  and  $J_\mu = T_\mu J_\mu$ , that  $J^* = T J^*$ . For the rest of the proof, use the vector  $\delta c$  similar to the proof of Prop. 1.2.

### 2.13 (A Contraction Counterexample)

Consider a stochastic shortest path problem with a single state 1, in addition to the termination state  $t$ . At state 1 there are two controls  $u$  and  $u'$ . Under  $u$  the cost is 1 and the system remains in state 1 for one more stage; under  $u'$  the cost is 2 and the system moves to  $t$ . Show that Assumptions 1.1 and 1.2 are satisfied, but  $T$  is not a contraction mapping with respect to any norm.

### 2.14 (Contraction Property – All Stationary Policies are Proper)

Assume that all stationary policies are proper. Show that the mappings  $T$  and  $T_\mu$  are contraction mappings with respect to some weighted sup norm

$$\|J\|_c = \max_{i=1, \dots, n} \frac{1}{v_i} |J(i)|,$$

where  $v$  is a vector whose components  $v_1, \dots, v_n$  are positive.

*Abbreviated proof (from [BuTs9a], p. 325; see also [Tsce90]):* Partition the state space as follows. Let  $S_1 = \{1\}$  and for  $k = 2, 3, \dots$ , define sequentially

$$S_k = \left\{ i \mid i \notin S_1 \cup \dots \cup S_{k-1} \text{ and } \min_{u \in P(i)} \max_{j \in S_1 \cup \dots \cup S_{k-1}} p_{ij}(u) > 0 \right\}.$$

Let  $S_m$  be the last of these sets that is nonempty. We claim that the sets  $S_k$  cover the entire state space, that is,  $\bigcup_{k=1}^m S_k = S$ . To see this, suppose that the set  $S_\infty = \{i \mid i \notin \bigcup_{k=1}^m S_k\}$  is nonempty. Then for each  $i \in S_\infty$ , there exists some  $u_i \in U(i)$  such that  $p_{ij}(u_i) = 0$  for all  $j \notin S_\infty$ . Take any  $\mu$  such that  $\mu(i) = u_i$  for all  $i \in S_\infty$ . The stationary policy  $\mu$  satisfies  $[P_\mu^N]_{ij} = 0$  for all  $i \in S_\infty$ ,  $j \notin S_\infty$ , and  $N$ , and therefore cannot be proper. This contradicts the hypothesis.

We will choose a vector  $v > 0$  so that  $T$  is a contraction mapping with respect to  $\|\cdot\|_v$ . We will take the  $i$ th component  $v_i$  to be the same for states  $i$  in the same set  $S_k$ . In particular, we will choose the components  $v_i$  of the vector  $v$  by

$$v_i = y_k \quad \text{if} \quad i \in S_k,$$

where  $y_1, \dots, y_m$  are appropriately chosen scalars satisfying

$$1 = y_1 < y_2 < \dots < y_m. \quad (4.3)$$

Let

$$c = \min_{k=2, \dots, m} \min_{\mu \in M} \min_{i \in S_k} \sum_{j \in S_1 \cup \dots \cup S_{k-1}} [P_\mu]_{ij}, \quad (4.4)$$

and note that  $0 < c \leq 1$ . We will show that it is sufficient to choose  $y_2, \dots, y_m$  so that for some  $\gamma < 1$ , we have

$$\frac{y_m}{y_k} (1 - c) + \frac{y_{k+1}}{y_k} c \leq \gamma < 1, \quad k = 2, \dots, m, \quad (4.5)$$

and then show that such a choice of  $y_2, \dots, y_m$  exists.

Indeed, for vectors  $J$  and  $J'$  in  $\mathfrak{R}^n$ , let  $\mu$  be such that  $T_\mu J = T J$ . Then we have for all  $i$ ,

$$\begin{aligned} (T J')(i) - (T J)(i) &= (T J')(i) - (T_\mu J)(i) \\ &\leq (T_\mu J')(i) - (T_\mu J)(i) \\ &\quad + \sum_{j=1}^n p_{ij}(\mu(i)) (J'(j) - J(j)). \end{aligned} \quad (4.6)$$

Let  $k(j)$  be such that  $j$  belongs to the set  $S_{k(j)}$ . Then we have for any constant  $c_j$ ,

$$\|J' - J\|_c \leq c \quad \Leftrightarrow \quad J'(j) - J(j) \leq c y_{k(j)}, \quad j = 2, \dots, n.$$

and Eq. (4.6) implies that for all  $i$ ,

$$\begin{aligned} \frac{(TJ')(i) - (TJ)(i)}{c y_{k(i)}} &\leq \frac{1}{y_{k(i)}} \sum_{j=1}^n p_{ij}(\mu(i)) y_{k(j)} \\ &\leq \frac{y_{k(i)-1}}{y_{k(i)}} \sum_{j \in S_1 \cup \dots \cup S_{k(i)-1}} p_{ij}(\mu(i)) \\ &\quad + \frac{y_m}{y_{k(i)}} \sum_{j \in S_{k(i)+1} \cup \dots \cup S_m} p_{ij}(\mu(i)) \\ &= \left( \frac{y_{k(i)-1}}{y_{k(i)}} - \frac{y_m}{y_{k(i)}} \right) \sum_{j \in S_1 \cup \dots \cup S_{k(i)-1}} p_{ij}(\mu(i)) + \frac{y_m}{y_{k(i)}} \\ &\leq \left( \frac{y_{k(i)-1}}{y_{k(i)}} - \frac{y_m}{y_{k(i)}} \right) c + \frac{y_m}{y_{k(i)}} \leq \gamma, \end{aligned}$$

where the second inequality follows from Eq. (4.3), the third inequality uses Eq. (4.4) and the fact  $y_{k(i)-1} = y_m \leq 0$ , and the last inequality follows from Eq. (4.5). Thus, we have

$$\frac{(TJ')(i) - (TJ)(i)}{v_i} \leq c\gamma, \quad i = 1, \dots, n,$$

and we obtain

$$\max_i \frac{(TJ')(i) - (TJ)(i)}{v_i} \leq c\gamma,$$

or

$$\|TJ - TJ'\|_v \leq c\gamma, \quad \text{for all } J, J' \in \mathfrak{W}^n \text{ with } \|J - J'\|_v \leq c.$$

It follows that  $T$  is a contraction mapping with respect to  $\|\cdot\|_v$ .

We now show how to choose the scalars  $y_1, y_2, \dots, y_m$  so that Eqs. (4.3) and (4.5) hold. Let  $y_0 = 0$ ,  $y_1 = 1$ , and suppose that  $y_1, y_2, \dots, y_k$  have been chosen. If  $c = 1$ , we choose  $y_{k+1} = y_k + 1$ . If  $c < 1$ , we choose  $y_{k+1}$  to be

$$y_{k+1} = \frac{1}{2}(y_k + M_k),$$

where

$$M_k = \min_{1 \leq i \leq k} \left[ y_i + \frac{c}{1-c}(y_i - y_{i-1}) \right].$$

Using the fact

$$M_{k+1} = \min \left\{ M_k, y_{k+1} + \frac{c}{1-c}(y_{k+1} - y_k) \right\},$$

it is seen by induction that for all  $k$ ,

$$y_k < y_{k+1} < M_{k+1}.$$

In particular, we have

$$y_m < M_m = \min_{1 \leq i \leq m} \left[ y_i + \frac{c}{1-c}(y_i - y_{i-1}) \right],$$

which implies Eq. (4.5).

## 2.15 (Multiple State Visits in Monte Carlo Simulation)

Argue that the Monte-Carlo simulation formula

$$J_\mu(i) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M c(i, m)$$

[cf. Eq. (3.1)] is valid even if a state may be revisited within the same sample trajectory. Hint: Suppose the  $M$  cost samples are generated from  $N$  trajectories, and that the  $k$ th trajectory involves  $n_k$  visits to state  $i$  and generates  $n_k$  corresponding cost samples. Denote  $m_k = n_1 + \dots + n_k$ . Write

$$\begin{aligned} \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{m=1}^M c(i, m) &= \lim_{N \rightarrow \infty} \frac{\frac{1}{N} \sum_{k=1}^N \sum_{m=m_{k-1}+1}^{m_k} c(i, m)}{\frac{1}{N}(n_1 + \dots + n_N)} \\ &= \frac{E \left\{ \sum_{m=m_{k-1}+1}^{m_k} c(i, m) \right\}}{E\{n_k\}}, \end{aligned}$$

and argue that

$$E \left\{ \sum_{m=m_{k-1}+1}^{m_k} c(i, m) \right\} = E\{n_k\} J_\mu(i),$$

(or see [Ros83b], Cor. 7.2.3 for a closely related result).

## 2.16 (Multistage Lookahead Policy Iteration)

- (a) Consider the stochastic shortest path problem under Assumptions 1.1 and 1.2. Let  $\mu$  be a stationary policy, let  $J$  be a function such that  $TJ \leq J \leq J_\mu$  ( $J = J_\mu$  is one possibility), and let  $\{\bar{\mu}_0, \bar{\mu}_1, \dots, \bar{\mu}_{N-1}\}$  be an optimal policy for the  $N$ -stage problem with terminal cost function  $J$ , i.e.

$$T_{\bar{\mu}_k} T^{N-k-1} J = T^{N-k} J, \quad k = 0, 1, \dots, N-1.$$

- (a) Show that

$$J_{\bar{\mu}_k} \leq J_\mu, \quad \text{for all } k = 0, 1, \dots, N-1.$$

Hint: First show that  $T^{k+1} J \leq T^k J \leq J$  for all  $k$ , and then show that the hypothesis  $T_{\bar{\mu}_k} T^{N-k-1} J = T^{N-k} J$  implies that  $J_{\bar{\mu}_k} \leq T^{N-k-1} J$ .

- (b) Use part (a) to show the validity of the multistage policy iteration algorithm discussed in Section 2.3.3.

### 2.17 (Viewing Q-Factors as Optimal Costs)

Consider the stochastic shortest path problem under Assumptions 1.1 and 1.2. Show that the  $Q$ -factors  $Q(i, u)$  can be viewed as state costs associated with a modified stochastic shortest path problem. Use this fact to show that the  $Q$ -factors  $Q(i, u)$  are the unique solution of the system of equations

$$Q(i, u) = \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right).$$

*Hint:* Introduce a new state for each pair  $(i, u)$ , with transition probabilities  $p_{ij}(u)$  to the states  $j = 1, \dots, n, t$ .

### 2.18 (Advantage Updating)

Consider the optimal  $Q$ -factors  $Q^*(i, u)$  of the stochastic shortest path problem under Assumptions 1.1 and 1.2. Define the *advantage function* by

$$A^*(i, u) = \min_{u' \in U(i)} Q^*(i, u') - Q^*(i, u).$$

- (a) Show that  $A^*(i, u)$  together with the optimal costs  $J^*(i)$  solve uniquely the system of equations

$$J^*(i) = A^*(i, u) + \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + J^*(j)),$$

$$\max_{u \in U(i)} A^*(i, u) = 0, \quad i = 1, \dots, n.$$

- (b) Introduce approximating functions  $\hat{A}(i, u, r)$  and  $\hat{J}(i, r)$ , and derive a gradient method aimed at minimizing the sum of the squared errors of the Bellman-like equations of part (a) (cf. Section 2.3.3).

## Undiscounted Problems

### Contents

3.1. Unbounded Costs per Stage . . . . .	p. 134
3.2. Linear Systems and Quadratic Cost . . . . .	p. 150
3.3. Inventory Control . . . . .	p. 153
3.4. Optimal Stopping . . . . .	p. 155
3.5. Optimal Gambling Strategies . . . . .	p. 160
3.6. Nonstationary and Periodic Problems . . . . .	p. 167
3.7. Notes, Sources, and Exercises . . . . .	p. 172

In this chapter we consider total cost infinite horizon problems where we allow costs per stage that are unbounded above or below. Also, the discount factor  $\alpha$  does not have to be less than one. The complications resulting are substantial, and the analysis required is considerably more sophisticated than the one given thus far. We also consider applications of the theory to important classes of problems. The problem section touches on several related topics.

### 3.1 UNBOUNDED COSTS PER STATE

In this section we consider the total cost infinite horizon problem of Section 1.1 under one of the following two assumptions.

**Assumption P: (Positivity)** The cost per stage  $g$  satisfies

$$0 \leq g(x, u, w), \quad \text{for all } (x, u, w) \in S \times C \times D. \quad (1.1)$$

**Assumption N: (Negativity)** The cost per stage  $g$  satisfies

$$g(x, u, w) \leq 0, \quad \text{for all } (x, u, w) \in S \times C \times D. \quad (1.2)$$

Problems corresponding to Assumption P are sometimes referred to in the research literature as *negative DP problems*. This name was used in the original reference [Str66], where the problem of maximizing the infinite sum of negative rewards per stage was considered. Similarly, problems corresponding to Assumption N are sometimes referred to as *positive DP problems* [Bla65], [Str66]. Assumption N arises in problems where there is a nonnegative reward per stage and the total expected reward is to be maximized.

Note that when  $\alpha < 1$  and  $g$  is either bounded above or below, we may add a suitable scalar to  $g$  in order to satisfy Eq. (1.1) or Eq. (1.2), respectively. An optimal policy will not be affected by this change since, in view of the presence of the discount factor, the addition of a constant  $r$  to  $g$  merely adds  $(1 - \alpha)^{-1}r$  to the cost associated with every policy.

One complication arising from unbounded costs per stage is that, for some initial states  $x_0$  and some genuinely interesting admissible policies

$\pi = \{\mu_0, \mu_1, \dots\}$ , the cost  $J_\pi(x_0)$  may be  $\infty$  (in the case of Assumption P) or  $-\infty$  (in the case of Assumption N). Here is an example:

#### Example 1.1

Consider the scalar system

$$x_{k+1} = \beta x_k + u_k, \quad k = 0, 1, \dots,$$

where  $x_k \in \mathbb{R}$  and  $u_k \in \mathbb{R}$ , for all  $k$ , and  $\beta$  is a positive scalar. The control constraint is  $|u_k| \leq 1$ , and the cost is

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^k |x_k|.$$

Consider the policy  $\tilde{\pi} = \{\tilde{\mu}_0, \tilde{\mu}_1, \dots\}$ , where  $\tilde{\mu}(x) = 0$  for all  $x \in \mathbb{R}$ . Then

$$J_{\tilde{\pi}}(x_0) = \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^k / \beta^k |x_0|,$$

and hence

$$J_{\tilde{\pi}}(x_0) = \begin{cases} 0 & \text{if } x_0 = 0 \\ \infty & \text{if } x_0 \neq 0 \end{cases} \quad \text{if } \alpha\beta \geq 1,$$

while

$$J_{\tilde{\pi}}(x_0) = \frac{|x_0|}{1 - \alpha\beta} \quad \text{if } \alpha\beta < 1.$$

Note a peculiarity here: if  $\beta > 1$  the state  $x_k$  diverges to  $\infty$  or to  $-\infty$ , but if the discount factor is sufficiently small ( $\alpha < 1/\beta$ ), the cost  $J_{\tilde{\pi}}(x_0)$  is finite.

It is also possible to verify that when  $\beta > 1$  and  $\alpha\beta \geq 1$  the optimal cost  $J^*(x_0)$  is equal to  $\infty$  for  $|x_0| \geq 1/(\beta - 1)$  and is finite for  $|x_0| < 1/(\beta - 1)$ . The problem here is that when  $\beta > 1$  the system is unstable, and in view of the restriction  $|u_k| \leq 1$  on the control, it may not be possible to force the state near zero once it has reached sufficiently large magnitude.

The preceding example shows that there is not much that can be done about the possibility of the cost function being infinite for some policies. To cope with this situation, we conduct our analysis with the notational understanding that the costs  $J_\pi(x_0)$  and  $J^*(x_0)$  may be  $\infty$  (or  $-\infty$ ) under Assumption P (or N, respectively) for some initial states  $x_0$  and policies  $\pi$ . In other words, we consider  $J_\pi(\cdot)$  and  $J^*(\cdot)$  to be extended real-valued functions. In fact, the entire subsequent analysis is valid even if the cost  $g(x, u, w)$  is  $\infty$  or  $-\infty$  for some  $(x, u, w)$ , as long as Assumption P or Assumption N holds.

The line of analysis of this section is fundamentally different from the one of the discounted problem of Section 1.2. For the latter problem, the analysis was based on ignoring the “tails” of the cost sequences. In

this section, the tails of the cost sequences may not be small, and for this reason, the control is much more focused on affecting the long-term behavior of the state. For example, let  $\alpha = 1$ , and assume that the stage cost at all states is nonzero except for a cost-free and absorbing termination state. Then, a primary task of control under Assumption P (or Assumption N) is roughly to bring the state of the system to the termination state or to a region where the cost per stage is nearly zero as *quickly* as possible (as *late* as possible, respectively). Note the difference in control objective between Assumptions P and N. It accounts for some strikingly different results under the two assumptions.

### Main Results – Bellman's Equation

We now present results that characterize the optimal cost function  $J^*$ , as well as optimal stationary policies. We also give conditions under which value iteration converges to the optimal cost function  $J^*$ . In the proofs we will often need to interchange expectation and limit in various relations. This interchange is valid under the assumptions of the following theorem.

**Monotone Convergence Theorem:** Let  $P = (p_1, p_2, \dots)$  be a probability distribution over  $S = \{1, 2, \dots\}$ . Let  $\{h_N\}$  be a sequence of extended real-valued functions on  $S$  such that for all  $i \in S$  and  $N = 1, 2, \dots$ ,

$$0 \leq h_N(i) \leq h_{N+1}(i).$$

Let  $h : S \mapsto [0, \infty]$  be the limit function

$$h(i) = \lim_{N \rightarrow \infty} h_N(i).$$

Then

$$\lim_{N \rightarrow \infty} \sum_{i=1}^{\infty} p_i h_N(i) = \sum_{i=1}^{\infty} p_i \lim_{N \rightarrow \infty} h_N(i) = \sum_{i=1}^{\infty} p_i h(i).$$

**Proof:** We have

$$\sum_{i=1}^{\infty} p_i h_N(i) \leq \sum_{i=1}^{\infty} p_i h(i).$$

By taking the limit, we obtain

$$\lim_{N \rightarrow \infty} \sum_{i=1}^{\infty} p_i h_N(i) \leq \sum_{i=1}^{\infty} p_i h(i),$$

so there remains to prove the reverse inequality. For every integer  $M \geq 1$ , we have

$$\lim_{N \rightarrow \infty} \sum_{i=1}^{\infty} p_i h_N(i) \geq \lim_{N \rightarrow \infty} \sum_{i=1}^M p_i h_N(i) := \sum_{i=1}^M p_i h(i),$$

and by taking the limit as  $M \rightarrow \infty$  the reverse inequality follows. **Q.E.D.**

Similar to all the infinite horizon problems considered so far, the optimal cost function satisfies Bellman's equation.

**Proposition 1.1: (Bellman's Equation)** Under either Assumption P or N the optimal cost function  $J^*$  satisfies

$$J^*(x) = \min_{u \in U(x)} E_w \{g(x, u, w) + \alpha J^*(f(x, u, w))\}, \quad x \in S$$

or, equivalently,

$$J^* = TJ^*.$$

**Proof:** For any admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , consider the cost  $J_\pi(x)$  corresponding to  $\pi$  when the initial state is  $x$ . We have

$$J_\pi(x) = E_w \{g(x, \mu_0(x), w) + V_\pi(f(x, \mu_0(x), w))\}, \quad (1.3)$$

where, for all  $x_1 \in S$ ,

$$V_\pi(x_1) = \lim_{N \rightarrow \infty} E_{w_1} \left\{ \sum_{k=1}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}.$$

Thus,  $V_\pi(x_1)$  is the cost from stage 1 to infinity using  $\pi$  when the initial state is  $x_1$ . We have clearly

$$V_\pi(x_1) \geq \alpha J^*(x_1), \quad \text{for all } x_1 \in S.$$

Hence, from Eq. (1.3),

$$\begin{aligned} J_\pi(x) &\geq E_w \{g(x, \mu_0(x), w) + \alpha J^*(f(x, \mu_0(x), w))\} \\ &\geq \min_{u \in U(x)} E_w \{g(x, u, w) + \alpha J^*(f(x, u, w))\}. \end{aligned}$$

Taking the minimum over all admissible policies, we have

$$\begin{aligned} \min_{\pi} J_\pi(x) &= J^*(x) \\ &\geq \min_{u \in U(x)} E_w \{g(x, u, w) + \alpha J^*(f(x, u, w))\} \\ &= (TJ^*)(x). \end{aligned} \quad (1.4)$$

Thus there remains to prove that the reverse inequality also holds. We prove this separately for Assumption N and for Assumption P.

Assume P. The following proof of  $J^* \leq TJ^*$  under this assumption would be considerably simplified if we knew that there exists a  $\mu$  such that  $T_\mu J^* = TJ^*$ . Since in general such a  $\mu$  need not exist, we introduce a positive sequence  $\{\epsilon_k\}$ , and we choose an admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$  such that

$$(T_{\mu_k} J^*)(x) \leq (TJ^*)(x) + \epsilon_k, \quad x \in S, \quad k = 0, 1, \dots$$

Such a choice is possible because we know that, under P, we have  $-\infty < J^*(x)$  for all  $x$ . By using the inequality  $TJ^* \leq J^*$  shown earlier, we obtain

$$(T_{\mu_k} J^*)(x) \leq J^*(x) + \epsilon_k, \quad x \in S, \quad k = 0, 1, \dots$$

Applying  $T_{\mu_{k-1}}$  to both sides of this relation, we have

$$\begin{aligned} (T_{\mu_{k-1}} T_{\mu_k} J^*)(x) &\leq (T_{\mu_{k-1}} J^*)(x) + \alpha \epsilon_k \\ &\leq (TJ^*)(x) + \epsilon_{k-1} + \alpha \epsilon_k \\ &\leq J^*(x) + \epsilon_{k-1} + \alpha \epsilon_k. \end{aligned}$$

Continuing this process, we obtain

$$(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_k} J^*)(x) \leq (TJ^*)(x) + \sum_{i=0}^k \alpha^i \epsilon_i.$$

By taking the limit as  $k \rightarrow \infty$  and noting that

$$J^*(x) \leq J_\pi(x) = \lim_{k \rightarrow \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_k} J_0)(x) \leq \lim_{k \rightarrow \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_k} J^*)(x),$$

where  $J_0$  is the zero function, it follows that

$$J^*(x) \leq J_\pi(x) \leq (TJ^*)(x) + \sum_{i=0}^\infty \alpha^i \epsilon_i, \quad x \in S.$$

Since the sequence  $\{\epsilon_k\}$  is arbitrary, we can take  $\sum_{i=0}^\infty \alpha^i \epsilon_i$  as small as desired, and we obtain  $J^*(x) \leq (TJ^*)(x)$  for all  $x \in S$ . Combining this with the inequality  $J^*(x) \geq (TJ^*)(x)$  shown earlier, the result follows (under Assumption P).

Assume N and let  $J_N$  be the optimal cost function for the corresponding N-stage problem

$$J_N(x_0) = \min_{\pi} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}. \quad (1.5)$$

We first show that

$$J^*(x) = \lim_{N \rightarrow \infty} J_N(x), \quad x \in S. \quad (1.6)$$

Indeed, in view of Assumption N, we have  $J^* \leq J_N$  for all  $N$ , so

$$J^*(x) \leq \lim_{N \rightarrow \infty} J_N(x), \quad x \in S. \quad (1.7)$$

Also, for all  $\pi = \{\mu_0, \mu_1, \dots\}$ , we have

$$E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \geq J_N(x_0),$$

and by taking the limit as  $N \rightarrow \infty$ ,

$$J_\pi(x) \geq \lim_{N \rightarrow \infty} J_N(x), \quad x \in S.$$

Taking the minimum over  $\pi$ , we obtain  $J^*(x) \geq \lim_{N \rightarrow \infty} J_N(x)$ , and combining this relation with Eq. (1.7), we obtain Eq. (1.6).

For every admissible  $\mu$ , we have

$$T_\mu J_N \geq J_{N+1},$$

and by taking the limit as  $N \rightarrow \infty$ , and using the monotone convergence theorem and Eq. (1.6), we obtain

$$T_\mu J^* \geq J^*.$$

Taking the minimum over  $\mu$ , we obtain  $TJ^* \geq J^*$ , which combined with the inequality  $J^* \geq TJ^*$  shown earlier, proves the result under Assumption N. Q.E.D.

Similar to Cor. 2.2.1 in Section 1.2, we have:

**Corollary 1.1.1:** Let  $\mu$  be a stationary policy. Then under Assumption P or N, we have

$$J_\mu(x) = \underset{w}{E} \{g(x, \mu(x), w) + \alpha J_\mu(f(x, \mu(x), w))\}, \quad x \in S$$

or, equivalently,

$$J_\mu = T_\mu J_\mu. \quad (1.8)$$

Contrary to discounted problems with bounded cost per stage, the optimal cost function  $J^*$  under Assumption P or N need not be the unique solution of Bellman's equation. Consider the following example.

**Example 1.2**

Let  $S = [0, \infty)$  (or  $S = (-\infty, 0]$ ) and

$$g(x, u, w) = 0, \quad f(x, u, w) = \frac{x}{\alpha},$$

Then for every  $\beta$ , the function  $J$  given by

$$J(x) = \beta x, \quad x \in S,$$

is a solution of Bellman's equation, so  $T$  has an infinite number of fixed points. Note, however, that there is a unique fixed point within the class of bounded functions, the zero function  $J_0(x) \equiv 0$ , which is the optimal cost function for this problem. More generally, it can be shown by using the following Prop. 1.2 that if  $\alpha < 1$  and there exists a bounded function that is a fixed point of  $T$ , then that function must be equal to the optimal cost function  $J^*$  (see Exercise 3.5). When  $\alpha = 1$ , Bellman's equation may have an infinity of solutions even within the class of bounded functions. This is because if  $\alpha = 1$  and  $J(\cdot)$  is any solution, then for any scalar  $r$ ,  $J(\cdot) + r$  is also a solution.

The optimal cost function  $J^*$ , however, has the property that it is the smallest (under Assumption P) or largest (under Assumption N) fixed point of  $T$  in the sense described in the following proposition.

**Proposition 1.2:**

- (a) Under Assumption P, if  $\tilde{J} : S \mapsto (-\infty, \infty]$  satisfies  $\tilde{J} \geq T\tilde{J}$  and either  $\tilde{J}$  is bounded below and  $\alpha < 1$ , or  $\tilde{J} \geq 0$ , then  $\tilde{J} \geq J^*$ .
- (b) Under Assumption N, if  $\tilde{J} : S \mapsto [-\infty, \infty)$  satisfies  $\tilde{J} \leq T\tilde{J}$  and either  $\tilde{J}$  is bounded above and  $\alpha < 1$ , or  $\tilde{J} \leq 0$ , then  $\tilde{J} \leq J^*$ .

**Proof:** (a) Under Assumption P, let  $r$  be a scalar such that  $\tilde{J}(x) + r \geq 0$  for all  $x \in S$  and if  $\alpha \geq 1$  let  $r = 0$ . For any sequence  $\{\epsilon_k\}$  with  $\epsilon_k > 0$ , let  $\tilde{\pi} = \{\tilde{\mu}_0, \tilde{\mu}_1, \dots\}$  be an admissible policy such that, for every  $x \in S$  and  $k$ ,

$$E_w \left\{ g(x, \mu_k(x), w) + \alpha \tilde{J}(f(x, \mu_k(x), w)) \right\} \leq (T\tilde{J})(x) + \epsilon_k. \quad (1.9)$$

Such a policy exists since  $(T\tilde{J})(x) > -\infty$  for all  $x \in S$ . We have for any initial state  $x_0 \in S$ ,

$$\begin{aligned} J^*(x_0) &= \min_{\pi} \lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\leq \min_{\pi} \liminf_{N \rightarrow \infty} E \left\{ \alpha^N (\tilde{J}(x_N) + r) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\leq \liminf_{N \rightarrow \infty} E \left\{ \alpha^N (\tilde{J}(x_N) + r) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \tilde{\mu}_k(x_k), w_k) \right\}. \end{aligned}$$

Using Eq. (1.9) and the assumption  $\tilde{J} \geq T\tilde{J}$ , we obtain

$$\begin{aligned} &E \left\{ \alpha^N \tilde{J}(x_N) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \tilde{\mu}_k(x_k), w_k) \right\} \\ &= E \left\{ \alpha^N \tilde{J}(f(x_{N-1}, \tilde{\mu}_{N-1}(x_{N-1}), w_{N-1})) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \tilde{\mu}_k(x_k), w_k) \right\} \\ &\leq E \left\{ \alpha^{N-1} \tilde{J}(x_{N-1}) + \sum_{k=0}^{N-2} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} + \alpha^{N-1} \epsilon_{N-1} \\ &\leq E \left\{ \alpha^{N-2} \tilde{J}(x_{N-2}) + \sum_{k=0}^{N-3} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} + \alpha^{N-2} \epsilon_{N-2} \\ &\quad + \alpha^{N-1} \epsilon_{N-1} \\ &\vdots \\ &\leq \tilde{J}(x_0) + \sum_{k=0}^{N-1} \alpha^k \epsilon_k. \end{aligned}$$

Combining these inequalities, we obtain

$$J^*(x_0) \leq \tilde{J}(x_0) + \lim_{N \rightarrow \infty} \left( \alpha^N r + \sum_{k=0}^{N-1} \alpha^k \epsilon_k \right).$$

Since the sequence  $\{\epsilon_k\}$  is arbitrary (except for  $\epsilon_k > 0$ ), we may select  $\{\epsilon_k\}$  so that  $\lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^k \epsilon_k$  is arbitrarily close to zero, and the result follows.

(b) Under Assumption N, let  $r$  be a scalar such that  $\tilde{J}(x) + r \leq 0$  for all  $x \in S$ , and if  $\alpha \geq 1$ , let  $r = 0$ . We have for every initial state  $x_0 \in S$ ,

$$\begin{aligned} J^*(x_0) &= \min_{\pi} \lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\geq \min_{\pi} \limsup_{N \rightarrow \infty} E \left\{ \alpha^N (\tilde{J}(x_N) + r) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\geq \limsup_{N \rightarrow \infty} \min_{\pi} E \left\{ \alpha^N (\tilde{J}(x_N) + r) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}. \end{aligned} \quad (1.10)$$

where the last inequality follows from the fact that for any sequence  $\{h_N(\xi)\}$  of functions of a parameter  $\xi$  we have

$$\min_{\xi} \limsup_{N \rightarrow \infty} h_N(\xi) \geq \limsup_{N \rightarrow \infty} \min_{\xi} h_N(\xi).$$

This inequality follows by writing

$$h_N(\xi) \geq \min_{\xi} h_N(\xi)$$

and by subsequently taking the  $\limsup$  of both sides and the minimum over  $\xi$  of the left-hand side.

Now we have, by using the assumption  $\tilde{J} \leq TJ$ ,

$$\begin{aligned} & \min_{\pi} E \left\{ \alpha^N \tilde{J}(x_N) + \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &= \min_{\pi} E \left\{ \alpha^{N-1} \min_{u_{N-1} \in U(x_{N-1})} E_{w_{N-1}} \left\{ g(x_{N-1}, u_{N-1}, w_{N-1}) \right. \right. \\ &\quad \left. \left. + \alpha \tilde{J}(f(x_{N-1}, u_{N-1}, w_{N-1})) \right\} \right. \\ &\quad \left. + \sum_{k=0}^{N-2} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\geq \min_{\pi} E \left\{ \alpha^{N-1} \tilde{J}(x_{N-1}) + \sum_{k=0}^{N-2} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\} \\ &\vdots \\ &\geq \tilde{J}(x_0). \end{aligned}$$

Using this relation in Eq. (1.10), we obtain

$$J^*(x_0) \geq \tilde{J}(x_0) + \lim_{N \rightarrow \infty} \alpha^N r = \tilde{J}(x_0).$$

Q.E.D.

As before, we have the following corollary:

**Corollary 1.2.1:** Let  $\mu$  be an admissible stationary policy.

- (a) Under Assumption P, if  $\tilde{J} : S \mapsto (-\infty, \infty]$  satisfies  $\tilde{J} \geq T_{\mu} \tilde{J}$  and either  $\tilde{J}$  is bounded below and  $\alpha < 1$ , or  $\tilde{J} \geq 0$ , then  $\tilde{J} \geq J_{\mu}$ .
- (b) Under Assumption N, if  $\tilde{J} : S \mapsto [-\infty, \infty)$  satisfies  $\tilde{J} \leq T_{\mu} \tilde{J}$  and either  $\tilde{J}$  is bounded above and  $\alpha < 1$ , or  $\tilde{J} \leq 0$ , then  $\tilde{J} \leq J_{\mu}$ .

### Conditions for Optimality of a Stationary Policy

Under Assumption P, we have the same optimality condition as for discounted problems with bounded cost per stage.

**Proposition 1.3: (Necessary and Sufficient Condition for Optimality under P)** Let Assumption P hold. A stationary policy  $\mu$  is optimal if and only if

$$TJ^* = T_{\mu}J^*.$$

**Proof:** If  $TJ^* = T_{\mu}J^*$ , Bellman's equation ( $J^* = TJ^*$ ) implies that  $J^* = T_{\mu}J^*$ . From Cor. 1.2.1(a) we then obtain  $J^* \geq J_{\mu}$ , showing that  $\mu$  is optimal. Conversely, if  $J^* = J_{\mu}$ , we have using Cor. 1.1.1,  $TJ^* = J^* = J_{\mu} = T_{\mu}J_{\mu} = T_{\mu}J^*$ . Q.E.D.

Unfortunately, the sufficiency part of the above proposition need not be true under Assumption N; that is, we may have  $TJ^* = T_{\mu}J^*$  while  $\mu$  is not optimal. This is illustrated in the following example.

### Example 1.3

Let  $S = C = (-\infty, 0]$ ,  $U(x) = C$  for all  $x \in S$ , and

$$g(x, u, w) = f(x, u, w) = u,$$

for all  $(x, u, w) \in S \times C \times D$ . Then  $J^*(x) = -\infty$  for all  $x \in S$ , and every stationary policy  $\mu$  satisfies the condition of the preceding proposition. On the other hand, when  $\mu(x) = 0$  for all  $x \in S$ , we have  $J_{\mu}(x) = 0$  for all  $x \in S$ , and hence  $\mu$  is not optimal.

It is worth noting that Prop. 1.3 implies the existence of an optimal stationary policy under Assumption P when  $U(x)$  is a finite set for every  $x \in S$ . This need not be true under Assumption N (see Example 4.1 in Section 3.4).

Under Assumption N, we have a different characterization of an optimal stationary policy.

**Proposition 1.4: (Necessary and Sufficient Condition for Optimality under N)** Let Assumption N hold. A stationary policy  $\mu$  is optimal if and only if

$$TJ_{\mu} = T_{\mu}J_{\mu}. \quad (1.11)$$

**Proof:** If  $TJ_{\mu} = T_{\mu}J_{\mu}$ , then from Cor. 1.1.1 we have  $J_{\mu} = T_{\mu}J_{\mu}$ , so that  $J_{\mu}$  is a fixed point of  $T$ . Then by Prop. 1.2, we have  $J_{\mu} \leq J^*$ , which implies that  $\mu$  is optimal. Conversely, if  $J_{\mu} = J^*$ , then  $T_{\mu}J_{\mu} = J_{\mu} = J^* = TJ^* = TJ_{\mu}$ . Q.E.D.

The interpretation of the preceding optimality condition is that persistently using  $\mu$  is optimal if and only if this performs at least as well as using any  $\bar{\mu}$  at the first stage and using  $\mu$  thereafter. Under Assumption P this condition is not sufficient to guarantee optimality of the stationary policy  $\mu$ , as the following example shows.

#### Example 1.4

Let  $S = (-\infty, \infty)$ ,  $U(x) = (0, 1]$  for all  $x \in S$ ,

$$g(x, u, w) = |x|, \quad f(x, u, w) = \alpha^{-1}ux,$$

for all  $(x, u, w) \in S \times C \times D$ . Let  $\mu(x) = 1$  for all  $x \in S$ . Then  $J_\mu(x) = \infty$  if  $x \neq 0$  and  $J_\mu(0) = 0$ . Furthermore, we have  $J_\mu = T_\mu J_\mu = TJ_\mu$ , as the reader can easily verify. It can also be verified that  $J^*(x) = |x|$ , and hence the stationary policy  $\mu$  is not optimal.

#### The Value Iteration Method

We now turn to the question whether the DP algorithm converges to the optimal cost function  $J^*$ . Let  $J_0$  be the zero function on  $S$ ,

$$J_0(x) = 0, \quad x \in S.$$

Then under Assumption P, we have

$$J_0 \leq TJ_0 \leq T^2J_0 \leq \dots \leq T^k J_0 \leq \dots,$$

while under Assumption N, we have

$$J_0 \geq TJ_0 \geq T^2J_0 \geq \dots \geq T^k J_0 \geq \dots$$

In either case the limit function

$$J_\infty(x) = \lim_{k \rightarrow \infty} (T^k J_0)(x), \quad x \in S, \quad (1.12)$$

is well defined, provided we allow the possibility that  $J_\infty$  can take the value  $\infty$  (under Assumption P) or  $-\infty$  (under Assumption N). The question is whether the value iteration method is valid in the sense

$$J_\infty = J^*. \quad (1.13)$$

This question is, of course, of computational interest, but it is also of analytical interest since, if one knows that  $J^* = \lim_{k \rightarrow \infty} T^k J_0$ , one can infer properties of the unknown function  $J^*$  from properties of the  $k$ -stage

optimal cost functions  $T^k J_0$ , which are defined in a concrete algorithmic manner.

We will show that  $J_\infty = J^*$  under Assumption N. It turns out, however, that under Assumption P, we may have  $J_\infty \neq J^*$  (see Exercise 3.1). We will later provide easily verifiable conditions that guarantee that  $J_\infty = J^*$  under Assumption P. We have the following proposition.

#### Proposition 1.5:

(a) Let Assumption P hold and assume that

$$J_\infty(x) = (TJ_\infty)(x), \quad x \in S.$$

Then if  $J : S \mapsto \mathbb{R}$  is any bounded function and  $\alpha < 1$ , or otherwise if  $J_0 \leq J \leq J^*$ , we have

$$\lim_{k \rightarrow \infty} (T^k J)(x) = J^*(x), \quad x \in S. \quad (1.14)$$

(b) Let Assumption N hold. Then if  $J : S \mapsto \mathbb{R}$  is any bounded function and  $\alpha < 1$ , or otherwise if  $J^* \leq J \leq J_0$ , we have

$$\lim_{k \rightarrow \infty} (T^k J)(x) = J^*(x), \quad x \in S. \quad (1.15)$$

**Proof:** (a) Since under Assumption P, we have

$$J_0 \leq TJ_0 \leq \dots \leq T^k J_0 \leq \dots \leq J^*,$$

it follows that  $\lim_{k \rightarrow \infty} T^k J_0 = J_\infty \leq J^*$ . Since  $J_\infty$  is also a fixed point of  $T$  by assumption, we obtain from Prop. 1.2(a) that  $J^* \leq J_\infty$ . It follows that

$$J_\infty = J^*, \quad (1.16)$$

and hence Eq. (1.14) is proved for the case  $J = J_0$ .

For the case where  $\alpha < 1$  and  $J$  is bounded, let  $r$  be a scalar such that

$$J_0 - rc \leq J \leq J_0 + rc. \quad (1.17)$$

Applying  $T^k$  to this relation, we obtain

$$T^k J_0 - \alpha^k rc \leq T^k J \leq T^k J_0 + \alpha^k rc. \quad (1.18)$$

Since  $T^k J_0$  converges to  $J^*$ , as shown earlier, this relation implies that  $T^k J$  converges also to  $J^*$ .

In the case where  $J_0 \leq J \leq J^*$ , we have by applying  $T^k$

$$T^k J_0 \leq T^k J \leq J^*, \quad k = 0, 1, \dots \quad (1.19)$$

Since  $T^k J_0$  converges to  $J^*$ , so does  $T^k J$ .

(b) It was shown earlier [cf. Eq. (1.6)] that under Assumption N, we have

$$J_\infty(x) = \lim_{k \rightarrow \infty} (T^k J_0)(x) = J^*(x). \quad (1.20)$$

The proof from this point is identical to that for part (a). Q.E.D.

We now derive conditions guaranteeing that  $J_\infty = TJ_\infty$  holds under Assumption P, which by Prop. 1.5 implies that  $J_\infty = J^*$ . We prove two propositions. The first admits an easy proof but requires a finiteness assumption on the control constraint set. The second is harder to prove but requires a weaker compactness assumption.

**Proposition 1.6:** Let Assumption P hold and assume that the control constraint set is finite for every  $x \in S$ . Then

$$J_\infty = TJ_\infty = J^*. \quad (1.21)$$

**Proof:** As shown in the proof of Prop. 1.5(a), we have for all  $k$ ,  $T^k J_0 \leq J_\infty \leq J^*$ . Applying  $T$  to this relation, we obtain

$$\begin{aligned} (T^{k+1} J_0)(x) &= \min_{u \in U(x)} E_w \{g(x, u, w) + \alpha(T^k J_0)(f(x, u, w))\} \\ &\leq (TJ_\infty)(x), \end{aligned} \quad (1.22)$$

and by taking the limit as  $k \rightarrow \infty$ , it follows that

$$J_\infty \leq TJ_\infty.$$

Suppose that there existed a state  $\tilde{x} \in S$  such that

$$J_\infty(\tilde{x}) < (TJ_\infty)(\tilde{x}). \quad (1.23)$$

Let  $u_k$  minimize in Eq. (1.22) when  $x = \tilde{x}$ . Since  $U(\tilde{x})$  is finite, there must exist some  $\hat{u} \in U(\tilde{x})$  such that  $u_k = \hat{u}$  for all  $k$  in some infinite subset  $K$  of the positive integers. By Eq. (1.22) we have for all  $k \in K$

$$\begin{aligned} (T^{k+1} J_0)(\tilde{x}) &= E_w \{g(\tilde{x}, \hat{u}, w) + \alpha(T^k J_0)(f(\tilde{x}, \hat{u}, w))\} \\ &\leq (TJ_\infty)(\tilde{x}). \end{aligned}$$

Taking the limit as  $k \rightarrow \infty$ ,  $k \in K$ , we obtain

$$\begin{aligned} J_\infty(\tilde{x}) &= E_w \{g(\tilde{x}, \hat{u}, w) + \alpha J_\infty(f(\tilde{x}, \hat{u}, w))\} \\ &\geq (TJ_\infty)(\tilde{x}) \\ &= \min_{u \in U(\tilde{x})} E_w \{g(\tilde{x}, u, w) + \alpha J_\infty(f(\tilde{x}, u, w))\}. \end{aligned}$$

This contradicts Eq. (1.23), so we have  $J_\infty(\tilde{x}) = (TJ_\infty)(\tilde{x})$ . Q.E.D.

The following proposition strengthens Prop. 1.6 in that it requires a compactness rather than a finiteness assumption. We recall (see Appendix A of Vol. I) that a subset  $X$  of the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$  is said to be *compact* if every sequence  $\{x_k\}$  with  $x_k \in X$  contains a subsequence  $\{x_{k_j}\}_{j \in K}$  that converges to a point  $x \in X$ . Equivalently,  $X$  is compact if and only if it is closed and bounded. The empty set is (trivially) considered compact. Given any collection of compact sets, their intersection is a compact set (possibly empty). Given a sequence of nonempty compact sets  $X_1, X_2, \dots, X_k, \dots$  such that

$$X_1 \supset X_2 \supset \dots \supset X_k \supset X_{k+1} \supset \dots \quad (1.24)$$

their intersection  $\cap_{k=1}^\infty X_k$  is both nonempty and compact. In view of this fact, it follows that if  $f : \mathbb{R}^n \mapsto [-\infty, \infty]$  is a function such that the set

$$F_\lambda = \{x \in \mathbb{R}^n \mid f(x) \leq \lambda\} \quad (1.25)$$

is compact for every  $\lambda \in R$ , then there exists a vector  $x^*$  minimizing  $f$ ; that is, there exists an  $x^* \in \mathbb{R}^n$  such that

$$f(x^*) = \min_{x \in \mathbb{R}^n} f(x). \quad (1.26)$$

To see this, take a sequence  $\{\lambda_k\}$  such that  $\lambda_k \rightarrow \min_{x \in \mathbb{R}^n} f(x)$  and  $\lambda_k \geq \lambda_{k+1}$  for all  $k$ . If  $\min_{x \in \mathbb{R}^n} f(x) < \infty$ , such a sequence exists and the sets

$$F_{\lambda_k} = \{x \in \mathbb{R}^n \mid f(x) \leq \lambda_k\} \quad (1.27)$$

are nonempty and compact. Furthermore,  $F_{\lambda_k} \supset F_{\lambda_{k+1}}$  for all  $k$ , and hence the intersection  $\cap_{k=1}^\infty F_{\lambda_k}$  is also nonempty and compact. Let  $x^*$  be any vector in  $\cap_{k=1}^\infty F_{\lambda_k}$ . Then

$$f(x^*) \leq \lambda_k, \quad k = 1, 2, \dots, \quad (1.28)$$

and taking the limit as  $k \rightarrow \infty$ , we obtain  $f(x^*) \leq \min_{x \in \mathbb{R}^n} f(x)$ , proving that  $x^*$  minimizes  $f(x)$ . The most common case where we can guarantee

that the set  $P_\lambda$  of Eq. (1.25) is compact for all  $\lambda$  if  $f$  is continuous and  $f(x) \rightarrow \infty$  as  $\|x\| \rightarrow \infty$ .

**Proposition 1.7:** Let Assumption P hold, and assume that the sets

$$U_k(x, \lambda) = \left\{ u \in U(x) \mid E_w \{ g(x, u, w) + \alpha(T^k J_0)(f(x, u, w)) \} \leq \lambda \right\} \quad (1.29)$$

are compact subsets of a Euclidean space for every  $x \in S$ ,  $\lambda \in \mathfrak{N}$ , and for all  $k$  greater than some integer  $\bar{k}$ . Then

$$J_\infty = TJ_\infty = J^*. \quad (1.30)$$

Furthermore, there exists a stationary optimal policy.

**Proof:** As in Prop. 1.6, we have  $J_\infty \leq TJ_\infty$ . Suppose that there existed a state  $\tilde{x} \in S$  such that

$$J_\infty(\tilde{x}) < (TJ_\infty)(\tilde{x}). \quad (1.31)$$

Clearly, we must have  $J_\infty(\tilde{x}) < \infty$ . For every  $k \geq \bar{k}$ , consider the sets

$$\begin{aligned} & U_k(\tilde{x}, J_\infty(\tilde{x})) \\ &= \left\{ u \in U(\tilde{x}) \mid E_w \{ g(\tilde{x}, u, w) + \alpha(T^k J_0)(f(\tilde{x}, u, w)) \} \leq J_\infty(\tilde{x}) \right\}. \end{aligned}$$

Let also  $u_k$  be a point attaining the minimum in

$$(T^{k+1} J_0)(\tilde{x}) = \min_{u \in U(\tilde{x})} E_w \{ g(\tilde{x}, u, w) + \alpha(T^k J_0)(f(\tilde{x}, u, w)) \};$$

that is,  $u_k$  is such that

$$(T^{k+1} J_0)(\tilde{x}) = E_w \{ g(\tilde{x}, u_k, w) + \alpha(T^k J_0)(f(\tilde{x}, u_k, w)) \}.$$

Such minimizing points  $u_k$  exist by our compactness assumption. For every  $k \geq \bar{k}$ , consider the sequence  $\{u_i\}_{i=k}^\infty$ . Since  $T^k J_0 \leq T^{k+1} J_0 \leq \dots \leq J_\infty$ , it follows that

$$\begin{aligned} & E_w \{ g(\tilde{x}, u_i, w) + \alpha(T^k J_0)(f(\tilde{x}, u_i, w)) \} \\ & \leq E_w \{ g(\tilde{x}, u_i, w) + \alpha(T^i J_0)(f(\tilde{x}, u_i, w)) \} \\ & \leq J_\infty(\tilde{x}), \quad i \geq k. \end{aligned}$$

Therefore  $\{u_i\}_{i=k}^\infty \subset U_k(\tilde{x}, J_\infty(\tilde{x}))$ , and since  $U_k(\tilde{x}, J_\infty(\tilde{x}))$  is compact, all the limit points of  $\{u_i\}_{i=k}^\infty$  belong to  $U_k(\tilde{x}, J_\infty(\tilde{x}))$  and at least one such limit point exists. Hence the same is true of the limit points of the whole sequence  $\{u_i\}_{i=k}^\infty$ . It follows that if  $\hat{u}$  is a limit point of  $\{u_i\}_{i=k}^\infty$  then

$$\hat{u} \in \bigcap_{k=\bar{k}}^\infty U_k(\tilde{x}, J_\infty(\tilde{x})).$$

This implies by Eq. (1.29) that for all  $k \geq \bar{k}$

$$J_\infty(\tilde{x}) \geq E_w \{ g(\tilde{x}, \hat{u}, w) + \alpha(T^k J_0)(f(\tilde{x}, \hat{u}, w)) \} \geq (T^{k+1} J_0)(\tilde{x}).$$

Taking the limit as  $k \rightarrow \infty$ , we obtain

$$J_\infty(\tilde{x}) = E_w \{ g(\tilde{x}, \hat{u}, w) + \alpha J_\infty(f(\tilde{x}, \hat{u}, w)) \}.$$

Since the right-hand side is greater than or equal to  $(TJ_\infty)(\tilde{x})$ , Eq. (1.31) is contradicted. Hence  $J_\infty = TJ_\infty$ , and Eq. (1.30) is proved in view of Prop. 1.5(a).

To show that there exists an optimal stationary policy, observe that Eq. (1.30) and the last relation imply that  $\hat{u}$  attains the minimum in

$$J^*(\tilde{x}) = \min_{u \in U(\tilde{x})} E_w \{ g(\tilde{x}, u, w) + \alpha J^*(f(\tilde{x}, u, w)) \}$$

for a state  $\tilde{x} \in S$  with  $J^*(\tilde{x}) < \infty$ . For states  $\tilde{x} \in S$  such that  $J^*(\tilde{x}) = \infty$ , every  $u \in U(\tilde{x})$  attains the preceding minimum. Hence by Prop. 1.3(a) an optimal stationary policy exists. **Q.E.D.**

The reader may verify by inspection of the preceding proof that if  $\mu_k(\tilde{x})$ ,  $k = 0, 1, \dots$ , attains the minimum in the relation

$$(T^{k+1} J_0)(\tilde{x}) = \min_{u \in U(\tilde{x})} E_w \{ g(\tilde{x}, u, w) + \alpha(T^k J_0)(f(\tilde{x}, u, w)) \},$$

then if  $\mu^*(\tilde{x})$  is a limit point of  $\{\mu_k(\tilde{x})\}$ , for every  $\tilde{x} \in S$ , the stationary policy  $\mu^*$  is optimal. Furthermore,  $\{\mu_k(\tilde{x})\}$  has at least one limit point for every  $\tilde{x} \in S$  for which  $J^*(\tilde{x}) < \infty$ . Thus *the value iteration method under the assumptions of either Prop. 1.6 or Prop. 1.7 yields in the limit not only the optimal cost function  $J^*$  but also an optimal stationary policy*.

### Other Computational Methods

Unfortunately, policy iteration is not a valid procedure under either P or N in the absence of further conditions. If  $\mu$  and  $\bar{\mu}$  are stationary policies such that  $T_{\bar{\mu}} J_\mu = TJ_\mu$ , then it can be shown that under Assumption P we have

$$J_{\bar{\mu}}(x) \leq J_\mu(x), \quad x \in S. \quad (1.32)$$

To see this, note that  $T_{\bar{\mu}} J_{\mu} = T J_{\mu} \leq T_{\mu} J_{\mu} = J_{\mu}$  from which we obtain  $\lim_{N \rightarrow \infty} T_{\bar{\mu}}^N J_{\mu} \leq J_{\mu}$ . Since  $J_{\bar{\mu}} = \lim_{N \rightarrow \infty} T_{\bar{\mu}}^N J_0$  and  $J_0 \leq J_{\mu}$ , we obtain  $J_{\bar{\mu}} \leq J_{\mu}$ . However,  $J_{\mu} \leq J_{\bar{\mu}}$  by itself is not sufficient to guarantee the validity of policy iteration. For example, it is not clear that strict inequality holds in Eq. (1.32) for at least one state  $x \in S$  when  $\mu$  is not optimal. The difficulty here is that the equality  $J_{\mu} = T J_{\mu}$  does not imply that  $\mu$  is optimal, and additional conditions are needed to guarantee the validity of policy iteration. However, for special cases such conditions can be verified (see for example Section 3.2 and Exercise 3.16).

It is possible to devise a computational method based on mathematical programming when  $S$ ,  $C$ , and  $D$  are finite sets by making use of Prop. 1.2. Under  $N$  and  $\alpha = 1$ , the corresponding (linear) program is (compare with Section 1.3.4)

$$\begin{aligned} & \text{maximize } \sum_{i=1}^n \lambda_i \\ & \text{subject to } \lambda_i \leq g(i, u) + \sum_{j=1}^n p_{ij}(u) \lambda_j, \quad i = 1, 2, \dots, n, \quad u \in U(i). \end{aligned}$$

When  $\alpha = 1$  and Assumption P holds, the corresponding program takes the form

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n \lambda_i \\ & \text{subject to } \lambda_i \geq \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) \lambda_j \right], \quad i = 1, \dots, n, \end{aligned}$$

but unfortunately this program is not linear or even convex.

### 3.2 LINEAR SYSTEMS AND QUADRATIC COST

Consider the case of the linear system

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, 1, \dots,$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$  for all  $k$ , and the matrices  $A$ ,  $B$  are known. As in Sections 4.1 and 5.2 of Vol. I, we assume that the random disturbances

$w_k$  are independent with zero mean and finite second moments. The cost function is quadratic and has the form

$$J_{\mu}(x_0) = \lim_{N \rightarrow \infty} E_{\mu} \left\{ \sum_{k=0}^{N-1} \alpha^k (x'_k Q x_k + \mu_k(x_k)' R \mu_k(x_k)) \right\},$$

where  $Q$  is a positive semidefinite symmetric  $n \times n$  matrix and  $R$  is a positive definite symmetric  $m \times m$  matrix. Clearly, Assumption P of Section 3.1 holds.

Our approach will be to use the DP algorithm to obtain the functions  $TJ_0, T^2J_0, \dots$ , as well as the pointwise limit function  $J_{\infty} = \lim_{k \rightarrow \infty} T^k J_0$ . Subsequently, we show that  $J_{\infty}$  satisfies  $J_{\infty} = TJ_{\infty}$  and hence, by Prop. 1.5(a) of Section 3.1,  $J_{\infty} = J^*$ . The optimal policy is then obtained from the optimal cost function  $J^*$  by minimizing in Bellman's equation (cf. Prop. 1.3 of Section 3.1).

As in Section 4.1 of Vol. I, we have

$$\begin{aligned} J_0(x) &= 0, \quad x \in \mathbb{R}^n, \\ (TJ_0)(x) &= \min_u [x' Q x + u' R u] = x' Q x, \quad x \in \mathbb{R}^n, \\ (T^2J_0)(x) &= \min_u E \{ x' Q x + u' R u + \alpha(Ax + Bu + w)' Q(Ax + Bu + w) \} \\ &= x' K_1 x + \alpha E \{ w' Q w \}, \quad x \in \mathbb{R}^n, \\ (T^{k+1}J_0)(x) &= x' K_k x + \sum_{m=0}^{k-1} \alpha^{k-m} E \{ w' K_m w \}, \quad x \in \mathbb{R}^n, \quad k = 1, 2, \dots, \end{aligned}$$

where the matrices  $K_0, K_1, K_2, \dots$  are given recursively by

$$K_0 = Q,$$

$$K_{k+1} = A' (\alpha K_k - \alpha^2 K_k B (\alpha B' K_k B + R)^{-1} B' K_k) A + Q, \quad k = 0, 1, \dots$$

By defining  $\tilde{R} = R/\alpha$  and  $\tilde{A} = \sqrt{\alpha}A$ , the preceding equation may be written as

$$K_{k+1} = \tilde{A}' (K_k - K_k B (B' K_k B + \tilde{R})^{-1} B' K_k) \tilde{A} + Q,$$

and is of the form considered in Section 4.1 of Vol. I. By using the result shown there, we have that the generated matrix sequence  $\{K_k\}$  converges to a positive definite symmetric matrix  $K$ ,

$$K_k \rightarrow K,$$

provided the pairs  $(\tilde{A}, B)$  and  $(\tilde{A}, C)$ , where  $Q = C' C$ , are controllable and observable, respectively. Since  $\tilde{A} = \sqrt{\alpha}A$ , controllability and observability

of  $(A, B)$  or  $(A, C)$  are clearly equivalent to controllability and observability of  $(\bar{A}, B)$  or  $(\bar{A}, C)$ , respectively. The matrix  $K$  is the unique solution of the equation

$$K = A'(\alpha K - \alpha^2 KB(\alpha B'KB + R)^{-1}B'K)A + Q. \quad (2.1)$$

Because  $K_k \leftarrow K$ , it can also be seen that the limit

$$c = \lim_{k \rightarrow \infty} \sum_{m=0}^{k-1} \alpha^{k-m} E\{w' K_m w\}$$

is well defined, and in fact

$$c = \frac{\alpha}{1-\alpha} E\{w' K w\}. \quad (2.2)$$

Thus, in conclusion, if the pairs  $(A, B)$  and  $(A, C)$  are controllable and observable, respectively, the limit of the functions  $T^k J_0$  is given by

$$J_\infty(x) = \lim_{k \rightarrow \infty} (T^k J_0)(x) = x' K x + c. \quad (2.3)$$

Using Eqs. (2.1) to (2.3), it can be verified by straightforward calculation that for all  $x \in S$

$$J_\infty(x) = (TJ_\infty)(x) = \min_u [x' Qx + u' Ru + \alpha E\{J_\infty(Ax + Bu + w)\}] \quad (2.4)$$

and hence, by Prop. 1.5(a) of Section 3.1,  $J_\infty = J^*$ . Another method for proving that  $J_\infty = TJ_\infty$  is to show that the assumption of Prop. 1.7 of Section 3.1, is satisfied; that is, the sets

$$U_k(x, \lambda) = \{u \mid E\{x' Qx + u' Ru + \alpha(T^k J_0)(Ax + Bu + w)\} \leq \lambda\}$$

are compact for all  $k$  and scalars  $\lambda$ . This can be verified using the fact that  $T^k J_0$  is a positive semidefinite quadratic function and  $R$  is positive definite. The optimal stationary policy  $p^*$ , obtained by minimization in Eq. (2.4), has the form

$$p^*(x) = -\alpha(\alpha B'KB + R)^{-1}B'KAx, \quad x \in \mathbb{R}^n.$$

This policy is attractive for practical implementation since it is linear and stationary. A number of generalized versions of the problem of this section, including the case of imperfect state information, are treated in the exercises. Interestingly, the problem can be solved by policy iteration (see Exercise 3.16), even though, as discussed in Section 3.1, policy iteration is not valid in general under Assumption P.

### 3.3 INVENTORY CONTROL

Let us consider a discounted, infinite horizon version of the inventory control problem of Section 4.2 in Vol. I. Inventory stock evolves according to the equation

$$x_{k+1} = x_k + u_k - w_k, \quad k = 0, 1, \dots \quad (3.1)$$

We assume that the successive demands  $w_k$  are independent and bounded, and have identical probability distributions. We also assume for simplicity that there is no fixed cost. The case of a nonzero fixed cost can be treated similarly. The cost function is

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \min_{\substack{u_k \\ k=0,1,\dots,N-1}} \left\{ \sum_{k=0}^{N-1} \alpha^k (\mu_k(x_k) + H(x_k + \mu(x_k) - w_k)) \right\},$$

where

$$H(y) = p \max(0, -y) + h \max(0, y).$$

The DP algorithm is given by

$$J_0(x) = 0,$$

$$(T^{k+1} J_0)(x) = \min_u E\{cu + H(x + u - w) + \alpha(T^k J_0)(x + u - w)\}. \quad (3.2)$$

We first show that the optimal cost is finite for all initial states, that is,

$$J^*(x_0) = \min_\pi J_\pi(x_0) < \infty, \quad \text{for all } x_0 \in S. \quad (3.3)$$

Indeed, consider the policy  $\tilde{\pi} = \{\tilde{\mu}_0, \tilde{\mu}_1, \dots\}$ , where  $\tilde{\mu}$  is defined by

$$\tilde{\mu}(x) = \begin{cases} 0 & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

Since  $w_k$  is nonnegative and bounded, it follows that the inventory stock  $x_k$  when the policy  $\tilde{\pi}$  is used satisfies

$$-w_{k-1} \leq x_k \leq \max(0, x_0), \quad k = 1, 2, \dots,$$

and is bounded. Hence  $\tilde{\mu}(x_k)$  is also bounded. It follows that the cost per stage incurred when  $\tilde{\pi}$  is used is bounded, and in view of the presence of the discount factor we have

$$J_{\tilde{\pi}}(x_0) < \infty, \quad x_0 \in S.$$

Since  $J^* \leq J_{\tilde{\pi}}$ , the finiteness of the optimal cost follows.

Next we observe that, under the assumption  $c < p$ , the functions  $T^k J_0$  are real-valued and convex. Indeed, we have

$$J_0 \leq T J_0 \leq \cdots \leq T^k J_0 \leq \cdots \leq J^*,$$

which implies that  $T^k J_0$  is real-valued. Convexity follows by induction as shown in Section 4.2 of Vol. I.

Consider now the sets

$$U_k(x, \lambda) = \{u \geq 0 \mid E\{cu + H(x+u-w) + \alpha(T^k J_0)(x_u - w)\} \leq \lambda\}. \quad (3.4)$$

These sets are bounded since the expected value within the braces above tends to  $\infty$  as  $u \rightarrow \infty$ . Also, the sets  $U_k(x, \lambda)$  are closed since the expected value in Eq. (3.4) is a continuous function of  $u$  [recall that  $T^k J_0$  is a real-valued convex and hence continuous function]. Thus we may invoke Prop. 1.7 of Section 3.1 and assert that

$$\lim_{k \rightarrow \infty} (T^k J_0)(x) = J^*(x), \quad x \in S.$$

It follows from the convexity of the functions  $T^k J_0$  that the limit function  $J^*$  is a real-valued convex function. Furthermore, an optimal stationary policy  $\mu^*$  can be obtained by minimizing in the right-hand side of Bellman's equation

$$J^*(x) = \min_{u \geq 0} E\{cu + H(x+u-w) + \alpha J^*(x+u-w)\}.$$

We have

$$\mu^*(x) = \begin{cases} S^* - x & \text{if } x \leq S^*, \\ 0 & \text{otherwise,} \end{cases}$$

where  $S^*$  is a minimizing point of

$$G^*(y) = cy + L(y) + \alpha E\{J^*(y-w)\},$$

with

$$L(y) = E\{H(y-w)\}.$$

It can be seen that if  $p > c$ , we have  $\lim_{|y| \rightarrow \infty} G^*(y) = \infty$ , so that such a minimizing point exists. Furthermore, by using the observation made near the end of Section 3.1, it follows that a minimizing point  $S^*$  of  $G^*(y)$  may be obtained as a limit point of a sequence  $\{S_k\}$ , where for each  $k$  the scalar  $S_k$  minimizes

$$G_k(y) = cy + L(y) + \alpha E\{(T^k J_0)(y-w)\}$$

and is obtained by means of the value iteration method.

It turns out that the critical level  $S^*$  has a simple characterization. It can be shown that  $S^*$  minimizes over  $y$  the expression  $(1-\alpha)cy + L(y)$ , and it can be essentially obtained in closed form (see Exercise 3.18, and [HeS84], Ch. 2).

In the case where there is a positive fixed cost ( $K > 0$ ), the same line of argument may be used. Similarly, we prove that  $J^*$  is a real-valued  $K$ -convex function. A separate argument is necessary to prove that  $J^*$  is also continuous (this is intuitively clear and is left for the reader). Once  $K$ -convexity and continuity of  $J^*$  are established, the optimality of a stationary  $(s^*, S^*)$  policy follows from the equation

$$J^*(x) = \min_{u \geq 0} E\{C(u) + H(x+u-w) + \alpha J^*(x+u-w)\},$$

where  $C(u) = K + cu$  if  $u > 0$  and  $C(0) = 0$ .

### 3.4 OPTIMAL STOPPING

Consider an infinite horizon version of the stopping problems of Section 4.4 of Vol. I. At each state  $x$ , we must choose between two actions: pay a stopping cost  $s(x)$  and *stop*, or pay a cost  $c(x)$  and *continue* the process according to the system equation

$$x_{k+1} = f_c(x_k, w_k), \quad k = 0, 1, \dots \quad (4.1)$$

The objective is to find the optimal stopping policy that minimizes the total expected cost over an infinite number of stages. It is assumed that the input disturbances  $w_k$  have the same probability distribution for all  $k$ , which depends only on the current state  $x_k$ .

This problem may be viewed as a special case of the stochastic shortest path problem of Section 2.1, but here we will not assume that the state space is finite and that only proper policies can be optimal, as we did in Section 2.1. Instead we will rely on the general theory of unbounded cost problems developed in Section 3.1.

To put the problem within the framework of the total cost infinite horizon problem, we introduce an additional state  $t$  (termination state) and we complete the system equation (4.1) as in Section 4.4 of Vol. I by letting

$$x_{k+1} = t, \quad \text{if } u_k = \text{stop or } x_k = t.$$

Once the system reaches the termination state, it remains there permanently at no cost.

We first assume that

$$s(x) \geq 0, \quad c(x) \geq 0, \quad \text{for all } x \in S, \quad (4.2)$$

thus coming under the framework of Assumption P of Section 3.1. The case corresponding to Assumption N, where  $s(x) \leq 0$  and  $c(x) \leq 0$  for all  $x \in S$  will be considered later. Actually, whenever there exists an  $\epsilon > 0$  such that  $c(x) \geq \epsilon$  for all  $x \in S$ , the results to be obtained under the assumption (4.2) apply also to the case where  $s(x)$  is bounded below by some scalar rather than bounded by zero. The reason is that, if  $c(x)$  is assumed to be greater than  $\epsilon > 0$  for all  $x \in S$ , any policy that will not stop within a finite expected number of stages results in infinite cost and can be excluded from consideration. As a result, if we reformulate the problem and add a constant  $r$  to  $s(x)$  so that  $s(x) + r \geq 0$  for all  $x \in S$ , the optimal cost  $J^*(x)$  will merely be increased by  $r$ , while optimal policies will remain unaffected.

The mapping  $T$  that defines the DP algorithm takes the form

$$(TJ)(x) = \begin{cases} \min[s(x), c(x) + E\{J(f_c(x, w))\}] & \text{if } x \neq t, \\ 0 & \text{if } x = t, \end{cases} \quad (4.3)$$

where  $s(x)$  is the cost of the stopping action, and  $c(x) + E\{J(f_c(x, w))\}$  is the cost of the continuation action. Since the control space has only two elements, by Prop. 4.6 of Section 3.1, we have

$$\lim_{k \rightarrow \infty} (T^k J_0)(x) = J^*(x), \quad x \in S, \quad (4.4)$$

where  $J_0$  is the zero function [ $J_0(x) = 0$ , for all  $x \in S$ ]. By Prop. 4.3 of Section 3.1, there exists a stationary optimal policy given by

$$\begin{aligned} \text{stop} & \quad \text{if } s(x) < c(x) + E\{J^*(f_c(x, w))\}, \\ \text{continue} & \quad \text{if } s(x) \geq c(x) + E\{J^*(f_c(x, w))\}. \end{aligned}$$

Let us denote by  $S^*$  the optimal stopping set (which may be empty)

$$S^* = \{x \in S \mid s(x) < c(x) + E\{J^*(f_c(x, w))\}\}.$$

Consider also the sets

$$S_k = \{x \in S \mid s(x) < c(x) + E\{(T^k J_0)(f_c(x, w))\}\}$$

that determine the optimal policy for finite horizon versions of the stopping problem. Since we have

$$J_0 \leq TJ_0 \leq \dots \leq T^k J_0 \leq \dots \leq J^*,$$

it follows that

$$S_1 \subset S_2 \subset \dots \subset S_k \subset \dots \subset S^*$$

and therefore  $\cup_{k=1}^{\infty} S_k \subset S^*$ . Also, if  $\tilde{x} \notin \cup_{k=1}^{\infty} S_k$ , then we have

$$s(\tilde{x}) \geq c(\tilde{x}) + E\{(T^k J_0)(f_c(\tilde{x}, w))\}, \quad k = 0, 1, \dots,$$

and by taking the limit and using the monotone convergence theorem and the fact  $T^k J_0 \rightarrow J^*$ , we obtain

$$s(\tilde{x}) \geq c(\tilde{x}) + E\{J^*(f_c(\tilde{x}, w))\},$$

from which  $\tilde{x} \notin S^*$ . Hence

$$S^* = \cup_{k=1}^{\infty} S_k. \quad (4.5)$$

In other words, the *optimal stopping set*  $S^*$  for the infinite horizon problem is equal to the union of all the finite horizon stopping sets  $S_k$ .

Consider now, as in Section 4.4 of Vol. 1, the one-step-to-go stopping set

$$\tilde{S}_1 = \{x \in S \mid s(x) \leq c(x) + E\{t(f_c(x, w))\}\} \quad (4.6)$$

and assume that  $\tilde{S}_1$  is *absorbing* in the sense

$$f_c(x, w) \in \tilde{S}_1, \quad \text{for all } x \in \tilde{S}_1, \quad w \in D. \quad (4.7)$$

Then, as in Section 4.4 of Vol. 1, it follows that the one-step lookahead policy

stop if and only if  $x \in \tilde{S}_1$

is optimal. We now provide some examples.

#### Example 4.1 (Asset Selling)

Consider the version of the asset selling example of Sections 4.4 and 7.3 of Vol. 1, where the rate of interest  $r$  is zero and there is instead a maintenance cost  $c > 0$  per period for which the house remains unsold. Furthermore, past offers can be accepted at any future time. We have the following optimality equation:

$$J^*(x) = \max[x, -c + E\{J^*(\max(x, w))\}].$$

In this case we consider maximization of total expected reward, the continuation cost is strictly negative, and the stopping reward  $x$  is positive. Hence the assumption (4.2) is not satisfied. If, however, we assume that  $x$  takes values in a bounded interval  $[0, M]$ , where  $M$  is an upper bound on the possible values of offers, our analysis is still applicable [cf. the discussion following Eq. (4.2)]. Consider the one-step-to-go stopping set given by

$$\tilde{S}_1 = \{x \mid x \geq -c + E\{\max(x, w)\}\}.$$

After a calculation similar to the one given in Section 4.4 of Vol. 1, we see that

$$\tilde{S}_1 = \{x \mid x \geq \bar{a}\},$$

where  $\bar{a}$  is the scalar satisfying

$$\bar{a} = P(\bar{a})\bar{a} + \int_{\bar{a}}^{\infty} w dP(w) - c.$$

Clearly,  $\tilde{S}_1$  is absorbing in the sense of Eq. (4.7) and therefore the one-step lookahead policy that accepts the first offer greater than or equal to  $\bar{a}$  is optimal.

**Example 4.2 (Sequential Hypothesis Testing)**

Consider the hypothesis testing problem of Section 5.5 of Vol. I for the case where the number of possible observations is unlimited. Here the states are  $x^0$  and  $x^1$  (true distribution of the observations is  $f_0$  and  $f_1$ , respectively). The set  $S$  is the interval  $[0, 1]$  and corresponds to the sufficient statistic

$$p_k = P(r_k = x^0 \mid z_0, z_1, \dots, z_k).$$

To each  $p \in [0, 1]$  we may assign the stopping cost

$$s(p) = \min[(1-p)L_0, pL_1],$$

that is, the cost associated with optimal choice between the distributions  $f_0$  and  $f_1$ . The mapping  $T$  of Eq. (4.3) takes the form

$$(TJ)(p) = \min \left[ (1-p)L_0, pL_1, c + E_z \left\{ J \left( \frac{pf_0(z)}{pf_0(z) + (1-p)f_1(z)} \right) \right\} \right]$$

for all  $p \in [0, 1]$ , where the expectation over  $z$  is taken with respect to the probability distribution

$$P(z) = pf_0(z) + (1-p)f_1(z), \quad z \in Z.$$

The optimal cost function  $J^*$  satisfies Bellman's equation

$$J^*(p) = \min \left[ (1-p)L_0, pL_1, c + E_z \left\{ J^* \left( \frac{pf_0(z)}{pf_0(z) + (1-p)f_1(z)} \right) \right\} \right]$$

and is obtained in the limit through the equation

$$J^*(p) = \lim_{k \rightarrow \infty} (T^k J_0)(p), \quad p \in [0, 1],$$

where  $J_0$  is the zero function on  $[0, 1]$ .

Now consider the functions  $T^k J_0$ ,  $k = 0, 1, \dots$ . It is clear that

$$J_0 \leq TJ_0 \leq \dots \leq T^k J_0 \leq \dots \leq \min[(1-p)L_0, pL_1].$$

Furthermore, in view of the analysis of Section 5.5 of Vol. I, we have that the function  $T^k J_0$  is concave on  $[0, 1]$  for all  $k$ . Hence the pointwise limit function  $J^*$  is also concave on  $[0, 1]$ . In addition, Bellman's equation implies that

$$J^*(0) = J^*(1) = 0,$$

$$J^*(p) \leq \min[(1-p)L_0, pL_1].$$

Using the reasoning illustrated in Fig. 3.4.1 it follows that [provided  $c < L_0 L_1 / (L_0 + L_1)$ ] there exist two scalars  $\bar{\alpha}$ ,  $\bar{\beta}$  with  $0 < \bar{\beta} \leq \bar{\alpha} < 1$ , that determine an optimal stationary policy of the form

$$\begin{aligned} \text{accept } f_0 &\quad \text{if } p \leq \bar{\alpha}, \\ \text{accept } f_1 &\quad \text{if } p \leq \bar{\beta}, \\ \text{continue the observations} &\quad \text{if } \bar{\beta} < p < \bar{\alpha}. \end{aligned}$$

In view of the optimality of the preceding stationary policy, the sequential probability ratio test described in Section 5.5 of Vol. I is justified when the number of possible observations is infinite.

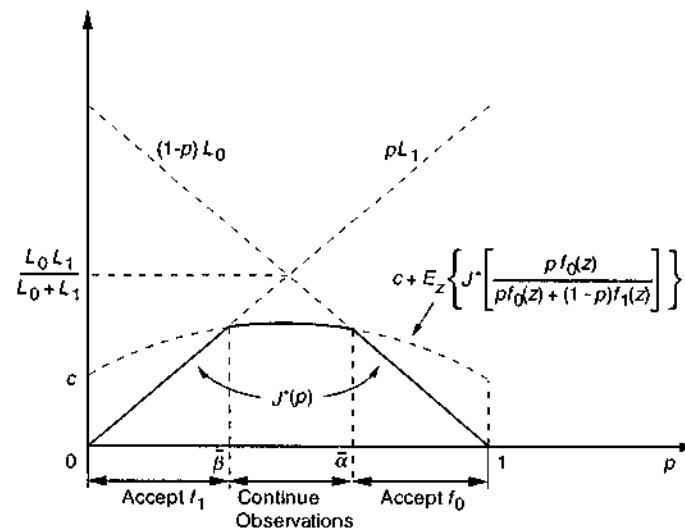


Figure 3.4.1 Derivation of the sequential probability ratio test.

**The Case of Negative Transition Costs**

We now consider the stopping problem under Assumption N, that is,

$$s(x) \leq 0, \quad c(x) \leq 0, \quad \text{for all } x \in S.$$

Under these circumstances there is no penalty for continuing operation of the system (although by not stopping at a given state, a favorable opportunity may be missed). The mapping  $T$  is given by

$$(TJ)(x) = \min[s(x), c(x) + E\{J(f_c(x, w))\}].$$

The optimal cost function  $J^*$  satisfies  $J^*(x) \leq s(x)$  for all  $x \in S$ , and by using Props. 1.1 and 1.5(b) of Section 3.1, we have

$$J^* = TJ^*, \quad J^* = \lim_{k \rightarrow \infty} T^k J_0 = \lim_{k \rightarrow \infty} T^k s,$$

where  $J_0$  is the zero function. It can also be seen that if the one-step-to-go stopping set  $\tilde{S}_1$  is *absorbing* [cf. Eq. (4.7)], a one-step lookahead policy is optimal.

### Example 4.3 (The Rational Burglar)

This example was considered at the end of Section 4.1 of Vol. I where it was shown that a one-step lookahead policy is optimal for any finite horizon length. The optimality equation is

$$J^*(x) = \max\{x, (1-p)E\{J^*(x+w)\}\}.$$

The problem is equivalent to a minimization problem where

$$s(x) = -x, \quad c(x) = 0,$$

so Assumption N holds. From the preceding analysis, we have that  $T^k s \rightarrow J^*$  and that a one-step lookahead policy is optimal if the one-step stopping set is absorbing [cf. Eqs. (4.6) and (4.7)]. It can be shown (see the analysis of Section 4.4 of Vol. I) that this condition holds, so the finite horizon optimal policy whereby the burglar retires when his accumulated earnings reach or exceed  $(1-p)\bar{w}/p$  is optimal for an infinite horizon as well.

### Example 4.4 (A Problem with no Optimal Policy)

This is a deterministic stopping problem where Assumption N holds, and an optimal policy does not exist, even though only two controls are available at each state (stop and continue). The states are the positive integers, and continuation from state  $i$  leads to state  $i+1$  with certainty and no cost, that is,  $S = \{1, 2, \dots\}$ ,  $c(t) = 0$ , and  $f_c(i, w) = i+1$  for all  $i \in S$  and  $w \in D$ . The stopping cost is  $s(i) = -1 + (1/i)$  for all  $i \in S$ , so that there is an incentive to delay stopping at every state. We have  $J^*(i) = -1$  for all  $i$ , and the optimal cost  $-1$  can be approached arbitrarily closely by postponing the stopping action for a sufficiently long time. However, there does not exist an optimal policy that attains the optimal cost.

## 3.5 OPTIMAL GAMBLING STRATEGIES

A gambler enters a certain game played as follows. The gambler may stake at any time  $k$  any amount  $u_k \geq 0$  that does not exceed his current fortune  $x_k$  (defined to be his initial capital plus his gain or minus his loss thus far). He wins his stake back and as much more with probability  $p$  and he loses his stake with probability  $(1-p)$ . Thus the gambler's fortune evolves according to the equation

$$x_{k+1} = x_k + w_k u_k, \quad k = 0, 1, \dots, \quad (5.1)$$

where  $w_k = 1$  with probability  $p$  and  $w_k = -1$  with probability  $(1-p)$ . Several games, such as playing red and black in roulette, fit this description.

The gambler enters the game with an initial capital  $x_0$ , and his goal is to increase his fortune up to a level  $X$ . He continues gambling until he either reaches his goal or loses his entire initial capital, at which point he leaves the game. The problem is to determine the optimal gambling strategy for maximizing the probability of reaching his goal. By a gambling strategy, we mean a rule that specifies what the stake should be at time  $k$  when the gambler's fortune is  $x_k$ , for every  $x_k$  with  $0 < x_k < X$ .

The problem may be cast within the total cost, infinite horizon framework, where we consider maximization in place of minimization. Let us assume for convenience that fortunes are normalized so that  $X = 1$ . The state space is the set  $[0, 1] \cup \{t\}$ , where  $t$  is a termination state to which the system moves with certainty from both states 0 and 1 with corresponding rewards 0 and 1. When  $x_k \neq 0, x_k \neq 1$ , the system evolves according to Eq. (5.1). The control constraint set is specified by

$$0 \leq u_k \leq x_k, \quad 0 \leq u_k \leq 1 - x_k.$$

The reward per stage when  $x_k \neq 0$  and  $x_k \neq 1$  is zero. Under these circumstances the probability of reaching the goal is equal to the total expected reward. Assumption N holds since our problem is equivalent to a problem of minimizing expected total cost with nonpositive costs per stage.

The mapping  $T$  defining the DP algorithm takes the form

$$(TJ)(x) = \begin{cases} \max_{\substack{0 \leq u \leq x \\ 0 \leq u \leq 1-x}} [pJ(x+u) + (1-p)J(x-u)] & \text{if } x \in (0, 1), \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x = 1, \end{cases} \quad (5.2)$$

for any function  $J : [0, 1] \mapsto [0, \infty]$ .

Consider now the case where

$$0 < p < \frac{1}{2},$$

that is, the game is unfair to the gambler. A discretized version of the case where  $1/2 \leq p < 1$  is considered in Exercise 3.21. When  $0 < p < 1/2$ , it is intuitively clear that if the gambler follows a very conservative strategy and stakes a very small amount at each time, he is all but certain to lose his capital. For example, if the gambler adopts a strategy of betting  $1/n$  at each time, then it may be shown (see Exercise 3.21 or [Asht70], p. 182) that his probability of attaining the target fortune of 1 starting with an initial capital  $i/n$ ,  $0 < i < n$ , is given by

$$\left( \left( \frac{1-p}{p} \right)^i - 1 \right) \left( \left( \frac{1-p}{p} \right)^n - 1 \right)^{-1}.$$

If  $0 < p < 1/2$ ,  $n$  tends to infinity, and  $i/n$  tends to a constant, the above probability tends to zero, thus indicating that placing consistently small bets is a bad strategy.

We are thus led to a policy that places large bets and, in particular, the *bold strategy* whereby the gambler stakes at each time  $k$  his entire fortune  $x_k$  or just enough to reach his goal, whichever is least. In other words, the bold strategy is the stationary policy  $\mu^*$  given by

$$\mu^*(x) = \begin{cases} x & \text{if } 0 < x \leq 1/2, \\ 1-x & \text{if } 1/2 \leq x < 1. \end{cases}$$

We will prove that the bold strategy is indeed an optimal policy. To this end it is sufficient to show that for every initial fortune  $x \in [0, 1]$  the value of the reward function  $J_{\mu^*}(x)$  corresponding to the bold strategy  $\mu^*$  satisfies the sufficiency condition (cf. Prop. 1.4, Section 3.1)

$$TJ_{\mu^*} = J_{\mu^*},$$

or equivalently

$$\begin{aligned} J_{\mu^*}(0) &= 0, & J_{\mu^*}(1) &= 1, \\ J_{\mu^*}(x) &\geq pJ_{\mu^*}(x+u) + (1-p)J_{\mu^*}(x-u), \end{aligned} \quad (5.3)$$

for all  $x \in (0, 1)$  and  $u \in [0, x] \cap [0, 1-x]$ .

By using the definition of the bold strategy, Bellman's equation

$$J_{\mu^*} = T_{\mu^*}J_{\mu^*},$$

is written as

$$J_{\mu^*}(0) = 0, \quad J_{\mu^*}(1) = 1, \quad (5.4)$$

$$J_{\mu^*}(x) = \begin{cases} pJ_{\mu^*}(2x) & \text{if } 0 < x \leq 1/2, \\ p + (1-p)J_{\mu^*}(2x-1) & \text{if } 1/2 \leq x < 1. \end{cases} \quad (5.5)$$

The following lemma shows that  $J_{\mu^*}$  is uniquely defined from these relations.

**Lemma 5.1:** For every  $p$ , with  $0 < p \leq 1/2$ , there is only one bounded function on  $[0, 1]$  satisfying Eqs. (5.4) and (5.5), the function  $J_{\mu^*}$ . Furthermore,  $J_{\mu^*}$  is continuous and strictly increasing on  $[0, 1]$ .

**Proof:** Suppose that there existed two bounded functions  $J_1 : [0, 1] \mapsto \mathbb{R}$  and  $J_2 : [0, 1] \mapsto \mathbb{R}$  such that  $J_i(0) = 0$ ,  $J_i(1) = 1$ ,  $i = 1, 2$ , and

$$J_i(x) = \begin{cases} pJ_i(2x) & \text{if } 0 < x \leq 1/2, \\ p + (1-p)J_i(2x-1) & \text{if } 1/2 \leq x < 1, \end{cases} \quad i = 1, 2.$$

Then we have

$$J_1(2x) - J_2(2x) = \frac{J_1(x) - J_2(x)}{p}, \quad \text{if } 0 \leq x \leq 1/2, \quad (5.6)$$

$$J_1(2x-1) - J_2(2x-1) = \frac{J_1(x) - J_2(x)}{1-p}, \quad \text{if } 1/2 \leq x < 1. \quad (5.7)$$

Let  $z$  be any real number with  $0 \leq z \leq 1$ . Define

$$z_1 = \begin{cases} 2z & \text{if } 0 \leq z \leq 1/2, \\ 2z-1 & \text{if } 1/2 < z \leq 1, \end{cases}$$

⋮

$$z_k = \begin{cases} 2z_{k-1} & \text{if } 0 \leq z_{k-1} \leq 1/2, \\ 2z_{k-1}-1 & \text{if } 1/2 < z_{k-1} \leq 1, \end{cases}$$

for  $k = 1, 2, \dots$ . Then from Eqs. (5.6) and (5.7) it follows (using  $p \leq 1/2$ ) that

$$|J_1(z_k) - J_2(z_k)| \geq \frac{|J_1(z) - J_2(z)|}{(1-p)^k}, \quad k = 1, 2, \dots$$

Since  $J_1(z_k) - J_2(z_k)$  is bounded, it follows that  $J_1(z) - J_2(z) = 0$ , for otherwise the right side of the inequality would tend to  $\infty$ . Since  $z \in [0, 1]$  is arbitrary, we obtain  $J_1 = J_2$ . Hence  $J_{\mu^*}$  is the unique bounded function on  $[0, 1]$  satisfying Eqs. (5.4) and (5.5).

To show that  $J_{\mu^*}$  is strictly increasing and continuous, we consider the mapping  $T_{\mu^*}$ , which operates on functions  $J : [0, 1] \mapsto [0, 1]$  and is defined by

$$(T_{\mu^*}J)(x) = \begin{cases} pJ(2x) + (1-p)J(0) & \text{if } 0 < x \leq 1/2, \\ pJ(1) + (1-p)J(2x-1) & \text{if } 1/2 \leq x < 1, \end{cases}$$

$$(T_{\mu^*}J)(0) = 0, \quad (T_{\mu^*}J)(1) = 1. \quad (5.8)$$

Consider the functions  $J_0$ ,  $T_{\mu^*}J_0$ ,  $\dots$ ,  $T_{\mu^*}^k J_0$ ,  $\dots$ , where  $J_0$  is the zero function [ $J_0(x) = 0$  for all  $x \in [0, 1]$ ]. We have

$$J_{\mu^*}(x) = \lim_{k \rightarrow \infty} (T_{\mu^*}^k J_0)(x), \quad x \in [0, 1]. \quad (5.9)$$

Furthermore, the functions  $T_{\mu^*}^k J_0$  can be shown to be monotonically nondecreasing in the interval  $[0, 1]$ . Hence, by Eq. (5.9),  $J_{\mu^*}$  is also monotonically nondecreasing.

Consider now for  $n = 0, 1, \dots$  the sets

$$S_n = \{x \in [0, 1] \mid x = k2^{-n}, k = \text{nonnegative integer}\}.$$

It is straightforward to verify that

$$(T_p^m, J_0)(x) = (T_p^n, J_0)(x), \quad x \in S_{n-1}, \quad m \geq n \geq 1.$$

As a result of this equality and Eq. (5.9),

$$J_{\mu^*}(x) = (T_p^n, J_0)(x), \quad x \in S_{n-1}, \quad n \geq 1. \quad (5.10)$$

A further fact that may be verified by using induction and Eqs. (5.8) and (5.10) is that for any nonnegative integers  $k, n$  for which  $0 \leq k2^{-n} < (k+1)2^{-n} \leq 1$ , we have

$$p^n \leq J_{\mu^*}((k+1)2^{-n}) - J_{\mu^*}(k2^{-n}) \leq (1-p)^n. \quad (5.11)$$

Since any number in  $[0, 1]$  can be approximated arbitrarily closely from above and below by numbers of the form  $k2^{-n}$ , and since  $J_{\mu^*}$  has been shown to be monotonically nondecreasing, it follows from Eq. (5.11) that  $J_{\mu^*}$  is continuous and strictly increasing. **Q.E.D.**

We are now in a position to prove the following proposition.

**Proposition 5.1:** The bold strategy is an optimal stationary gambling policy.

**Proof:** We will prove the sufficiency condition

$$J_{\mu^*}(x) \geq pJ_{\mu^*}(x+u) + (1-p)J_{\mu^*}(x-u), \quad x \in [0, 1], \quad u \in [0, 1] \cap [0, 1-x]. \quad (5.12)$$

In view of the continuity of  $J_{\mu^*}$  established in the previous lemma, it is sufficient to establish Eq. (5.12) for all  $x \in [0, 1]$  and  $u \in [0, x] \cap [0, 1-x]$  that belong to the union  $\bigcup_{n=0}^{\infty} S_n$  of the sets  $S_n$  defined by

$$S_n = \{z \in [0, 1] \mid z = k2^{-n}, k = \text{nonnegative integer}\}.$$

We will use induction. By using the fact that  $J_{\mu^*}(0) = 0$ ,  $J_{\mu^*}(1/2) = p$ , and  $J_{\mu^*}(1) = 1$ , we can show that Eq. (5.12) holds for all  $x$  and  $u$  in  $S_0$  and  $S_1$ . Assume that Eq. (5.12) holds for all  $x, u \in S_n$ . We will show that it holds for all  $x, u \in S_{n+1}$ .

For any  $x, u \in S_{n+1}$  with  $u \in [0, x] \cap [0, 1-x]$ , there are four possibilities:

1.  $x+u \leq 1/2$ ,
2.  $x-u \geq 1/2$ ,
3.  $x-u \leq x \leq 1/2 \leq x+u$ ,

$$4. x-u \leq 1/2 \leq x \leq x+u,$$

We will prove Eq. (5.12) for each of these cases.

*Case 1.* If  $x, u \in S_{n+1}$ , then  $2x \in S_n$  and  $2u \in S_n$ , and by the induction hypothesis

$$J_{\mu^*}(2x) = pJ_{\mu^*}(2x+2u) - (1-p)J_{\mu^*}(2x-2u) \geq 0. \quad (5.13)$$

If  $x+u \leq 1/2$ , then by Eq. (5.5)

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= p(J_{\mu^*}(2x) - pJ_{\mu^*}(2x+2u) - (1-p)J_{\mu^*}(2x-2u)) \end{aligned}$$

and using Eq. (5.13), the desired relation Eq. (5.12) is proved for the case under consideration.

*Case 2.* If  $x, u \in S_{n+1}$ , then  $(2x-u) \in S_n$  and  $2u \in S_n$ , and by the induction hypothesis

$$J_{\mu^*}(2x-u) = pJ_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u-1) \geq 0.$$

If  $x-u \geq 1/2$ , then by Eq. (5.5)

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= p + (1-p)J_{\mu^*}(2x-1) - p(p + (1-p)J_{\mu^*}(2x+2u-1)) \\ &\quad - (1-p)(p + (1-p)J_{\mu^*}(2x-2u-1)) \\ &= (1-p)(J_{\mu^*}(2x-1) - pJ_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u-1)) \\ &\geq 0, \end{aligned}$$

and Eq. (5.12) follows from the preceding relations.

*Case 3.* Using Eq. (5.5), we have

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= pJ_{\mu^*}(2x) - p(p + (1-p)J_{\mu^*}(2x+2u-1)) + p(1-p)J_{\mu^*}(2x-2u) \\ &= p(J_{\mu^*}(2x) - p - (1-p)J_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u)). \end{aligned}$$

Now we must have  $x \geq \frac{1}{4}$ , for otherwise  $u < \frac{1}{4}$  and  $x+u < 1/2$ . Hence  $2x \geq 1/2$  and the sequence of equalities can be continued as follows:

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= p(p + (1-p)J_{\mu^*}(4x-1) - p) \\ &\quad - (1-p)J_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u)) \\ &= p(1-p)(J_{\mu^*}(4x-1) - J_{\mu^*}(2x+2u-1) - J_{\mu^*}(2x-2u)) \\ &= (1-p)(J_{\mu^*}(2x-1/2) - pJ_{\mu^*}(2x+2u-1) - pJ_{\mu^*}(2x-2u)). \end{aligned}$$

Since  $p \leq (1-p)$ , the last expression is greater than or equal to both

$$(1-p)(J_{\mu^*}(2x - 1/2) - pJ_{\mu^*}(2x + 2u - 1) - (1-p)J_{\mu^*}(2x - 2u))$$

and

$$(1-p)(J_{\mu^*}(2x - 1/2) - (1-p)J_{\mu^*}(2x + 2u - 1) - pJ_{\mu^*}(2x - 2u)).$$

Now for  $x, u \in S_{n+1}$ , and  $n \geq 1$ , we have  $(2x - 1/2) \in S_n$  and  $(2u - 1/2) \in S_n$  if  $(2u - 1/2) \in [0, 1]$ , and  $(1/2 - 2u) \in S_n$  if  $(1/2 - 2u) \in [0, 1]$ . By the induction hypothesis, the first or the second of the preceding expressions is nonnegative, depending on whether  $2x + 2u - 1 \geq 2x - 1/2$  or  $2x - 2u \geq 2x - 1/2$  (i.e.,  $u \geq \frac{1}{4}$  or  $u \leq \frac{1}{4}$ ). Hence Eq. (5.12) is proved for case 3.

*Case 4.* The proof resembles the one for case 3. Using Eq. (5.5), we have

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= p + (1-p)J_{\mu^*}(2x-1) - p(p + (1-p)J_{\mu^*}(2x+2u-1)) \\ &\quad - (1-p)pJ_{\mu^*}(2x-2u) \\ &= p(1-p) \\ &\quad + (1-p)(J_{\mu^*}(2x-1) - pJ_{\mu^*}(2x+2u-1) - pJ_{\mu^*}(2x-2u)). \end{aligned}$$

We must have  $x \leq \frac{3}{4}$  for otherwise  $u < \frac{1}{4}$  and  $x-u > \frac{1}{2}$ . Hence  $0 \leq 2x-1 \leq 1/2 \leq 2x-1/2 \leq 1$ , and using Eq. (5.5) we have

$$(1-p)J_{\mu^*}(2x-1) = (1-p)pJ_{\mu^*}(4x-2) = p(J_{\mu^*}(2x-1/2) - p).$$

Using the preceding relations, we obtain

$$\begin{aligned} J_{\mu^*}(x) &= pJ_{\mu^*}(x+u) - (1-p)J_{\mu^*}(x-u) \\ &= p(1-p) + p(J_{\mu^*}(2x-1/2) - p) - p(1-p)J_{\mu^*}(2x+2u-1) \\ &\quad - p(1-p)J_{\mu^*}(2x-2u) \\ &= p((1-2p) + J_{\mu^*}(2x-1/2) - (1-p)J_{\mu^*}(2x+2u-1) \\ &\quad - (1-p)J_{\mu^*}(2x-2u)). \end{aligned}$$

These relations are equal to both

$$\begin{aligned} p((1-2p)(1 - J_{\mu^*}(2x+2u-1)) \\ + J_{\mu^*}(x-1/2) - pJ_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u)) \end{aligned}$$

and

$$\begin{aligned} p((1-2p)(1 - J_{\mu^*}(2x-2u)) \\ + J_{\mu^*}(2x-1/2) - (1-p)J_{\mu^*}(2x+2u-1) - pJ_{\mu^*}(2x-2u)). \end{aligned}$$

Since  $0 \leq J_{\mu^*}(2x+2u-1) \leq 1$  and  $0 \leq J_{\mu^*}(2x-2u) \leq 1$ , these expressions are greater than or equal to both

$$p(J_{\mu^*}(2x-1/2) - pJ_{\mu^*}(2x+2u-1) - (1-p)J_{\mu^*}(2x-2u))$$

and

$$p(J_{\mu^*}(2x-1/2) - (1-p)J_{\mu^*}(2x+2u-1) - pJ_{\mu^*}(2x-2u))$$

and the result follows as in case 3. **Q.E.D.**

We note that the bold strategy is not the unique optimal stationary gambling strategy. For a characterization of all optimal strategies, see [DuS65], p. 90. Several other gambling problems where strategies of the bold type are optimal are described in [DuS65], Chapters 5 and 6.

### 3.6 NONSTATIONARY AND PERIODIC PROBLEMS

The standing assumption so far in this chapter has been that the problem involves a stationary system and a stationary cost per stage (except for the presence of the discount factor). Problems with nonstationary system or cost per stage arise occasionally in practice or in theoretical studies and are thus of some interest. It turns out that such problems can be converted to stationary ones by a simple reformulation. We can then obtain results analogous to those obtained earlier for stationary problems.

Consider a nonstationary system of the form

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \dots,$$

and a cost function of the form

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} E_{w_k^{(0,1,\dots,N-1)}} \left\{ \sum_{k=0}^{N-1} \alpha^k g_k(x_k, \mu_k(x_k), w_k) \right\}. \quad (6.1)$$

In these equations, for each  $k$ ,  $x_k$  belongs to a space  $S_k$ ,  $u_k$  belongs to a space  $C_k$  and satisfies  $u_k \in U_k(x_k)$  for all  $x_k \in S_k$ , and  $w_k$  belongs to a countable space  $D_k$ . The sets  $S_k$ ,  $C_k$ ,  $U_k(x_k)$ ,  $D_k$  may differ from one stage to the next. The random disturbances  $w_k$  are characterized by probabilities  $P_k(\cdot | x_k, u_k)$ , which depend on  $x_k$  and  $u_k$  as well as the time index  $k$ . The set of admissible policies  $\Pi$  is the set of all sequences  $\pi = \{\mu_0, \mu_1, \dots\}$  with  $\mu_k : S_k \mapsto C_k$  and  $\mu_k(x_k) \in U_k(x_k)$  for all  $x_k \in S_k$  and  $k = 0, 1, \dots$ . The functions  $g_k : S_k \times C_k \times D_k \mapsto \mathbb{R}$  are given and are assumed to satisfy one of the following three assumptions:

**Assumption D'**: We have  $\alpha < 1$ , and the functions  $g_k$  satisfy, for all  $k = 0, 1, \dots$ ,

$$|g_k(x_k, u_k, w_k)| \leq M, \quad \text{for all } (x_k, u_k, w_k) \in S_k \times C_k \times D_k,$$

where  $M$  is some scalar.

**Assumption P'**: The functions  $g_k$  satisfy, for all  $k = 0, 1, \dots$ ,

$$0 \leq g_k(x_k, u_k, w_k), \quad \text{for all } (x_k, u_k, w_k) \in S_k \times C_k \times D_k.$$

**Assumption N'**: The functions  $g_k$  satisfy, for all  $k = 0, 1, \dots$ ,

$$g_k(x_k, u_k, w_k) \leq 0, \quad \text{for all } (x_k, u_k, w_k) \in S_k \times C_k \times D_k.$$

We will refer to the problem formulated as the *nonstationary problem* (NSP for short). We can get an idea on how the NSP can be converted to a stationary problem by considering the special case where the state space is the same for each stage (i.e.,  $S_k = S$  for all  $k$ ). We consider an augmented state

$$\tilde{x} = (x, k),$$

where  $x \in S$ , and  $k$  is the time index. The new state space is  $\tilde{S} = S \times K$ , where  $K$  denotes the set of nonnegative integers. The augmented system evolves according to

$$(x, k) \rightarrow (f_k(x, u_k, w_k), k + 1), \quad (x, k) \in \tilde{S},$$

Similarly, we can define a cost per stage as

$$\hat{g}((x, k), u_k, w_k) = g_k(x, u_k, w_k), \quad (x, k) \in \tilde{S}.$$

It is evident that the problem corresponding to the augmented system is stationary. If we restrict attention to initial states  $\tilde{x}_0 \in S \times \{0\}$ , it can be seen that this stationary problem is equivalent to the NSP.

Let us now consider the more general case. To simplify notation, we will assume that the state spaces  $S_i$ ,  $i = 0, 1, \dots$ , the control spaces  $C_i$ ,

$i = 0, 1, \dots$ , and the disturbance spaces  $D_i$ ,  $i = 0, 1, \dots$ , are all mutually disjoint. This assumption does not involve a loss of generality since, if necessary, we may relabel the elements of  $S_i$ ,  $C_i$ , and  $D_i$  without affecting the structure of the problem. Define now a new state space  $S$ , a new control space  $C$ , and a new (countable) disturbance space  $D$  by

$$S = \bigcup_{i=0}^{\infty} S_i, \quad C = \bigcup_{i=0}^{\infty} C_i, \quad D = \bigcup_{i=0}^{\infty} D_i.$$

Introduce a new (stationary) system

$$\tilde{x}_{k+1} = f(\tilde{x}_k, \tilde{u}_k, \tilde{w}_k), \quad k = 0, 1, \dots, \quad (6.2)$$

where  $\tilde{x}_k \in S$ ,  $\tilde{u}_k \in C$ ,  $\tilde{w}_k \in D$ , and the system function  $f : S \times C \times D \mapsto S$  is defined by

$$f(\tilde{x}, \tilde{u}, \tilde{w}) = f_i(\tilde{w}, \tilde{u}, w), \quad \text{if } \tilde{x} \in S_i, \quad u \in C_i, \quad w \in D_i, \quad i = 0, 1, \dots$$

For triplets  $(\tilde{x}, \tilde{u}, \tilde{w})$ , where for some  $i = 0, 1, \dots$ , we have  $\tilde{x} \in S_i$ , but  $\tilde{u} \notin C_i$  or  $\tilde{w} \notin D_i$ , the definition of  $f$  is immaterial; any definition is adequate for our purposes in view of the control constraints to be introduced. The control constraint is taken to be  $\tilde{u} \in U(\tilde{x})$  for all  $\tilde{x} \in S$ , where  $U(\cdot)$  is defined by

$$U(\tilde{x}) = U_i(\tilde{x}), \quad \text{if } \tilde{x} \in S_i, \quad i = 0, 1, \dots$$

The disturbance  $\tilde{w}$  is characterized by probabilities  $P(\tilde{w} | \tilde{x}, \tilde{u})$  such that

$$P(\tilde{w} \in D_i | \tilde{x} \in S_i, \tilde{u} \in C_i) = 1, \quad i = 0, 1, \dots$$

$$P(\tilde{w} \notin D_i | \tilde{x} \in S_i, \tilde{u} \in C_i) = 0, \quad i = 0, 1, \dots$$

Furthermore, for any  $w_i \in D_i$ ,  $x_i \in S_i$ ,  $u_i \in C_i$ ,  $i = 0, 1, \dots$ , we have

$$P(w_i | x_i, u_i) = P_i(w_i | x_i, u_i).$$

We also introduce a new cost function

$$\tilde{J}_{\pi}(\tilde{x}_0) = \lim_{N \rightarrow \infty} E_{\tilde{w}_k^N, k=0,1,\dots,N-1} \left\{ \sum_{k=0}^{N-1} \alpha^k g(\tilde{x}_k, \mu_k(\tilde{x}_k), \tilde{w}_k) \right\}, \quad (6.3)$$

where the (stationary) cost per stage  $g : S \times C \times D \mapsto \mathbb{R}$  is defined for all  $i = 0, 1, \dots$  by

$$g(x, u, w) = g_i(x, u, w), \quad \text{if } x \in S_i, \quad u \in C_i, \quad w \in D_i.$$

For triplets  $(\tilde{x}, \tilde{u}, \tilde{w})$ , where for some  $i = 0, 1, \dots$ , we have  $\tilde{x} \in S_i$  but  $\tilde{u} \notin C_i$  or  $\tilde{w} \notin D_i$ , any definition of  $g$  is adequate provided  $|g(\tilde{x}, \tilde{u}, \tilde{w})| \leq M$  for all  $(\tilde{x}, \tilde{u}, \tilde{w})$  when Assumption D' holds,  $0 \leq g(\tilde{x}, \tilde{u}, \tilde{w})$  when P' holds, and

$g(\hat{x}, \hat{u}, \hat{w}) \leq 0$  when  $N'$  holds. The set of admissible policies  $\hat{\Pi}$  for the new problem consists of all sequences  $\hat{\pi} = \{\hat{\mu}_0, \hat{\mu}_1, \dots\}$ , where  $\hat{\mu}_k : S \mapsto C$  and  $\hat{\mu}_k(\hat{x}) \in U(\hat{x})$  for all  $\hat{x} \in S$  and  $k = 0, 1, \dots$ .

The construction given defines a problem that clearly fits the framework of the infinite horizon total cost problem. We will refer to this problem as the *stationary problem* (SP for short).

It is important to understand the nature of the intimate connection between the NSP and the SP formulated here. Let  $\pi = \{\mu_0, \mu_1, \dots\}$  be an admissible policy for the NSP. Also, let  $\hat{\pi} = \{\hat{\mu}_0, \hat{\mu}_1, \dots\}$  be an admissible policy for the SP such that

$$\hat{\mu}_i(\hat{x}) = \mu_i(x), \quad \text{if } \hat{x} \in S_i, \quad i = 0, 1, \dots \quad (6.4)$$

Let  $x_0 \in S_0$  be the initial state for the NSP and consider the same initial state for the SP (i.e.,  $\hat{x}_0 = x_0 \in S_0$ ). Then the sequence of states  $\{\hat{x}_i\}$  generated in the SP will satisfy  $\hat{x}_i \in S_i$ ,  $i = 0, 1, \dots$ , with probability 1 (i.e., the system will move from the set  $S_0$  to the set  $S_1$ , then to  $S_2$ , etc., just as in the NSP). Furthermore, the probabilistic law of generation of states and costs is identical in the NSP and the SP. As a result, it is easy to see that for any admissible policies  $\pi$  and  $\hat{\pi}$  satisfying Eq. (6.4) and initial states  $x_0, \hat{x}_0$  satisfying  $x_0 = \hat{x}_0 \in S_0$ , the sequence of generated states in the NSP and the SP is the same ( $x_i = \hat{x}_i$ , for all  $i$ ) provided the generated disturbances  $w_i$  and  $\hat{w}_i$  are also the same for all  $i$  ( $w_i = \hat{w}_i$ , for all  $i$ ). Furthermore, if  $\pi$  and  $\hat{\pi}$  satisfy Eq. (6.4), we have  $J_\pi(x_0) = J_{\hat{\pi}}(\hat{x}_0)$  if  $x_0 = \hat{x}_0 \in S_0$ . Let us also consider the optimal cost functions for the NSP and the SP:

$$J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_0), \quad x_0 \in S_0,$$

$$\tilde{J}^*(\hat{x}_0) = \min_{\hat{\pi} \in \hat{\Pi}} J_{\hat{\pi}}(\hat{x}_0), \quad \hat{x}_0 \in S_0.$$

Then it follows from the construction of the SP that

$$J^*(x_0) = \tilde{J}^*(\hat{x}_0, i), \quad \text{if } \hat{x}_0 \in S_i, \quad i = 0, 1, \dots, \quad (6.5)$$

where, for all  $i = 0, 1, \dots$ ,

$$\tilde{J}^*(\hat{x}_0, i) = \min_{\pi \in \Pi} \lim_{N \rightarrow \infty} E \left\{ \sum_{k=i}^{N-1} \alpha^{k-i} g_k(x_k, \mu_k(x_k), w_k) \right\}, \quad (6.6)$$

if  $\hat{x}_0 = x_i \in S_i$ . Note that in this equation, the right-hand side is defined in terms of the data of the NSP. As a special case of this equation, we obtain

$$\tilde{J}^*(\hat{x}_0) = \tilde{J}^*(\hat{x}_0, 0) = J^*(x_0), \quad \text{if } \hat{x}_0 = x_0 \in S_0. \quad (6.7)$$

Thus the optimal cost function  $J^*$  of the NSP can be obtained from the optimal cost function  $\tilde{J}^*$  of the SP. Furthermore, if  $\hat{\pi}^* = \{\hat{\mu}_0^*, \hat{\mu}_1^*, \dots\}$  is an optimal policy for the SP, then the policy  $\pi^* = \{\mu_0^*, \mu_1^*, \dots\}$  defined by

$$\mu_i^*(x_i) = \hat{\mu}_i^*(x_i), \quad \text{for all } x_i \in S_i, \quad i = 0, 1, \dots, \quad (6.8)$$

is an optimal policy for the NSP. Thus optimal policies for the SP yield optimal policies for the NSP via Eq. (6.8). Another point to be noted is that if Assumption D' (P', N') is satisfied for the NSP, then Assumption D (P, N) introduced earlier in this chapter is satisfied for the SP.

These observations show that one may analyze the NSP by means of the SP. Every result given in the preceding sections when applied to the SP yields a corresponding result for the NSP. We will just provide the form of the optimality equation for the NSP in the following proposition.

**Proposition 6.1:** Under Assumption D' (P', N'), there holds

$$J^*(x_0) = \tilde{J}^*(x_0, 0), \quad x_0 \in S_0,$$

where for all  $i = 0, 1, \dots$ , the functions  $\tilde{J}^*(\cdot, i)$  map  $S_i$  into  $\mathbb{R} ([0, \infty], [-\infty, 0])$ , are given by Eq. (6.6), and satisfy for all  $x_i \in S_i$  and  $i = 0, 1, \dots$ ,

$$\tilde{J}^*(x_i, i) = \min_{u_i \in U_i(x_i)} E \{ g_i(x_i, u_i, w_i) + \alpha \tilde{J}^*(f_i(x_i, u_i, w_i), i+1) \}. \quad (6.9)$$

Under Assumption D' the functions  $\tilde{J}^*(\cdot, i)$ ,  $i = 0, 1, \dots$ , are the unique bounded solutions of the set of equations Eq. (6.9). Furthermore, under Assumption D' or P', if  $\mu_i^*(x_i) \in U_i(x_i)$  attains the minimum in Eq. (6.9) for all  $x_i \in S_i$  and  $i$ , then the policy  $\pi^* = \{\mu_0^*, \mu_1^*, \dots\}$  is optimal for the NSP.

### Periodic Problems

Assume within the framework of the NSP that there exists an integer  $p \geq 2$  (called the *period*) such that for all integers  $i$  and  $j$  with  $|i - j| = mp$ ,  $m = 1, 2, \dots$ , we have

$$S_i = S_j, \quad C_i = C_j, \quad D_i = D_j, \quad U_i(\cdot) = U_j(\cdot),$$

$$f_i = f_j, \quad g_i = g_j, \quad P_i(\cdot | x, j) = P_j(\cdot | x, i), \quad (x, u) \in S_i \times C_i.$$

We assume that the spaces  $S_i$ ,  $C_i$ ,  $D_i$ ,  $i = 0, 1, \dots, p-1$ , are mutually disjoint. We define new state, control, and disturbance spaces by

$$S = \bigcup_{i=0}^{p-1} S_i, \quad C = \bigcup_{i=0}^{p-1} C_i, \quad D = \bigcup_{i=0}^{p-1} D_i.$$

The optimality equation for the equivalent stationary problem reduces to the system of  $p$  equations

$$J^*(x_0, 0) = \min_{u_0 \in U_0(x_0)} E \{ g_0(x_0, u_0, w_0) + \alpha J^*(f_0(x_0, u_0, w_0), 1) \},$$

$$J^*(x_1, 1) = \min_{u_1 \in U_1(x_1)} E \{ g_1(x_1, u_1, w_1) + \alpha J^*(f_1(x_1, u_1, w_1), 2) \}.$$

⋮

$$\begin{aligned} J^*(x_{p-1}, p-1) = & \min_{u_{p-1} \in U_{p-1}(x_{p-1})} E \{ g_{p-1}(x_{p-1}, u_{p-1}, w_{p-1}) \\ & + \alpha J^*(f_{p-1}(x_{p-1}, u_{p-1}, w_{p-1}), 0) \}. \end{aligned}$$

These equations may be used to obtain (under Assumption D' or P') a periodic policy of the form  $\{\mu_0^*, \dots, \mu_{p-1}^*, \mu_p^*, \dots, \mu_{p-1}^*, \dots\}$  whenever the minimum of the right-hand side is attained for all  $x_i$ ,  $i = 0, 1, \dots, p-1$ . When all spaces involved are finite, an optimal policy may be found by means of the algorithms of Section 1.3, appropriately adapted to the corresponding SP.

### 3.7 NOTES, SOURCES, AND EXERCISES

Undiscounted problems and discounted problems with unbounded cost per stage were first analyzed systematically in [DuS65], [Bla65], and [Str66]. An extensive treatment, which also resolves the associated measurability questions, is [BeS78]. Sufficient conditions for convergence of the value iteration method under Assumption P (cf. Props. 1.6 and 1.7) were derived independently in [Ber77] and [Sch75]. The former reference also derives necessary conditions for convergence. Problems involving convexity assumptions are analyzed in [Ber73b].

We have bypassed a number of complex theoretical issues relating to stationary policies that historically have played an important role in the development of the subject of this chapter. The main question is to what extent is it possible to restrict attention to stationary policies. Much theoretical work has been done on this question [BeS79], [Bla65], [Bla70], [DuS65], [Fei78], [FeS83], [Fei92a], [Fei92b], [Orn69], and some aspects are still open. Suppose, for example, that we are given an  $\epsilon > 0$ . One issue is whether there exists an  $\epsilon$ -optimal stationary policy, that is, a stationary policy  $\mu$  such that

$$J_\mu(x) \leq J^*(x) + \epsilon, \quad \text{for all } x \in S \text{ with } J^*(x) > -\infty,$$

$$J_\mu(x) \leq -\frac{1}{c}, \quad \text{for all } x \in S \text{ with } J^*(x) = -\infty.$$

The answer is positive under any one of the following conditions:

1. Assumption P holds and  $\alpha < 1$  (see Exercise 3.8).
2. Assumption N holds,  $S$  is a finite set,  $\alpha = 1$ , and  $J^*(x) > -\infty$  for all  $x \in S$  (see Exercise 3.11 or [Bla65], [Bla70], and [Orn69]).
3. Assumption N holds,  $S$  is a countable set,  $\alpha = 1$ , and the problem is deterministic (see [BeS79]).

The answer can be negative under any one of the following conditions:

1. Assumption P holds and  $\alpha = 1$  (see Exercise 3.8).
2. Assumption N holds and  $\alpha < 1$  (see Exercise 3.11 or [BeS79]).

The existence of an  $\epsilon$ -optimal stationary policy for stochastic shortest path problems with a finite state space, but under somewhat different assumptions than the ones of Section 2.1 is established in [Fei92b].

Another issue is whether there exists an optimal stationary policy whenever there exists an optimal policy for each initial state. This is true under Assumption P (see Exercise 3.9). It is also true (but very hard to prove) under Assumption N if  $J^*(x) > -\infty$  for all  $x \in S$ ,  $\alpha = 1$ , and the disturbance space  $D$  is countable [Bla70], [DuS65], [Orn69]. Simple two-state examples can be constructed showing that the result fails to hold if  $\alpha = 1$  and  $J^*(x) = -\infty$  for some state  $x$  (see Exercise 3.10). However, these examples rely on the presence of a stochastic element in the problem. If the problem is deterministic, stronger results are available: one can find an optimal stationary policy if there exists an optimal policy at each initial state and either  $\alpha = 1$  or  $\alpha < 1$  and  $J^*(x) > -\infty$  for all  $x \in S$ . These results also require a difficult proof [BeS79].

The gambling problem and its solution are taken from [DuS65]. In [Bil83], a surprising property of the optimal reward function  $J^*$  for this problem is shown:  $J^*$  is almost everywhere differentiable with derivative zero, yet it is strictly increasing, taking values that range from 0 to 1.

---

### EXERCISES

---

#### 3.1

Let  $S = [0, \infty)$  and  $C = U(x) = (0, \infty)$  be the state and control spaces,

respectively, let the system equation be

$$x_{k+1} = \left( \frac{2}{\alpha} \right) x_k + u_k, \quad k = 0, 1, \dots,$$

where  $\alpha$  is the discount factor, and let

$$g(x_k, u_k) = x_k + u_k$$

be the cost per stage. Show that for this deterministic problem, Assumption P holds and that  $J^*(x) = \infty$  for all  $x \in S$ , but  $(T^k J_0)(0) = 0$  for all  $k$  [ $J_0$  is the zero function,  $J_0(x) = 0$ , for all  $x \in S$ ].

### 3.2

Let Assumption P hold and consider the finite-state case  $S = D = \{1, 2, \dots, n\}$ ,  $\alpha = 1$ ,  $x_{k+1} = w_k$ . The mapping  $T$  is represented as

$$(TJ)(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)J(j) \right], \quad i = 1, \dots, n,$$

where  $p_{ij}(u)$  denotes the transition probability that the next state will be  $j$  when the current state is  $i$  and control  $u$  is applied. Assume that the sets  $U(i)$  are compact subsets of  $\mathbb{R}^m$  for all  $i$ , and that  $p_{ij}(u)$  and  $g(i, u)$  are continuous on  $U(i)$  for all  $i$  and  $j$ . Show that  $\lim_{k \rightarrow \infty} (T^k J_0)(i) = J^*(i)$ , where  $J_0(i) = 0$  for all  $i = 1, \dots, n$ . Show also that there exists an optimal stationary policy.

### 3.3

Consider a deterministic problem involving a linear system

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, \dots,$$

where the pair  $(A, B)$  is controllable and  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$ . Assume no constraints on the control and a cost per stage  $g$  satisfying

$$0 \leq g(x, u), \quad (x, u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

Assume furthermore that  $g$  is continuous in  $x$  and  $u$ , and that  $g(x_n, u_n) \rightarrow \infty$  if  $\{x_n\}$  is bounded and  $\|u_n\| \rightarrow \infty$ .

- (a) Show that for a discount factor  $\alpha < 1$ , the optimal cost satisfies  $0 \leq J^*(x) < \infty$ , for all  $x \in \mathbb{R}^n$ . Furthermore, there exists an optimal stationary policy and

$$\lim_{k \rightarrow \infty} (T^k J_0)(x) = J^*(x), \quad x \in \mathbb{R}^n.$$

- (b) Show that the same is true, except perhaps for  $J^*(x) < \infty$ , when the system is of the form  $x_{k+1} = f(x_k, u_k)$ , with  $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$  being a continuous function.

- (c) Prove the same results assuming that the control is constrained to lie in a compact set  $U \in \mathbb{R}^m$  [ $U(x) = U$  for all  $x$ ] in place of the assumption  $g(x_n, u_n) \rightarrow \infty$  if  $\{x_n\}$  is bounded and  $\|u_n\| \rightarrow \infty$ . Hint: Show that  $T^k J_0$  is real valued and continuous for every  $k$ , and use Prop. 1.7.

### 3.4

Under Assumption P, let  $\mu$  be such that for all  $x \in S$ ,  $\mu(x) \in U(x)$  and

$$(T_\mu J^*)(x) \leq (TJ^*)(x) + \epsilon,$$

where  $\epsilon$  is some positive scalar. Show that, if  $\alpha < 1$ ,

$$J_\mu(x) \leq J^*(x) + \frac{\epsilon}{1-\alpha}, \quad x \in S.$$

*Hint:* Show that  $(T_\mu^k J^*)(x) \leq J^*(x) + \sum_{i=0}^{k-1} \alpha^i \epsilon$ . Alternatively, let  $J' = J^* + (\epsilon/(1-\alpha))e$ , show that  $T_\mu J' \leq J'$ , and use Cor. 7.1.1.

### 3.5

Under Assumption P or N, show that if  $\alpha < 1$  and  $J' : S \mapsto \mathbb{R}$  is a bounded function satisfying  $J' = TJ'$ , then  $J' = J^*$ . Hint: Under P, let  $r$  be a scalar such that  $J^* + re \geq J'$ . Argue that  $J^* \geq J'$  and use Prop. 1.2(a).

### 3.6

We want to find a scalar sequence  $\{u_0, u_1, \dots\}$  that satisfies  $\sum_{k=0}^{\infty} u_k \leq c$ ,  $u_k \geq 0$ , for all  $k$ , and maximizes  $\sum_{k=0}^{\infty} g(u_k)$ , where  $c > 0$  and  $g(u) \geq 0$  for all  $u \geq 0$ ,  $g(0) = 0$ . Assume that  $g$  is monotonically nondecreasing on  $[0, \infty)$ . Show that the optimal value of the problem is  $J^*(c)$ , where  $J^*$  is a monotonically nondecreasing function on  $[0, \infty)$  satisfying  $J^*(0) = 0$  and

$$J^*(x) = \max_{0 \leq u \leq x} \{g(u) + J^*(x-u)\}, \quad x \in [0, \infty).$$

### 3.7

Let Assumption P hold and assume that  $\pi^* = \{\mu_0^*, \mu_1^*, \dots\} \in \Pi$  satisfies  $J^* = T_{\mu_k^*} J^*$  for all  $k$ . Show that  $\pi^*$  is optimal, i.e.,  $J_{\pi^*} = J^*$ .

### 3.8

Under Assumption P, show that, given  $\epsilon > 0$ , there exists a policy  $\pi_\epsilon \in \Pi$  such that  $J_{\pi_\epsilon}(x) \leq J^*(x) + \epsilon$  for all  $x \in S$ , and that for  $\alpha < 1$  the policy  $\pi_\epsilon$  can be taken stationary. Give an example where  $\alpha = 1$  and for each stationary policy  $\pi$  we have  $J_\pi(x) = \infty$ , while  $J^*(x) = 0$  for all  $x$ . Hint: See the proof of Prop. 1.1.

## 3.9

Under Assumption P, show that if there exists an optimal policy (a policy  $\pi^* \in \Pi$  such that  $J_{\pi^*} = J^*$ ), then there exists an optimal stationary policy.

## 3.10

Use the following counterexample to show that the result of Exercise 3.9 may fail to hold under Assumption N if  $J^*(x) = -\infty$  for some  $x \in S$ . Let  $S = D = \{0, 1\}$ ,  $f(x, u, w) = w$ ,  $g(x, u, w) = u$ ,  $U(0) = (-\infty, 0]$ ,  $U(1) = \{0\}$ ,  $p(w=0|x=0, u)=\frac{1}{2}$ , and  $p(w=1|x=1, u)=1$ . Show that  $J^*(0) = -\infty$ ,  $J^*(1) = 0$  and that the admissible nonstationary policy  $\{\mu_0^k, \mu_1^k, \dots\}$  with  $\mu_0^k(0) = -(2/\alpha)^k$  is optimal. Show that every stationary policy  $\mu$  satisfies  $J_\mu(0) = (2/(2+\alpha))\mu(0)$ ,  $J_\mu(1) = 0$  (see [Bla70], [Dug65], and [Ora69] for related analysis).

## 3.11

Show that the result of Exercise 3.8 holds under Assumption N if  $S$  is a finite set,  $\alpha = 1$ , and  $J^*(x) > -\infty$  for all  $x \in S$ . Construct a counterexample to show that the result can fail to hold if  $S$  is countable and  $\alpha < 1$  [even if  $J^*(x) > -\infty$  for all  $x \in S$ ]. Hint: Consider an integer  $N$  such that the  $N$ -stage optimal cost  $J_N$  satisfies  $J_N(x) \leq J^*(x) + \epsilon$  for all  $x$ . For a counterexample, see [BcS79].

## 3.12 (Deterministic Linear-Quadratic Problems)

Consider the deterministic linear-quadratic problem involving the system

$$x_{k+1} = Ax_k + Bu_k$$

and the cost

$$J_\pi(x_0) = \sum_{k=0}^N (x_k' Q x_k + \mu_k(x_k)' R \mu_k(x_k)).$$

We assume that  $R$  is positive definite symmetric,  $Q$  is of the form  $C'C$ , and the pairs  $(A, B)$ ,  $(A, C)$  are controllable and observable, respectively. Use the theory of Sections 4.1 of Vol. I and 8.4 to show that the stationary policy  $\mu^*$  with

$$\mu^*(x) = -(B'KB + R)^{-1}B'Kx$$

is optimal, where  $K$  is the unique positive semidefinite symmetric solution of the algebraic Riccati equation (cf. Section 4.1 of Vol. I):

$$K = A' \left( K - KB(B'KB + R)^{-1}B'K \right) A + Q,$$

Provide a similar result under an appropriate controllability assumption for the case of a periodic deterministic linear system and a periodic quadratic cost (cf. Section 3.6).

## 3.13

Consider the linear-quadratic problem of Section 3.2 with the only difference that the disturbances  $w_k$  have zero mean, but their covariance matrices are nonstationary and uniformly bounded over  $k$ . Show that the optimal control law remains unchanged.

## 3.14 (Periodic Linear-Quadratic Problems)

Consider the linear system

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, 1, \dots,$$

and the quadratic cost

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \mathbb{E}_{w_k} \left[ \sum_{k=0}^{N-1} \alpha^k (x_k' Q_k x_k + u_k' R_k u_k) \right],$$

where the matrices have appropriate dimensions,  $Q_k$  and  $R_k$  are positive semidefinite and positive definite symmetric, respectively, for all  $k$ , and  $0 < \alpha < 1$ . Assume that the system and cost are periodic with period  $p$  (cf. Section 3.6), that the controls are unconstrained, and that the disturbances are independent, and have zero mean and finite covariance. Assume further that the following (controllability) condition is in effect.

For any state  $\bar{x}_0$ , there exists a finite sequence of controls  $\{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_r\}$  such that  $\bar{x}_{r+1} = 0$ , where  $\bar{x}_{r+1}$  is generated by

$$\bar{x}_{k+1} = A_k \bar{x}_k + B_k \bar{u}_k, \quad k = 0, 1, \dots, r.$$

Show that there is an optimal periodic policy  $\pi^*$  of the form

$$\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{p-1}^*, \mu_0^*, \mu_1^*, \dots, \mu_{p-1}^*, \dots\},$$

where  $\mu_0^*, \mu_1^*, \dots, \mu_{p-1}^*$  are given by

$$\mu_i^*(x) = -\alpha(\alpha B_i' K_{i+1} B_i + R_i)^{-1} B_i' K_{i+1} A_i x, \quad i = 0, 1, \dots, p-2,$$

$$\mu_{p-1}^*(x) = -\alpha(\alpha B_{p-1}' K_0 B_{p-1} + R_{p-1})^{-1} B_{p-1}' K_0 A_{p-1} x,$$

and the matrices  $K_0, K_1, \dots, K_{p-1}$  satisfy the coupled set of  $p$  algebraic Riccati equations given for  $i = 0, 1, \dots, p-1$  by

$$K_i = A_i' \left( \alpha K_{i+1} + \alpha^2 K_{i+1} B_i (\alpha B_i' K_{i+1} B_i + R_i)^{-1} B_i' K_{i+1} A_i \right) + Q_i,$$

with

$$K_p = K_0.$$

### 3.15 (Linear-Quadratic Problems - Imperfect State Information)

Consider the linear-quadratic problem of Section 3.2 with the difference that the controller, instead of having perfect state information, has access to measurements of the form

$$z_k = Cx_k + v_k, \quad k = 0, 1, \dots$$

As in Section 5.2 of Vol. 1, the disturbances  $v_k$  are independent and have identical statistics, zero mean, and finite covariance matrix. Assume that for every admissible policy  $\pi$  the matrices

$$E\{(x_k - E\{x_k | I_k\})(x_k - E\{x_k | I_k\})' | \pi\}$$

are uniformly bounded over  $k$ , where  $I_k$  is the information vector defined in Section 5.2 of Vol. 1. Show that the stationary policy  $\mu^*$  given by

$$\mu^*(I_k) = -\alpha(\alpha B'KB + R)^{-1}B'KA E\{x_k | I_k\}, \quad \text{for all } I_k, \quad k = 0, 1, \dots$$

is optimal. Show also that the same is true if  $w_k$  and  $v_k$  are nonstationary with zero mean and covariance matrices that are uniformly bounded over  $k$ . Hint: Combine the theory of Sections 5.2 of Vol. 1 and 3.2.

### 3.16 (Policy Iteration for Linear-Quadratic Problems [Kle68])

Consider the problem of Section 3.2 and let  $L_0$  be an  $m \times n$  matrix such that the matrix  $(A + BL_0)$  has eigenvalues strictly within the unit circle.

- (a) Show that the cost corresponding to the stationary policy  $\mu_0$ , where  $\mu_0(x) = L_0x$  is of the form

$$J_{\mu_0}(x) = x'K_0x + \text{constant},$$

where  $K_0$  is a positive semidefinite symmetric matrix satisfying the (linear) equation

$$K_0 = \alpha(A + BL_0)'K_0(A + BL_0) + Q + L_0'RL_0.$$

- (b) Let  $\mu_1(x)$  attain the minimum for each  $x$  in the expression

$$\min_u \{u'Ru + \alpha(Ax + Bu)'K_0(Ax + Bu)\}.$$

Show that for all  $x$  we have

$$J_{\mu_1}(x) = x'K_1x + \text{constant} \leq J_{\mu_0}(x),$$

where  $K_1$  is some positive semidefinite symmetric matrix.

- (c) Show that the policy iteration process described in parts (a) and (b) yields a sequence  $\{K_k\}$  such that

$$K_k \rightarrow K,$$

where  $K$  is the optimal cost matrix of the problem.

### 3.17 (Periodic Inventory Control Problems)

In the inventory control problem of Section 3.3, consider the case where the statistics of the demands  $w_k$ , the prices  $c_k$ , and the holding and the shortage costs are periodic with period  $p$ . Show that there exists an optimal periodic policy of the form  $\pi^* = \{\mu_0^*, \dots, \mu_p^*, \mu_0^*, \dots, \mu_p^*, \dots\}$ ,

$$\mu_i^*(x) = \begin{cases} S_i^* - x & \text{if } x \leq S_i^*, \\ 0 & \text{if otherwise,} \end{cases} \quad i = 0, 1, \dots, p-1,$$

where  $S_0^*, \dots, S_{p-1}^*$  are appropriate scalars.

### 3.18 [HeS84]

Show that the critical level  $S^*$  for the inventory problem with zero fixed cost of Section 3.3 minimizes  $(1-\alpha)cy + L(y)$  over  $y$ . Hint: Show that the cost can be expressed as

$$J_n(x_0) = E \left\{ \sum_{k=0}^{\infty} \alpha^k ((1-\alpha)cy_k + L(y_k)) + \frac{c\alpha}{1-\alpha} E\{w\} - cx_0 \right\},$$

where  $y_k = x_k + \mu_k(x_k)$ .

### 3.19

Consider a machine that may break down and can be repaired. When it operates over a time unit, it costs  $-1$  (that is, it produces a benefit of 1 unit), and it may break down with probability 0.1. When it is in the breakdown mode, it may be repaired with an effort  $u$ . The probability of making it operative over one time unit is then  $u$ , and the cost is  $Cu^2$ . Determine the optimal repair effort over an infinite time horizon with discount factor  $\alpha < 1$ .

### 3.20

Let  $z_0, z_1, \dots$  be a sequence of independent and identically distributed random variables taking values on a finite set  $Z$ . We know that the probability distribution of the  $z_k$ 's is one out of  $n$  distributions  $f_1, \dots, f_n$ , and we are trying to decide which distribution is the correct one. At each time  $k$  after observing  $z_1, \dots, z_k$ , we may either stop the observations and accept one of the  $n$  distributions as correct, or take another observation at a cost  $c > 0$ . The cost for accepting  $f_j$  given that  $f_j$  is correct is  $L_{ij}$ ,  $i, j = 1, \dots, n$ . We assume  $L_{ij} > 0$  for  $i \neq j$ ,  $L_{ii} = 0$ ,  $i = 1, \dots, n$ . The a priori distribution of  $f_1, \dots, f_n$  is denoted

$$P_0 = \{p_0^1, p_0^2, \dots, p_0^n\}, \quad p_0^i \geq 0, \quad \sum_{i=1}^n p_0^i = 1,$$

Show that the optimal cost  $J^*(P_0)$  is a concave function of  $P_0$ . Characterize the optimal acceptance regions and show how they can be obtained in the limit by means of a value iteration method.

### 3.21 (Gambling Strategies for Favorable Games)

A gambler plays a game such as the one of Section 3.5, but where the probability of winning  $p$  satisfies  $1/2 \leq p < 1$ . His objective is to reach a final fortune  $n$ , where  $n$  is an integer with  $n \geq 2$ . His initial fortune is an integer  $i$  with  $0 < i < n$ , and his stake at time  $k$  can take only integer values  $u_k$  satisfying  $0 \leq u_k \leq x_k$ ,  $0 \leq u_k \leq n - x_k$ , where  $x_k$  is his fortune at time  $k$ . Show that the strategy that always stakes one unit is optimal [i.e.,  $\mu^*(x) = 1$  for all integers  $x$  with  $0 < x < n$  is optimal]. Hint: Show that if  $p \in (1/2, 1)$ ,

$$J_{\mu^*}(i) = \left[ \left( \frac{1-p}{p} \right)' + 1 \right] \left[ \left( \frac{1-p}{p} \right)^n - 1 \right]^{-1}, \quad 0 \leq i \leq n,$$

and if  $p = 1/2$ ,

$$J_{\mu^*}(i) = \frac{i}{n}, \quad 0 \leq i \leq n,$$

(or see [Ash70], p. 482, for a proof). Then use the sufficiency condition of Prop. 4.1 in Section 3.4.

### 3.22 [Sch84]

Consider a network of  $n$  queues whereby a customer at queue  $i$  upon completion of service is routed to queue  $j$  with probability  $p_{ij}$ , and exits the network with probability  $1 - \sum_j p_{ij}$ . For each queue  $i$  denote:

$r_i$ : the external customer arrival rate,

$\frac{1}{\mu_i}$ : the average customer service time,

$\lambda_i$ : the customer departure rate,

$a_i$ : the total customer arrival rate (sum of external rate and departure rates from upstream queues weighted by the corresponding probabilities).

We have

$$a_i = r_i + \sum_{j=1}^n \lambda_j p_{ji}, \quad \text{for all } i,$$

and we assume that any portion of the arrival rate  $a_i$  in excess of the service rate  $\mu_i$  is lost; so the departure rate at queue  $i$  satisfies

$$\lambda_i = \min[\mu_i, a_i] = \min \left[ \mu_i, r_i + \sum_{j=1}^n \lambda_j p_{ji} \right].$$

Assume that  $r_i > 0$  for at least one  $i$ , and that for every queue  $i_1$  with  $r_{i_1} > 0$ , there is a queue  $i$  with  $1 - \sum_j p_{ij} > 0$ , and a sequence  $i_1, i_2, \dots, i_k$ ,  $i$  such that  $p_{i_1 i_2} > 0, \dots, p_{i_k i} > 0$ . Show that the departure rates  $\lambda_i$  satisfying the preceding equations are unique and can be found by value iteration or policy iteration. Hint: This problem does not quite fit our framework because we may have  $\sum_j p_{ij} > 1$  for some  $i$ . However, it is possible to carry out an analysis based on  $m$ -stage contraction mappings.

### 3.23 (Infinite Time Reachability [Ber71], [Ber72])

Consider the stationary system

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, 1, \dots,$$

where the disturbance space  $D$  is an arbitrary (not necessarily countable) set. The disturbances  $w_k$  can take values in a subset  $W(x_k, u_k)$  of  $D$  that may depend on  $x_k$  and  $u_k$ . This problem deals with the following question: Given a nonempty subset  $X$  of the state space  $S$ , under what conditions does there exist an admissible policy that keeps the state of the (closed-loop) system

$$x_{k+1} = f(x_k, \mu_k(x_k), w_k) \quad (7.1)$$

in the set  $X$  for all  $k$  and all possible values  $w_k \in W(x_k, \mu_k(x_k))$ , that is,

$$x_k \in X, \quad \text{for all } w_k \in W(x_k, \mu_k(x_k)), \quad k = 0, 1, \dots \quad (7.2)$$

The set  $X$  is said to be *infinitely reachable* if there exists an admissible policy  $\{\mu_0, \mu_1, \dots\}$  and *some* initial state  $x_0 \in X$  for which the above relations are satisfied. It is said to be *strongly reachable* if there exists an admissible policy  $\{\mu_0, \mu_1, \dots\}$  such that for *all* initial states  $x_0 \in X$  the above relations are satisfied.

Consider the function  $R$  mapping any subset  $Z$  of the state space  $S$  into a subset  $R(Z)$  of  $S$  defined by

$$R(Z) = \{x \mid \text{for some } u \in U(x), f(x, u, w) \in Z, \text{ for all } w \in W(x, u)\} \cap Z.$$

(a) Show that the set  $X$  is strongly reachable if and only if  $R(X) = X$ .

(b) Given  $X$ , consider the set  $X^*$  defined as follows:  $x_0 \in X^*$  if and only if  $x_0 \in X$  and there exists an admissible policy  $\{\mu_0, \mu_1, \dots\}$  such that that Eqs. (7.1) and (7.2) are satisfied when  $x_0$  is taken as the initial state of the system. Show that a set  $X$  is infinitely reachable if and only if it contains a nonempty strongly reachable set. Furthermore, the largest such set is  $X^*$  in the sense that  $X^*$  is strongly reachable whenever nonempty, and if  $\tilde{X} \in X$  is another strongly reachable set, then  $\tilde{X} \subseteq X^*$ .

(c) Show that if  $X$  is infinitely reachable, there exists an admissible stationary policy  $\mu$  such that if the initial state  $x_0$  belongs to  $X^*$ , then all subsequent states of the closed-loop system  $x_{k+1} = f(x_k, \mu(x_k), w_k)$  are guaranteed to belong to  $X^*$ .

(d) Given  $X$ , consider the sets  $R^k(X)$ ,  $k = 1, 2, \dots$ , where  $R^k(X)$  denotes the set obtained after  $k$  applications of the mapping  $R$  on  $X$ . Show that

$$X^* \subseteq \bigcap_{k=1}^{\infty} R^k(X).$$

(e) Given  $X$ , consider for each  $x \in X$  and  $k = 1, 2, \dots$  the set

$$U_k(x) = \{u \mid f(x, u, w) \in R^k(X) \text{ for all } w \in W(x, u)\}.$$

Show that, if there exists an index  $\bar{k}$  such that for all  $x \in X$  and  $k \geq \bar{k}$  the set  $U_k(x)$  is a compact subset of a Euclidean space, then  $X^* = \bigcap_{k=1}^{\infty} R^k(X)$ .

### 3.24 (Infinite Time Reachability for Linear Systems)

Consider the linear stationary system

$$x_{k+1} = Ax_k + Bu_k + Cw_k,$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$ , and  $w_k \in \mathbb{R}^r$ , and the matrices  $A$ ,  $B$ , and  $C$  are known and have appropriate dimensions. The matrix  $A$  is assumed invertible. The controls  $u_k$  and the disturbances  $w_k$  are restricted to take values in the ellipsoids  $U = \{u \mid u'Ru \leq 1\}$  and  $W = \{w \mid w'Qw \leq 1\}$ , respectively, where  $R$  and  $Q$  are positive definite symmetric matrices of appropriate dimensions. Show that in order for the ellipsoid  $X = \{x \mid x'Kx \leq 1\}$ , where  $K$  is a positive definite symmetric matrix, to be strongly reachable (in the terminology of Exercise 3.23), it is sufficient that for some positive definite symmetric matrix  $M$  and for some scalar  $\beta \in (0, 1)$  we have

$$K = A' \left[ (1 - \beta)K^{-1} - \frac{1 - \beta}{\beta} GQ^{-1}G' + BR^{-1}B' \right]^{-1} A + M,$$

$$K^{-1} - \frac{1}{\beta} GQ^{-1}G' : \text{positive definite.}$$

Show also that if the above relations are satisfied, the linear stationary policy  $\mu^*$ , where  $\mu^*(x) = Lx$  and

$$L = -(R + B'FB)^{-1}B'FA,$$

$$F = \left[ (1 - \beta)K^{-1} - \frac{1 - \beta}{\beta} GQ^{-1}G' \right]^{-1},$$

achieves reachability of the ellipsoid  $X = \{x \mid x'Kx \leq 1\}$ . Furthermore, the matrix  $(A + BL)$  has all its eigenvalues strictly within the unit circle. (For a proof together with a computational procedure for finding matrices  $K$  satisfying the above, see [Ber71] and [Ber72b].)

### 3.25 (The Blackmailer's Dilemma)

Consider Example 1.1 of Section 2.1. Here, there are two states, state 1 and a termination state  $t$ . At state 1, we can choose a control  $u$  with  $0 < u \leq 1$ ; we then move to state  $t$  at no cost with probability  $p(u)$ , and stay in state 1 at a cost  $-u$  with probability  $1 - p(u)$ .

- (a) Let  $p(u) = u^2$ . For this case it was shown in Example 1.1 of Section 2.1, that the optimal costs are  $J^*(1) = -\infty$  and  $J^*(t) = 0$ . Furthermore, it was shown that there is no optimal stationary policy, although there is an optimal nonstationary policy. Find the set of solutions to Bellman's equation and verify the result of Prop. 1.2(b).
- (b) Let  $p(u) = u$ . Find the set of solutions to Bellman's equation and use Prop. 1.2(b) to show that the optimal costs are  $J^*(1) = -1$  and  $J^*(t) = 0$ . Show that there is no optimal policy (stationary or not).

## Average Cost per Stage Problems

### Contents

4.1. Preliminary Analysis . . . . .	p. 181
4.2. Optimality Conditions . . . . .	p. 191
4.3. Computational Methods . . . . .	p. 202
4.3.1. Value Iteration . . . . .	p. 202
4.3.2. Policy Iteration . . . . .	p. 213
4.3.3. Linear Programming . . . . .	p. 221
4.3.4. Simulation-Based Methods . . . . .	p. 222
4.4. Infinite State Space . . . . .	p. 226
4.5. Notes, Sources, and Exercises . . . . .	p. 229

The results of the preceding chapters apply mainly to problems where the optimal total expected cost is finite either because of discounting or because of a cost-free absorbing state that the system eventually enters. In many situations, however, discounting is inappropriate and there is no natural cost-free absorbing state. In such situations it is often meaningful to optimize the average cost per stage, to be defined shortly. In this chapter, we discuss this type of optimization, with an emphasis on the case of a finite-state Markov chain.

An introductory analysis of the problem of this chapter was given in Section 7.4 of Vol. I. That analysis was based on a connection between the average cost per stage and the stochastic shortest path problem. While this connection can be further extended to obtain more powerful results (see Exercises 4.13-4.16), we develop here an alternative line of analysis that is based on a relation with the discounted cost problem. This relation allows us to use discounted cost results, derived in Sections 1.2 and 1.3, in order to conjecture and prove results for the average cost problem.

## 4.1 PRELIMINARY ANALYSIS

Let us formulate the problem of this chapter for the case of finite state and control spaces. We adopt the Markov chain notation used in Section 1.3. In particular, we denote the states by  $1, \dots, n$ . To each state  $i$  and control  $u$  there corresponds a set of transition probabilities  $p_{ij}(u)$ ,  $j = 1, \dots, n$ . Each time the system is in state  $i$  and control  $u$  is applied, we incur an expected cost  $g(i, u)$ , and the system moves to state  $j$  with probability  $p_{ij}(u)$ . The objective is to minimize over all policies  $\pi = \{\mu_0, \mu_1, \dots\}$  with  $\mu_k(i) \in U(i)$ , for all  $i$  and  $k$ , the average cost per stage  $\dagger$

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \right\},$$

for any given initial state  $x_0$ .

$\dagger$  When the limit defining the average cost is not known to exist, we use instead the definition

$$J_\pi(x_0) = \limsup_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \right\},$$

We will show, however, as part of our subsequent analysis that the limit exists at least for those policies  $\pi$  that are of interest.

As in Section 1.3, we use the following shorthand notation for a stationary policy  $\mu$ :

$$g_\mu = \begin{pmatrix} g(1, \mu(1)) \\ \vdots \\ g(n, \mu(n)) \end{pmatrix}, \quad P_\mu = \begin{pmatrix} p_{11}(\mu(1)) & \dots & p_{1n}(\mu(1)) \\ \vdots & \ddots & \vdots \\ p_{n1}(\mu(n)) & \dots & p_{nn}(\mu(n)) \end{pmatrix}.$$

$$J_\mu = \begin{pmatrix} J_\mu(1) \\ \vdots \\ J_\mu(n) \end{pmatrix}.$$

Since the  $(i, j)$ th element of the matrix  $P_\mu^k$  ( $P_\mu$  to the  $k$ th power) is the  $k$ -step transition probability  $P(x_k = j | x_0 = i)$  corresponding to  $\mu$ , it can be seen that

$$J_\mu = \left( \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P_\mu^k \right) g_\mu.$$

An important result regarding transition probability matrices is that the limit in the preceding equation exists. We show this fact shortly in the context of a more general result, which establishes the connection between the average cost per stage problem and the discounted cost problem.

### An Overview of Results

While the material of this chapter does not rely on the analysis of the average cost problem of Section 7.4 in Vol. I, it is worth summarizing some of the salient features of that analysis (see also Exercises 4.13-4.16). We assumed there that there is a special state, by convention state  $n$ , which is recurrent in the Markov chain corresponding to each stationary policy. If we consider a sequence of generated states, and divide it into cycles marked by successive visits to the special state  $n$ , we see that each of the cycles can be viewed as a state trajectory of a corresponding stochastic shortest path problem with the termination state being essentially  $n$ . More precisely, this stochastic shortest path problem has states  $1, 2, \dots, n$ , plus an artificial termination state  $t$  to which we move from state  $i$  with transition probability  $p_{in}(u)$ . The transition probabilities from a state  $i$  to a state  $j \neq n$  are the same as those of the original problem, while  $p_{in}(u)$  is zero. For any scalar  $\lambda$ , we considered the stochastic shortest path problem with expected stage cost  $g(i, u) - \lambda$  for each state  $i = 1, \dots, n$ . We then argued that if we fix the expected stage cost incurred at state  $i$  to be

$$g(t, u) = \lambda^*,$$

where  $\lambda^*$  is the optimal average cost per stage starting from the special state  $n$ , then the associated stochastic shortest path problem becomes essentially equivalent to the original average cost per stage problem. Furthermore, Bellman's equation for the associated stochastic shortest path

problem can be viewed as Bellman's equation for the original average cost per stage problem. Based on this line of analysis, we showed a number of results, which will be strengthened in the present chapter by using different methods. In summary, these results are the following:

- (a) The optimal average cost per stage is independent of the initial state. This property is a generic feature for almost all average cost problems of practical interest.
- (b) Bellman's equation takes the form

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h^*(j) \right], \quad i = 1, \dots, n,$$

where  $h^*(n) = 0$ ,  $\lambda^*$  is the optimal average cost per stage, and  $h^*(i)$  has the interpretation of a relative or differential cost for each state  $i$  (it is the minimum of the difference between the expected cost to reach  $n$  from  $i$  for the first time and the cost that would be incurred if the cost per stage was the average  $\lambda^*$ ).

- (c) There are versions of the value iteration, policy iteration, adaptive aggregation, and linear programming methods that can be used for computational solution under reasonable conditions.

We will now provide the foundation for the analysis of this chapter by developing the connection between average cost and discounted problems.

### Relation with the Discounted Cost Problem

Let us consider the cost of a stationary policy  $\mu$  for the corresponding  $\alpha$ -discounted problem. It is given by

$$J_{\alpha, \mu} = \sum_{k=0}^{\infty} \alpha^k P_{\mu}^k g_{\mu} = \left( \sum_{k=0}^{\infty} \alpha^k P_{\mu}^k \right) g_{\mu} = (I - \alpha P_{\mu})^{-1} g_{\mu}, \quad \alpha \in (0, 1). \quad (1.1)$$

To get a sense of the relation with the average cost of  $\mu$ , we note that this latter cost is written as

$$\begin{aligned} J_{\mu}(i) &= \lim_{N \rightarrow \infty} \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu(x_k)) \right\} \\ &= \lim_{N \rightarrow \infty} \lim_{\alpha \rightarrow 1} \frac{E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu(x_k)) \right\}}{\sum_{k=0}^{N-1} \alpha^k}. \end{aligned}$$

Assuming that the order of the two limits in the right-hand side above can be interchanged, we obtain

$$\begin{aligned} J_{\mu}(i) &= \lim_{\alpha \rightarrow 1} \lim_{N \rightarrow \infty} \frac{E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu(x_k)) \right\}}{\sum_{k=0}^{N-1} \alpha^k} \\ &= \lim_{\alpha \rightarrow 1} \frac{\lim_{N \rightarrow \infty} E \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu(x_k)) \right\}}{\lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^k} \\ &= \lim_{\alpha \rightarrow 1} (1 - \alpha) J_{\alpha, \mu}(i). \end{aligned}$$

The formal proof of the above relation will follow as a corollary to the next proposition.

**Proposition 1.1:** For any stochastic matrix  $P$  and  $\alpha \in (0, 1)$ , there holds

$$(I - \alpha P)^{-1} = (1 - \alpha)^{-1} P^* + H + O(|1 - \alpha|), \quad (1.2)$$

where  $O(|1 - \alpha|)$  is an  $\alpha$ -dependent matrix such that

$$\lim_{\alpha \rightarrow 1} O(|1 - \alpha|) = 0, \quad (1.3)$$

and the matrices  $P^*$  and  $H$  are given by

$$P^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k, \quad (1.4)$$

$$H = (I - P + P^*)^{-1} - P^*. \quad (1.5)$$

[It will be shown as part of the proof that the limit in Eq. (1.4) and the inverse in Eq. (1.5) exist.] Furthermore,  $P^*$  and  $H$  satisfy the following equations:

$$P^* = PP^* = P^*P = P^*P^*, \quad (1.6)$$

$$P^*H = 0, \quad (1.7)$$

$$P^* + H = I + PH. \quad (1.8)$$

**Proof:** From the matrix inversion formula that expresses each entry of the inverse as a ratio of two determinants, it is seen that the matrix

$$M(\alpha) = (1 - \alpha)(I - \alpha P)^{-1}$$

can be expressed as a matrix with elements that are either zero or fractions whose numerator and denominator are polynomials in  $\alpha$  with no common divisor. The denominator polynomials of the nonzero elements of  $M(\alpha)$  cannot have 1 as a root, since otherwise some elements of  $M(\alpha)$  would tend to infinity as  $\alpha \rightarrow 1$ ; this is not possible, because from Eq. (1.1) for any  $\mu$ , we have  $(1 - \alpha)^{-1}M(\alpha)g_\mu = (I - \alpha P)^{-1}g_\mu = J_{\alpha, \mu}$  and  $|J_{\alpha, \mu}(j)| \leq (1 - \alpha)^{-1} \max_i |g_\mu(i)|$ , implying that the absolute values of the coordinates of  $M(\alpha)g_\mu$  are bounded by  $\max_i |g_\mu(i)|$  for all  $\alpha < 1$ . Therefore, the  $(i, j)$ th element of the matrix  $M(\alpha)$  is of the form

$$m_{ij}(\alpha) = \frac{\gamma(\alpha - \zeta_1) \cdots (\alpha - \zeta_p)}{(\alpha - \xi_1) \cdots (\alpha - \xi_q)}$$

where  $\gamma, \zeta_i, i = 1, \dots, p$ , and  $\xi_i, i = 1, \dots, q$ , are scalars such that  $\zeta_i \neq 1$  for  $i = 1, \dots, q$ .

Define

$$P^* = \lim_{\alpha \rightarrow 1} M(\alpha), \quad (1.9)$$

and let  $H$  be the matrix having as  $(i, j)$ th element the 1st derivative of  $-m_{ij}(\alpha)$  evaluated at  $\alpha = 1$ . By the 1st-order Taylor expansion of the elements of  $m_{ij}(\alpha)$  of  $M(\alpha)$ , we have for all  $\alpha$  in a neighborhood of  $\alpha = 1$

$$M(\alpha) = P^* + (1 - \alpha)H + O((1 - \alpha)^2), \quad (1.10)$$

where  $O((1 - \alpha)^2)$  is an  $\alpha$ -dependent matrix such that

$$\lim_{\alpha \rightarrow 1} \frac{O((1 - \alpha)^2)}{(1 - \alpha)} = 0.$$

Multiplying Eq. (1.10) with  $(1 - \alpha)^{-1}$ , we obtain the desired relation (1.2) [although, we have yet to show that  $P^*$  and  $H$  are also given by Eqs. (1.4) and (1.5), respectively].

We will now show that  $P^*$  as defined by Eq. (1.9), satisfies Eqs. (1.6), (1.5), (1.7), (1.8), and (1.4), in that order.

We have

$$(I - \alpha P)(I - \alpha P)^{-1} = I \quad (1.11)$$

and

$$\alpha(I - \alpha P)(I - \alpha P)^{-1} = \alpha I. \quad (1.12)$$

Subtracting these two equations and rearranging terms, we obtain

$$\alpha P(1 - \alpha)(I - \alpha P)^{-1} = (1 - \alpha)(I - \alpha P)^{-1} + (\alpha - 1)I.$$

By taking the limit as  $\alpha \rightarrow 1$  and using the definition (1.9), it follows that

$$PP^* = P^*,$$

Also, by reversing the order of  $(I - \alpha P)$  and  $(I - \alpha P)^{-1}$  in Eqs. (1.11) and (1.12), it follows similarly that  $P^*P = P^*$ . From  $PP^* = P^*$ , we also obtain  $(I - \alpha P)P^* = (1 - \alpha)P^*$  or  $P^* = (1 - \alpha)(I - \alpha P)^{-1}P^*$ , and by taking the limit as  $\alpha \rightarrow 1$  and by using Eq. (1.9), we have  $P^* = P^*P^*$ . Thus Eq. (1.6) has been proved.

We have, using Eq. (1.6),  $(P - P^*)^2 = P^2 - P^*$ , and similarly

$$(P - P^*)^k = P^k - P^*, \quad k > 0.$$

Therefore,

$$\begin{aligned} (I - \alpha P)^{-1} - (1 - \alpha)^{-1}P^* &= \sum_{k=0}^{\infty} \alpha^k (P^k - P^*) \\ &= I - P^* + \sum_{k=1}^{\infty} \alpha^k (P - P^*)^k \\ &= (I - \alpha(P - P^*))^{-1} - P^*. \end{aligned}$$

On the other hand, from Eq. (1.10), we have

$$\begin{aligned} H &= \lim_{\alpha \rightarrow 1} ((1 - \alpha)^{-1}M(\alpha) - (1 - \alpha)^{-1}P^*) \\ &= \lim_{\alpha \rightarrow 1} ((I - \alpha P)^{-1} - (1 - \alpha)^{-1}P^*). \end{aligned}$$

By combining the last two equations, we obtain

$$H = \lim_{\alpha \rightarrow 1} (I - \alpha(P - P^*))^{-1} - P^* = (I - P + P^*)^{-1} - P^*,$$

which is Eq. (1.5).

From Eq. (1.5), we obtain

$$(I - P + P^*)H = I - (I - P + P^*)P^*$$

or, using Eq. (1.6),

$$H - PH + P^*H = I - P^*. \quad (1.13)$$

Multiplying this relation by  $P^*$  and using Eq. (1.6), we obtain  $P^*H = 0$ , which is Eq. (1.7). Equation (1.8) then follows from Eq. (1.13).

Multiplying Eq. (1.8) with  $P^k$  and using Eq. (1.6), we obtain

$$P^* + P^k H = P^k + P^{k+1}H, \quad k = 0, 1, \dots$$

Adding this relation over  $k = 0, \dots, N - 1$ , we have

$$NP^* + H = \sum_{k=0}^{N-1} P^k + P^N H.$$

Dividing by  $N$  and taking the limit as  $N \rightarrow \infty$ , we obtain Eq. (1.4). **Q.E.D.**

Note that the matrix  $P^*$  of Eq. (1.4) can be used to express concisely the average cost vector  $J$  of any Markov chain with transition probability matrix  $P$  and cost vector  $g$  as

$$J = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k g = \left( \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k \right) g = P^* g.$$

To interpret this equation, note that we may view the  $i$ th row of  $P^*$  as a vector of steady-state occupancy probabilities corresponding to starting at state  $i$ ; that is, the  $ij$ th element  $p_{ij}^*$  of  $P^*$  represents the long-term fraction of time that the Markov chain spends at state  $j$  given that it starts at state  $i$ . Thus the above equation gives the average cost per stage  $J(i)$ , starting from state  $i$ , as the sum  $\sum_{j=1}^n p_{ij}^* g_j$  of all the single-stage costs  $g_j$  weighted by the corresponding occupancy probabilities.

From Eq. (1.4) and Prop. 1.1, we obtain the following relation between  $\alpha$ -discounted and average cost corresponding to a stationary policy.

**Proposition 1.2:** For any stationary policy  $\mu$  and  $\alpha \in (0, 1)$ , we have

$$J_{\alpha, \mu} = (1 - \alpha)^{-1} J_\mu + h_\mu + O(|1 - \alpha|), \quad (1.14)$$

where

$$J_\mu = P_\mu^* g_\mu = \left( \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P_\mu^k \right) g_\mu$$

is the average cost vector corresponding to  $\mu$ , and  $h_\mu$  is a vector satisfying

$$J_\mu + h_\mu = g_\mu + P_\mu h_\mu. \quad (1.15)$$

**Proof:** Equation (1.14) follows from Eqs. (1.1) and (1.2) with the identifications  $P = P_\mu$ ,  $P^* = P_\mu^*$ , and  $h_\mu = Hg_\mu$ . Equation (1.15) follows by multiplying Eq. (1.8) with  $g_\mu$  and by using the same identifications. **Q.E.D.**

In the next section we use the preceding results to establish Bellman's equation for the average cost per stage problem. As in the earlier chapters, this equation involves the mappings  $T$  and  $T_\mu$ , which take the form

$$(TJ)(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) J(j) \right], \quad i = 1, \dots, n, \quad (1.16)$$

$$(T_\mu J)(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) J(j), \quad i = 1, \dots, n. \quad (1.17)$$

## 4.2 OPTIMALITY CONDITIONS

Our first result introduces the analog of Bellman's equation for the case of equal optimal cost for each initial state. This is the case that normally appears in practice, as discussed in Section 7.4 of Vol. I. The proposition shows that all solutions of this equation can be identified with the optimal average cost and an associated differential cost. However, it provides no assurance that the equation has a solution. For this we need further assumptions, which will be given in the sequel (see Prop. 2.6).

**Proposition 2.1:** If a scalar  $\lambda$  and an  $n$ -dimensional vector  $h$  satisfy

$$\lambda + h(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h(j) \right], \quad i = 1, \dots, n, \quad (2.1)$$

or equivalently

$$\lambda c + h = Th, \quad (2.2)$$

then  $\lambda$  is the optimal average cost per stage  $J^*(i)$  for all  $i$ .

$$\lambda = \min_\pi J_\pi(i) = J^*(i), \quad i = 1, \dots, n. \quad (2.3)$$

Furthermore, if  $\mu^*(i)$  attains the minimum in Eq. (2.1) for each  $i$ , the stationary policy  $\mu^*$  is optimal, that is,  $J_{\mu^*}(i) = \lambda$  for all  $i$ .

**Proof:** Let  $\pi = \{\mu_0, \mu_1, \dots\}$  be any admissible policy and let  $N$  be a positive integer. We have, from Eq. (2.2),

$$T_{\mu_{N+1}} h \geq \lambda c + h.$$

By applying  $T_{\mu_{N+2}}$  to both sides of this relation, and by using the monotonicity of  $T_{\mu_{N+2}}$  and Eq. (2.2), we see that

$$T_{\mu_{N+2}} T_{\mu_{N+1}} h \geq T_{\mu_{N+2}} (\lambda c + h) = \lambda c + T_{\mu_{N+2}} h \geq 2\lambda c + h.$$

Continuing in the same manner, we finally obtain

$$T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N+1}} h \geq N\lambda c + h, \quad (2.4)$$

with equality if each  $\mu_k$ ,  $k = 0, 1, \dots, N - 1$ , attains the minimum in Eq. (2.1). As discussed in Section 1.1,  $(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} h)(i)$  is equal to the  $N$ -stage cost corresponding to initial state  $i$ , policy  $\{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , and terminal cost function  $h$ ; that is,

$$(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} h)(i) = E \left\{ h(x_N) + \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \mid x_0 = i, \pi \right\}.$$

Using this relation in Eq. (2.1) and dividing by  $N$ , we obtain for all  $i$

$$\begin{aligned} \frac{1}{N} E \{ h(x_N) \mid x_0 = i, \pi \} &+ \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \mid x_0 = i, \pi \right\} \\ &\geq \lambda + \frac{1}{N} h(i). \end{aligned}$$

By taking the limit as  $N \rightarrow \infty$ , we see that

$$J_\pi(i) \geq \lambda, \quad i = 1, \dots, n,$$

with equality if  $\mu_k(i)$ ,  $k = 0, 1, \dots$ , attains the minimum in Eq. (2.1). Q.E.D.

Note that the proof of Prop. 2.1 carries through even if the state space and control space are infinite as long as the function  $h$  is bounded and the minimum in the optimality equation (2.1) is attained for each  $i$ .

In order to interpret the vector  $h$  in Bellman's equation  $\lambda c + h = Th$ , note that by iterating this equation  $N$  times (see also the proof of the preceding proposition), we obtain  $N\lambda c + h = T^N h$ . Thus for any two states  $i$  and  $j$  we have

$$\lambda + h(i) = (T^N h)(i), \quad \lambda + h(j) = (T^N h)(j),$$

which by subtraction yields

$$h(i) - h(j) = (T^N h)(i) - (T^N h)(j), \quad \text{for all } i, j.$$

For any  $i$ ,  $(T^N h)(i)$  is the optimal  $N$ -stage expected cost starting at  $i$  when the terminal cost function is  $h$ . Thus, according to the preceding equation,  $h(i) - h(j)$  represents, for every  $N$ , the difference in optimal  $N$ -stage expected cost due to starting at state  $i$  rather than starting at state  $j$ . Based on this interpretation, we refer to  $h$  as the *differential* or *relative* cost vector. (An alternative but similar interpretation is given in Section 7.4 of Vol. I.)

Now given a stationary policy  $\mu$ , we may consider, as in Section 1.2, a problem where the constraint set  $U(i)$  is replaced by the set  $\hat{U}(i) = \{\mu(i)\}$ ;

that is,  $\hat{U}(i)$  contains a single element, the control  $\mu(i)$ . Since then we would have only one admissible policy, the policy  $\mu$ , application of Prop. 2.1 yields the following corollary.

**Corollary 2.1.1:** Let  $\mu$  be a stationary policy. If a scalar  $\lambda_\mu$  and an  $n$ -dimensional vector  $h_\mu$  satisfy, for all  $i$ ,

$$\lambda_\mu + h_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{i,j}(\mu(i)) h_\mu(j), \quad (2.5)$$

or equivalently

$$\lambda_\mu c + h_\mu = T_\mu h_\mu,$$

then

$$\lambda_\mu = J_\mu(i), \quad i = 1, \dots, n.$$

### Blackwell Optimal Policies

It turns out that the converse of Prop. 2.1 also holds; that is, if for some scalar  $\lambda$  we have  $J^*(i) = \lambda$  for all  $i = 1, \dots, n$ , then  $\lambda$  together with a vector  $h$  satisfies Bellman's equation (2.1). We show this by introducing the notion of a Blackwell optimal policy, which was first formulated in [Bla62], together with the line of analysis of the present section.

**Definition 1.1:** A stationary policy  $\mu$  is said to be *Blackwell optimal* if it is simultaneously optimal for all the  $\alpha$ -discounted problems with  $\alpha$  in an interval  $(\bar{\alpha}, 1)$ , where  $\bar{\alpha}$  is some scalar with  $0 < \bar{\alpha} < 1$ .

The following proposition provides a useful characterization of Blackwell optimal policies, and essentially shows the converse of Prop. 2.1.

**Proposition 2.2:** The following hold true:

- (a) A Blackwell optimal policy is optimal for the average cost problem within the class of all stationary policies.
- (b) There exists a Blackwell optimal policy.

**Proof:** (a) If  $\mu^*$  is Blackwell optimal, then for all stationary policies  $\mu$

and  $\alpha$  in an interval  $(\bar{\alpha}, 1)$  we have  $J_{\alpha, \mu^*} \leq J_{\alpha, \mu}$ . Equivalently, using Eq. (1.14),

$$(1-\alpha)^{-1}J_{\mu^*} + h_{\mu^*} + O(|1-\alpha|) \leq (1-\alpha)^{-1}J_{\mu} + h_{\mu} + O(|1-\alpha|), \quad \alpha \in (\bar{\alpha}, 1)$$

or

$$J_{\mu^*} \leq J_{\mu} + (1-\alpha)(h_{\mu} - h_{\mu^*}) + (1-\alpha)O(|1-\alpha|), \quad \alpha \in (\bar{\alpha}, 1).$$

By taking the limit as  $\alpha \rightarrow 1$ , we obtain  $J_{\mu^*} \leq J_{\mu}$ .

(b) From Eq. (1.1), we know that, for each  $\mu$  and state  $i$ ,  $J_{\alpha, \mu}(i)$  is a rational function of  $\alpha$ , that is, a ratio of two polynomials in  $\alpha$ . Therefore, for any two policies  $\mu$  and  $\mu'$  the graphs of  $J_{\alpha, \mu}(i)$  and  $J_{\alpha, \mu'}(i)$  either coincide or cross only a finite number of times in the interval  $(0, 1)$ . Since there are only a finite number of policies, we conclude that for each state  $i$  there is a policy  $\mu^*$  and a scalar  $\bar{\alpha}_i \in (0, 1)$  such that  $\mu^*$  is optimal for the  $\alpha$ -discounted problem for  $\alpha \in (\bar{\alpha}_i, 1)$  when the initial state is  $i$ . Consider the stationary policy defined for each  $i$  by  $\mu^*(i) = \mu^*(i)$ . Then  $\mu^*(i)$  attains the minimum in Bellman's equation for the  $\alpha$ -discounted problem

$$J_{\alpha}(i) = \min_{u \in U(i)} \left[ g(i, u) + \alpha \sum_{j=1}^n p_{ij}(u) J_{\alpha}(j) \right]$$

for all  $i$  and for all  $\alpha$  in the interval  $(\max_i \bar{\alpha}_i, 1)$ . Therefore,  $\mu^*$  is a stationary optimal policy for the  $\alpha$ -discounted problem for all  $\alpha$  in  $(\max_i \bar{\alpha}_i, 1)$ , implying that  $\mu^*$  is Blackwell optimal. Q.E.D.

We note that the converse of Prop. 2.2(a) is not true; it is possible that a stationary average cost optimal policy is not Blackwell optimal (see Exercise 4.6). We mention also that one can show a stronger result than Prop. 2.2(b), namely that a Blackwell optimal policy is average cost optimal within the class of all policies (not just those that are stationary; see Exercise 4.7).

The next proposition provides a useful characterization of Blackwell optimal policies.

**Proposition 2.3:** If  $\mu^*$  is Blackwell optimal, then for all stationary policies  $\mu$  we have

$$J_{\mu^*} = P_{\mu^*} J_{\mu^*} \leq P_{\mu} J_{\mu^*}. \quad (2.6)$$

Furthermore, for all  $\mu$  such that  $P_{\mu} J_{\mu^*} = P_{\mu} J_{\mu^*}$ , we have

$$J_{\mu^*} + h_{\mu^*} = g_{\mu^*} + P_{\mu} h_{\mu^*} \leq g_{\mu} + P_{\mu} h_{\mu^*}, \quad (2.7)$$

where  $h_{\mu^*}$  is a vector corresponding to  $\mu^*$  as in Prop. 1.2.

**Proof:** Since  $\mu^*$  is optimal for the  $\alpha$ -discounted problem for all  $\alpha$  in an interval  $(\bar{\alpha}, 1)$ , we must have, for every  $\mu$  and  $\alpha \in (\bar{\alpha}, 1)$ ,

$$g_{\mu^*} + \alpha P_{\mu^*} J_{\alpha, \mu^*} \leq g_{\mu} + \alpha P_{\mu} J_{\alpha, \mu^*}. \quad (2.8)$$

From Prop. 1.2, we have, for all  $\alpha \in (\bar{\alpha}, 1)$ ,

$$J_{\alpha, \mu^*} = (1-\alpha)^{-1}J_{\mu^*} + h_{\mu^*} + O(|1-\alpha|).$$

Substituting this expression in Eq. (2.8), we obtain

$$0 \leq g_{\mu} - g_{\mu^*} + \alpha(P_{\mu} - P_{\mu^*})((1-\alpha)^{-1}J_{\mu^*} + h_{\mu^*} + O(|1-\alpha|)), \quad (2.9)$$

or equivalently

$$0 \leq (1-\alpha)(g_{\mu} - g_{\mu^*}) + \alpha(P_{\mu} - P_{\mu^*})(J_{\mu^*} + (1-\alpha)h_{\mu^*} + O((1-\alpha)^2)).$$

By taking the limit as  $\alpha \rightarrow 1$ , we obtain the desired relation  $P_{\mu^*} J_{\mu^*} \leq P_{\mu} J_{\mu^*}$ .

If  $\mu$  is such that  $P_{\mu^*} J_{\mu^*} = P_{\mu} J_{\mu^*}$ , then from Eq. (2.9) we obtain

$$0 \leq g_{\mu} - g_{\mu^*} + \alpha(P_{\mu} - P_{\mu^*})(h_{\mu^*} + O(|1-\alpha|)).$$

By taking the limit as  $\alpha \rightarrow 1$  and by using also the relation  $J_{\mu^*} + h_{\mu^*} = g_{\mu^*} + P_{\mu^*} h_{\mu^*}$  [cf. Eq. (1.15)], we obtain the desired relation (2.7). Q.E.D.

As a consequence of the preceding proposition, we obtain a converse of Prop. 2.1.

**Proposition 2.4:** If the optimal average cost over the class of stationary policies is equal to  $\lambda$  for all initial states, then there exists a vector  $h$  such that

$$\lambda + h(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h(j) \right], \quad i = 1, \dots, n. \quad (2.10)$$

or equivalently

$$\lambda e + h = Th.$$

**Proof:** Let  $\mu^*$  be a Blackwell optimal policy. We then have  $J_{\mu^*}(i) = \lambda$  for all  $i$ . For every  $\mu$ , each element of the vector  $P_{\mu} J_{\mu^*}$  is equal to  $\lambda$ , so that  $P_{\mu^*} J_{\mu^*} = P_{\mu} J_{\mu^*}$ . From Eq. (2.7), we then obtain the desired relation (2.10) with  $h = h_{\mu^*}$ . Q.E.D.

### Bellman's Equation for a Unichain Policy

We recall from Appendix D of Vol. I that in a finite-state Markov chain, a recurrent class is a set of states that communicate in the sense that from every state of the set, there is a probability of 1 to eventually go to all other states of the set and a probability of 0 to ever go to any state outside the set. There are two kinds of states: those that belong to some recurrent class (these are the states that after they are visited once, they will be visited an infinite number of times with probability 1), and those that are transient (these are the states that with probability 1 will be visited only a finite number of times regardless of the initial state).

Stationary policies whose associated Markov chains have a single recurrent class and a possibly empty set of transient states will play an important role in our development. Such policies are called *unichain*. The state trajectory of the Markov chain corresponding to a unichain policy, is eventually (with probability 1) confined to the recurrent class of states, so the average cost per stage corresponding to all initial states as well as the differential costs of the recurrent states are independent of the stage costs of the transient states. The next proposition shows that for a unichain policy  $\mu$ , the average cost per stage is the same for all initial states, and that Bellman's equation  $\lambda_\mu c + h_\mu = T_\mu h_\mu$  holds. Furthermore, we show that Bellman's equation has a unique solution, provided we fix the differential cost of some state at some arbitrary value (0, for example). This is necessary, since if  $\lambda_\mu$  and  $h_\mu$  satisfy Bellman's equation (2.5), the same is true for  $\lambda_\mu + \gamma c$ , where  $\gamma$  is any scalar.

**Proposition 2.5:** Let  $\mu$  be a unichain policy. Then:

- (a) There exists a constant  $\lambda_\mu$  and a vector  $h_\mu$  such that

$$J_\mu(i) = \lambda_\mu, \quad i = 1, \dots, n, \quad (2.11)$$

and

$$\lambda_\mu + h_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h_\mu(j), \quad i = 1, \dots, n. \quad (2.12)$$

- (b) Let  $t$  be a fixed state. The system of the  $n+1$  linear equations

$$\lambda + h(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h(j), \quad i = 1, \dots, n, \quad (2.13)$$

$$h(t) = 0, \quad (2.14)$$

in the  $n+1$  unknowns  $\lambda, h(1), \dots, h(n)$  has a unique solution.

**Proof:** (a) Let  $t$  be a recurrent state under  $\mu$ . For each state  $i \neq t$  let  $C_i$  and  $N_i$  be the expected cost and the expected number of stages, respectively, to reach  $t$  for the first time starting from  $i$  under policy  $\mu$ . Let also  $C_t$  and  $N_t$  be the expected cost and expected number of stages, respectively, to return to  $t$  for the first time starting from  $t$  under policy  $\mu$ . From Prop. 1.1 in Section 2.1, we have that  $C_i$  and  $N_i$  solve uniquely the systems of equations

$$C_i = g(i, \mu(i)) + \sum_{j=1, j \neq t}^n p_{ij}(\mu(i))C_j, \quad i = 1, \dots, n, \quad (2.15)$$

$$N_i = 1 + \sum_{j=1, j \neq t}^n p_{ij}(\mu(i))N_j, \quad i = 1, \dots, n. \quad (2.16)$$

Let

$$\lambda_\mu = \frac{C_t}{N_t}. \quad (2.17)$$

Multiplying Eq. (2.16) by  $\lambda_\mu$  and subtracting it from Eq. (2.15), we obtain

$$C_i - \lambda_\mu N_i = g(i, \mu(i)) - \lambda_\mu + \sum_{j=1, j \neq t}^n p_{ij}(\mu(i))(C_j - \lambda_\mu N_j), \quad i = 1, \dots, n.$$

By defining

$$h_\mu(i) = C_i - \lambda_\mu N_i, \quad i = 1, \dots, n, \quad (2.18)$$

and by noting that from Eq. (2.17), we have

$$h_\mu(t) = 0,$$

we obtain

$$\lambda_\mu + h_\mu(i) = g(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i))h_\mu(j), \quad i = 1, \dots, n,$$

which is Eq. (2.12). Equation (2.11) follows from Eq. (2.12) and Cor. 2.1.1.

(b) By part (a), for any solution  $(\lambda, h)$  of the system of equations (2.13) and (2.14), we have  $\lambda = \lambda_\mu$ , as well as  $h(t) = 0$ . Suppose that  $t$  belongs to

the recurrent class of states of the Markov chain corresponding to  $\mu$ . Then, in view of Eq. (2.14), the system of equations (2.13) can be written as

$$h(i) = g(i, \mu(i)) - \lambda_\mu + \sum_{j=1, j \neq i}^n p_{ij}(\mu(i))h(j), \quad i = 1, \dots, n, i \neq t,$$

and is the same as Bellman's equation for a corresponding stochastic shortest path problem where  $t$  is the termination state,  $g(i, \mu(i)) - \lambda_\mu$  is the expected stage cost at state  $i$ , and  $h(i)$  is the average cost starting from  $i$  up to reaching  $t$ . By Prop. 1.2 in Section 2.1, this system has a unique solution, so  $h(i)$  is uniquely defined by Eq. (2.13) for all  $i \neq t$ .

Suppose now that  $t$  is a transient state of the Markov chain corresponding to  $\mu$ . Then we choose another state  $\bar{t}$  that belongs to the recurrent class and make the transformation of variables  $\bar{h}(i) = h(i) - h(\bar{t})$ . The system of equations (2.13) and (2.14) can be written in terms of the variables  $\lambda$  and  $\bar{h}(i)$  as

$$\bar{h}(i) = g(i, \mu(i)) - \lambda + \sum_{j=1, j \neq \bar{t}}^n p_{ij}(\mu(i))\bar{h}(j), \quad i = 1, \dots, n, i \neq \bar{t},$$

$$\bar{h}(\bar{t}) = 0,$$

so by the stochastic shortest path argument given earlier, it has a unique solution, implying that the solution of the system of equations (2.13) and (2.14) is also unique. Q.E.D.

### Conditions for Equal Optimal Cost for All Initial States

We now turn to the case of multiple policies, and we provide conditions under which Bellman's equation  $\lambda e + h = Th$  has a solution, and by Prop. 2.1, the optimal cost is independent of the initial state.

**Proposition 2.6:** Assume any one of the following three conditions:

- (1) Every policy that is optimal within the class of stationary policies is unichain.
- (2) For every two states  $i$  and  $j$ , there exists a stationary policy  $\pi$  (depending on  $i$  and  $j$ ) such that, for some  $k$ ,

$$P(x_k = j \mid x_0 = i, \pi) > 0.$$

- (3) There exists a state  $t$ , and constants  $L > 0$  and  $\bar{\alpha} \in (0, 1)$  such that

$$|J_\alpha(i) - J_\alpha(t)| \leq L, \quad \text{for all } i = 1, \dots, n, \text{ and } \alpha \in (\bar{\alpha}, 1), \quad (2.19)$$

where  $J_\alpha$  is the  $\alpha$ -discounted optimal cost vector.

Then the optimal average cost per stage has the same value  $\lambda$  for all initial states  $i$ . Furthermore,  $\lambda$  satisfies

$$\lambda = \lim_{\alpha \rightarrow 1} (1 - \alpha)J_\alpha(i), \quad i = 1, \dots, n, \quad (2.20)$$

and for any state  $t$ , the vector  $h$  given by

$$h(i) = \lim_{\alpha \rightarrow 1} (J_\alpha(i) - J_\alpha(t)), \quad i = 1, \dots, n, \quad (2.21)$$

satisfies Bellman's equation

$$\lambda e + h = Th, \quad (2.22)$$

together with  $\lambda$ .

**Proof:** Assume condition (1). Proposition 2.2 asserts that a Blackwell optimal policy exists and is optimal within the class of stationary policies. Therefore, by condition (1), this policy is unichain, and by Prop. 2.5, the corresponding average cost is independent of the initial state. The result follows from Prop. 2.4.

Assume condition (2). Consider a Blackwell optimal policy  $\mu^*$ . If it yields average cost that is independent of the initial state, we are done, as earlier. Assume the contrary; that is, both the set

$$M = \left\{ i \mid J_{\mu^*}(i) = \max_j J_{\mu^*}(j) \right\}$$

and its complement  $\bar{M}$  are nonempty. The idea now is to use the hypothesis that every pair of states communicates under some stationary policy, in order to show that the average cost of states in  $M$  can be reduced by opening communication to the states in  $\bar{M}$ , thereby creating a contradiction. Take any states  $i \in M$  and  $j \in \bar{M}$ , and a stationary policy  $\mu$  such that, for some  $k$ ,  $P(x_k = j \mid x_0 = i, \mu) > 0$ . Then there must exist states  $m \in M$  and  $\bar{m} \in \bar{M}$  such that there is a positive transition probability from  $m$  to  $\bar{m}$  under  $\mu$ ; that is,  $[P_\mu]_{m\bar{m}} = P(x_{k+1} = \bar{m} \mid x_k = m, \mu) > 0$ . It can thus be seen that the  $m$ th component of  $P_\mu J_{\mu^*}$  is strictly less than

$\max_i J_{\mu^*}(i)$ , which is equal to the  $m$ th component of  $J_{\mu^*}$ . This contradicts the necessary condition (2.6).

Finally, assume condition (3). Let  $\mu^*$  be a Blackwell optimal policy. By Eq. (1.14), we have for all states  $i$  and  $\alpha$  in some interval  $(\bar{\alpha}, 1)$

$$J_\alpha(i) = (1 - \alpha)^{-1} J_{\mu^*}(i) + h_{\mu^*}(i) + O(|1 - \alpha|). \quad (2.23)$$

Writing this equation for state  $i$  and for state  $t$ , and subtracting, we obtain for all  $i \neq t$ ,

$$|J_{\mu^*}(i) - J_{\mu^*}(t)| \leq (1 - \alpha) |J_\alpha(i) - J_\alpha(t)| + (1 - \alpha) |h_{\mu^*}(i) - h_{\mu^*}(t)| + O((1 - \alpha)^2).$$

Taking the limit as  $\alpha \rightarrow 1$  and using the hypothesis that  $|J_\alpha(i) - J_\alpha(t)| \leq L$  for all  $\alpha \in (0, 1)$ , we obtain that  $J_{\mu^*}(i) = J_{\mu^*}(t)$  for all  $i$ . Thus the average cost of the Blackwell optimal policy is independent of the initial state, and we are done.

To show Eqs. (2.20)-(2.22), we note that the relation  $\lim_{\alpha \rightarrow 1} (1 - \alpha) J_\alpha(i) = \lambda$  for all  $i$  follows from Eq. (2.23) and the fact  $J_{\mu^*}(i) = \lambda$  for all  $i$ . Also, from Eq. (2.23), we have

$$J_\alpha(i) - J_\alpha(t) = h_{\mu^*}(i) - h_{\mu^*}(t) + O(|1 - \alpha|),$$

so that

$$\lim_{\alpha \rightarrow 1} (J_\alpha(i) - J_\alpha(t)) = h_{\mu^*}(i) - h_{\mu^*}(t).$$

Setting  $h(i) = h_{\mu^*}(i) - h_{\mu^*}(t)$  for all  $i$ , and using the fact  $J_\mu(i) = \lambda$  for all  $i$  and Eq. (2.7), we see that the condition  $\lambda c + h = Th$  is satisfied. Q.E.D.

The conditions of the preceding proposition are among the weakest guaranteeing that the optimal average cost per stage is independent of the initial state. In particular, it is clear that some sort of accessibility condition must be satisfied by the transition probability matrices corresponding to stationary policies or at least to optimal stationary policies. For if there existed two states neither of which could be reached from the other no matter which policy we use, then it can be only by accident that the same optimal cost per stage will correspond to each one. An extreme example is a problem where the state is forced to stay the same regardless of the control applied (i.e., each state is absorbing). Then the optimal average cost per stage for each state  $i$  is  $\min_{u \in U(i)} g(i, u)$ , and this cost may be different for different states.

### Example 2.1: (Machine Replacement)

Consider a machine that can be in any one of  $n$  states,  $1, 2, \dots, n$ . There is a cost  $g(i)$  for operating for one time period the machine when it is in state  $i$ . The options at the start of each period are to (a) let the machine

operate one more period in the state it currently is, or (b) repair the machine at a positive cost  $B$  and bring it to state 1 (corresponding to a machine in perfect condition). The transitions between different states over each time period are governed by given probabilities  $p_{ij}$ . Once repaired, the machine is guaranteed to stay in state 1 for one period, and in subsequent periods, it may deteriorate to states  $j \geq 1$  according to the transition probabilities  $p_{1j}$ . The problem is to find a policy that minimizes the average cost per stage. Note that we have analyzed the discounted cost version of this problem in Example 2.1 of Section 1.2. As in that example, we will assume that  $g(i)$  is nondecreasing in  $i$ , and that the transition probabilities satisfy

$$\sum_{j=1}^n p_{ij} J(j) \leq \sum_{j=1}^n p_{i,i+1,j} J(j), \quad i = 1, \dots, n-1,$$

for all functions  $J(i)$ , which are monotonically nondecreasing in  $i$ .

Note that not all policies are unichain here. For example, consider the stationary policy that replaces at every state except the worst state  $n$  (a poor but legitimate choice). The corresponding Markov chain has two recurrent classes,  $\{1, 2, \dots, n-1\}$  and  $\{n\}$  (assuming that  $p_{1n} = 0$ ). It can also be seen that condition (2) of Prop. 2.6 is not guaranteed in the absence of further assumptions. [This condition is satisfied if we assume in addition that for all  $i$  we have  $p_{(i+1)} > 0$ , because, by replacing, we can bring the system to state 1, from where, by not replacing, we can reach every other state.]

We can show, however, that condition (3) of Prop. 2.6 is satisfied. Indeed, consider the corresponding discounted problem with a discount factor  $\alpha < 1$ . We have for all  $i$

$$J_\alpha(i) = \min \left[ R + g(1) + \alpha J_\alpha(1), g(i) + \alpha \sum_{j=1}^n p_{ij} J_\alpha(j) \right],$$

and in particular,

$$\begin{aligned} J_\alpha(1) &\leq R + g(1) + \alpha J_\alpha(1), \\ J_\alpha(1) &= \min \left[ R + g(1) + \alpha J_\alpha(1), g(1) + \alpha \sum_{j=1}^n p_{1j} J_\alpha(j) \right]. \end{aligned}$$

From the last two equations, by subtraction we obtain

$$J_\alpha(i) - J_\alpha(1) \leq \max \left[ 0, R + \alpha \left( J_\alpha(1) - \sum_{j=1}^n p_{1j} J_\alpha(j) \right) \right] \leq R,$$

where the last inequality follows from the fact

$$0 \leq J_\alpha(i) - J_\alpha(1), \quad i = 1, \dots, n,$$

which holds since  $J_\alpha(i) - J_\alpha(1)$  is nondecreasing in  $i$ , as shown in Example 2.1 of Section 1.2. The last two relations imply that condition (3) of Prop.

2.6 is satisfied, and it follows that there exists a scalar  $\lambda$  and a vector  $h$ , such that for all  $i$ ,

$$\lambda + h(i) = \min \left[ R + g(1) + h(1), g(i) + \sum_{j=1}^n p_{ij}h(j) \right],$$

while the policy that chooses the minimizing action above is average cost optimal.

By Prop. 2.6, we can take  $h(i) = \lim_{n \rightarrow \infty} (J_n(i) - J_n(1))$ , and since  $J_n(i) - J_n(1)$  is nondecreasing in  $i$ , it follows that  $h(i)$  is also nondecreasing in  $i$ . Similar to Example 2.1 of Section 1.2, this implies that an optimal policy takes the form

replace if and only if  $i \geq i^*$ ,

where

$$i^* = \begin{cases} \text{smallest state in } S_R & \text{if } S_R \text{ is nonempty,} \\ n+1 & \text{otherwise,} \end{cases}$$

and

$$S_R = \left\{ i \mid R + g(1) + h(1) \leq g(i) + \sum_{j=1}^n p_{ij}h(j) \right\},$$

### 4.3 COMPUTATIONAL METHODS

All the computational methods developed for discounted and stochastic shortest path problems (cf. Sections 1.3 and 2.2) have average cost per stage counterparts, which we discuss in this section. However, the derivations of these methods are often intricate, and have no direct analogs in the discounted and stochastic shortest path context. In fact, the validity of these methods may depend on assumptions that relate to the structure of the underlying Markov chains, something that we have not encountered so far.

#### 4.3.1 Value Iteration

The natural version of the value iteration method for the average cost problem is simply to generate successively the finite horizon optimal costs  $T^k J_0$ ,  $k = 1, 2, \dots$ , starting with the zero function  $J_0$ . It is then natural to speculate that the  $k$ -stage average costs  $T^k J_0/k$  converge to the optimal average cost vector as  $k \rightarrow \infty$  (this is in fact proved under natural conditions in Section 7.4 of Vol. I). This method has two drawbacks. First, some of the components of  $T^k J_0$  typically diverge to  $\infty$  or  $-\infty$ , so

direct calculation of  $\lim_{k \rightarrow \infty} T^k J_0/k$  is numerically impractical. Second, this method will not provide us with a corresponding differential cost vector  $h$ .

We can bypass both difficulties by subtracting a multiple of the unit vector  $e$  from  $T^k J_0$ , so that the difference, call it  $h^k$ , remains bounded. In particular, we consider methods of the form

$$h^k = T^k J_0 - \delta^k e, \quad (3.1)$$

where  $\delta^k$  is some scalar satisfying

$$\min_{i=1,\dots,n} (T^k J_0)(i) \leq \delta^k \leq \max_{i=1,\dots,n} (T^k J_0)(i),$$

such as for example the average of  $(T^k J_0)(i)$

$$\delta^k = \frac{1}{n} \sum_{i=1}^n (T^k J_0)(i),$$

or

$$\delta^k = (T^k J_0)(t),$$

where  $t$  is some fixed state. Then if the differences  $\max_i (T^k J_0)(i) - \min_i (T^k J_0)(i)$  remain bounded as  $k \rightarrow \infty$  (this can be guaranteed under the assumptions of the subsequent Prop. 3.1), the vectors  $h^k$  also remain bounded, and we will see that with a proper choice of the scalar  $\delta^k$ , the vectors  $h^k$  converge to a differential cost vector.

Let us now restate the algorithm  $h^k = T^k J_0 - \delta^k e$  in a form that is suitable for iterative calculation. We have

$$h^{k+1} = T^{k+1} J_0 - \delta^{k+1} e,$$

and since

$$T^{k+1} J_0 = T(T^k J_0) = T(h^k + \delta^k e) = Th^k + \delta^k e,$$

we obtain

$$h^{k+1} = Th^k + (\delta^k - \delta^{k+1})e. \quad (3.2)$$

In the case where  $\delta^k$  is given by the average of  $(T^k J_0)(i)$ , we have

$$\delta^{k+1} = \frac{1}{n} \sum_{i=1}^n (T^{k+1} J_0)(i) = \frac{1}{n} \sum_{i=1}^n (T(h^k + \delta^k e))(i) = \frac{1}{n} \sum_{i=1}^n (Th^k)(i) + \delta^k,$$

so that the iteration (3.2) is written as

$$h^{k+1} = Th^k - \frac{1}{n} \sum_{i=1}^n (Th^k)(i)e. \quad (3.3)$$

Similarly, in the case where we fix a state  $t$  and we choose  $\delta^k = (T^k J_0)(t)$ , we have

$$\delta^{k+1} = (T^{k+1} J_0)(t) = (T(h^k + \delta^k e))(t) = (Th^k)(t) + \delta^k,$$

and the iteration (3.2) is written as

$$h^{k+1} = Th^k - (Th^k)(t)e. \quad (3.4)$$

We will henceforth restrict attention to the case where  $\delta^k = (T^k J_0)(t)$ , and we will call the corresponding algorithm (3.4) *relative value iteration*, since the iterate  $h^k$  is equal to  $T^k J_0 - (T^k J_0)(t)e$  and may be viewed as a  $k$ -stage optimal cost vector *relative to state t*. The following results also apply to other versions of the algorithm (see Exercises 4.4 and 4.5). Note that relative value iteration, which generates  $h^k$ , is not really different than ordinary value iteration, which generates  $T^k J_0$ . The vectors generated by the two methods merely differ by a multiple of the unit vector, and the minimization problems involved in the corresponding iterations of the two methods are mathematically equivalent.

It can be seen that if the relative value iteration (3.4) converges to some vector  $h$ , then

$$(Th)(t)e + h = Th,$$

which by Prop. 2.1, implies that  $(Th)(t)$  is the optimal average cost per stage for all initial states, and  $h$  is an associated differential cost vector. Thus convergence can only be expected when the optimal average cost per stage is independent of the initial state, indicating that at least one of the conditions of Prop. 2.6 is required. However, it turns out that a stronger hypothesis is needed for convergence. The following example illustrates the reason.

### Example 3.1:

Consider the iteration

$$h^{k+1} = T_\mu h^k - (T_\mu h^k)(t)e,$$

which is the relative value iteration (3.4) for the case of a fixed  $\mu$ . Using the expressions  $T_\mu h^k = g_\mu + P_\mu h^k$  and  $(T_\mu h^k)(t) = e'_t(g_\mu + P_\mu h^k)$ , where  $e'_t$  is the row vector having all coordinates equal to 0 except for coordinate  $t$  which is equal to 1, this iteration can be written as

$$h^{k+1} = g_\mu + P_\mu h^k - cc'_t(g_\mu + P_\mu h^k).$$

Equivalently, we have

$$h^{k+1} = (I - cc'_t)g_\mu + \hat{P}_\mu h^k, \quad (3.5)$$

where

$$\hat{P}_\mu = (I - cc'_t)P_\mu. \quad (3.6)$$

Convergence of iteration (3.5) depends on whether all the eigenvalues of  $\hat{P}_\mu$  lie strictly within the unit circle. We have for any eigenvalue  $\gamma$  of  $P_\mu$  with corresponding eigenvector  $v$ ,

$$P_\mu v = (I - cc'_t)P_\mu v = \gamma(v - cc'_t v),$$

and in particular, for the eigenvalue  $\gamma = 1$  and the corresponding eigenvector  $v = e$  we obtain using the fact  $e'_t e = 1$ ,

$$\hat{P}_\mu e = 0.$$

Therefore, we have

$$\hat{P}_\mu(e - cc'_t e) = \gamma(e - cc'_t e),$$

and it follows that each eigenvalue  $\gamma$  of  $P_\mu$  with corresponding eigenvector  $v$ , which is not a scalar multiple of  $e$ , is also an eigenvalue of  $\hat{P}_\mu$  with corresponding eigenvector  $(e - cc'_t e)$ . Thus, if  $P_\mu$  has an eigenvalue  $\gamma \neq 1$  that is on the unit circle, the iteration (3.5) is not convergent. This occurs when  $P_\mu$  has a periodic structure and some of its nonunit eigenvalues are on the unit circle. For example, suppose that

$$P_\mu = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

which has eigenvalues 1 and -1. Then taking  $t = 1$ , the matrix  $\hat{P}_\mu$  of Eq. (3.6) is given by

$$\hat{P}_\mu = \left( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}(1 - 0) \right) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix},$$

and has eigenvalues 0 and -1. As a result, iteration (3.5) does not converge even though  $\mu$  is a unichain policy.

The following proposition shows convergence of the relative value iteration (3.4) under a technical condition that excludes situations such as the one of the preceding example. When there is only one control available per state, that is, there is only one stationary policy  $\mu$ , the condition of the following proposition requires that for some positive integer  $m$ , the matrix  $P_\mu^m$  has at least one column all the components of which are positive. As can be seen from the preceding example, this condition need not hold if  $\mu$  is unichain. However, we will later provide a variant of the relative value iteration (3.4) that converges under the weaker condition that all stationary policies are unichain (see Prop. 3.3).

**Proposition 3.1:** Assume that there exists a positive integer  $m$  such that for every admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$ , there exists an  $\epsilon > 0$  and a state  $s$  such that

$$[P_{\mu_m} P_{\mu_{m-1}} \dots P_{\mu_1}]_{is} \geq \epsilon, \quad i = 1, \dots, n, \quad (3.7)$$

$$[P_{\mu_{m-1}} P_{\mu_{m-2}} \dots P_{\mu_0}]_{is} \geq \epsilon, \quad i = 1, \dots, n, \quad (3.8)$$

where  $[\cdot]_{is}$  denotes the element of the  $i$ th row and  $s$ th column of the corresponding matrix. Fix a state  $t$  and consider the relative value iteration algorithm

$$h^{k+1}(i) = (Th^k)(i) - (Th^k)(t), \quad i = 1, \dots, n, \quad (3.9)$$

where  $h^0(i)$  are arbitrary scalars. Then the sequence  $\{h^k\}$  converges to a vector  $h$  satisfying  $(Th)(t)c + h = Th$ , so that by Prop. 2.1,  $(Th)(t)$  is equal to the optimal average cost per stage for all initial states and  $h$  is an associated differential cost vector.

**Proof:** Denote

$$q^k(i) = h^{k+1}(i) - h^k(i), \quad i = 1, 2, \dots, n.$$

We will show that for all  $i$  and  $k \geq m$  we have

$$\max_i q^k(i) - \min_i q^k(i) \leq (1 - \epsilon) \left( \max_i q^{k-m}(i) - \min_i q^{k-m}(i) \right), \quad (3.10)$$

where  $m$  and  $c$  are as stated in the hypothesis. From this relation we then obtain, for some  $B > 0$  and all  $k$ ,

$$\max_i q^k(i) - \min_i q^k(i) \leq B(1 - \epsilon)^{k/m}.$$

Since  $q^k(t) = 0$ , it follows that, for all  $i$ ,

$$|h^{k+1}(i) - h^k(i)| = |q^k(i)| \leq \max_j q^k(j) - \min_j q^k(j) \leq B(1 - \epsilon)^{k/m}.$$

Therefore, for every  $r > 1$  and  $i$  we have

$$\begin{aligned} |h^{k+r}(i) - h^k(i)| &\leq \sum_{l=0}^{r-1} |h^{k+l+1}(i) - h^{k+l}(i)| \\ &\leq B(1 - \epsilon)^{k/m} \sum_{l=0}^{r-1} (1 - \epsilon)^{l/m} \\ &= \frac{B(1 - \epsilon)^{k/m} (1 - (1 - \epsilon)^{r/m})}{1 - (1 - \epsilon)^{1/m}}, \end{aligned} \quad (3.11)$$

so that  $\{h^k(i)\}$  is a Cauchy sequence and converges to a limit  $h(i)$ . From Eq. (3.9) we see then that the equation  $(Th)(t) + h(i) = (Th)(i)$  holds for all  $i$ . It will thus be sufficient to prove Eq. (3.10).

To prove Eq. (3.10), we denote by  $\mu_k(i)$  the control that attains the minimum in the relation

$$(Th^k)(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u) h^k(j) \right], \quad (3.12)$$

for every  $k$  and  $i$ . Denote

$$\lambda_k = (Th^k)(t).$$

Then we have

$$h^{k+1} = g_{\mu_k} + P_{\mu_k} h^k - \lambda_k c \leq g_{\mu_{k+1}} + P_{\mu_{k+1}} h^k - \lambda_k c,$$

$$h^k = g_{\mu_{k-1}} + P_{\mu_{k-1}} h^{k-1} - \lambda_{k-1} c \leq g_{\mu_k} + P_{\mu_k} h^{k-1} - \lambda_{k-1} c,$$

where  $c = (1, \dots, 1)^t$  is the unit vector. From these relations, using the definition  $q^k = h^{k+1} - h^k$ , we obtain

$$P_{\mu_k} q^{k-1} + (\lambda_{k-1} - \lambda_k) c \leq q^k \leq P_{\mu_{k-1}} q^{k-1} + (\lambda_{k-1} - \lambda_k) c.$$

Since this relation holds for every  $k \geq 1$ , by iterating we obtain

$$\begin{aligned} P_{\mu_k} \dots P_{\mu_{k-m+1}} q^{k-m} + (\lambda_{k-m} - \lambda_k) c &\leq q^k \\ &\leq P_{\mu_{k-1}} \dots P_{\mu_{k-m}} q^{k-m} + (\lambda_{k-m} - \lambda_k) c. \end{aligned} \quad (3.13)$$

First, let us assume that the special state  $s$  corresponding to  $\mu_{k-m}, \dots, \mu_k$  as in Eqs. (3.7) and (3.8) is the fixed state  $t$  used in iteration (3.9); that is,

$$[P_{\mu_k} \dots P_{\mu_{k-m+1}}]_{it} \geq \epsilon, \quad i = 1, \dots, n, \quad (3.14)$$

$$[P_{\mu_{k-1}} \dots P_{\mu_{k-m}}]_{it} \geq \epsilon, \quad i = 1, \dots, n. \quad (3.15)$$

The right-hand side of Eq. (3.13) yields

$$q^k(i) \leq \sum_{j=1}^n [P_{\mu_{k-1}} \dots P_{\mu_{k-m}}]_{ij} q^{k-m}(j) + \lambda_{k-m} - \lambda_k,$$

so using Eq. (3.15) and the fact  $q^{k-m}(t) = 0$ , we obtain

$$q^k(i) \leq (1 - \epsilon) \max_j q^{k-m}(j) + \lambda_{k-m} - \lambda_k, \quad i = 1, \dots, n,$$

implying that

$$\max_j q^k(j) \leq (1-\epsilon) \max_j q^{k-m}(j) + \lambda_{k-m} - \lambda_k. \quad (3.16)$$

Similarly, from the left-hand side of Eq. (3.13) we obtain

$$\min_j q^k(j) \geq (1-\epsilon) \min_j q^{k-m}(j) + \lambda_{k-m} - \lambda_k, \quad (3.17)$$

and by subtracting the last two relations, we obtain the desired Eq. (3.10).

When the special state  $s$  corresponding to  $\mu_{k-m}, \dots, \mu_k$  as in Eqs. (3.7) and (3.8) is not equal to  $t$ , we define a related iterative process

$$\begin{aligned} \tilde{h}^{k+1}(i) &= (T\tilde{h}^k)(i) - (T\tilde{h}^k)(s), \quad i = 1, \dots, n, \\ \tilde{h}^0(i) &= h^0(i), \quad i = 1, \dots, n. \end{aligned} \quad (3.18)$$

Then, as earlier, we have

$$\max_i \tilde{q}^k(i) - \min_i \tilde{q}^k(i) \leq (1-\epsilon) \left( \max_i \tilde{q}^{k-m}(i) - \min_i \tilde{q}^{k-m}(i) \right), \quad (3.19)$$

where

$$\tilde{q}^k = \tilde{h}^{k+1} - \tilde{h}^k.$$

It is straightforward to verify, using Eqs. (3.9) and (3.18), that for all  $i$  and  $k$  we have

$$h^k(i) = \tilde{h}^k(i) + (T\tilde{h}^{k-1})(s) - (T\tilde{h}^{k-1})(t).$$

Therefore, the coordinates of both  $h^k$  and  $q^k$  differ from the coordinates of  $\tilde{h}^k$  and  $\tilde{q}^k$ , respectively, by a constant. It follows that

$$\max_i q^k(i) - \min_i q^k(i) = \max_i \tilde{q}^k(i) - \min_i \tilde{q}^k(i),$$

and from Eq. (3.19) we obtain the desired Eq. (3.10). Q.E.D.

As a by-product of the preceding proof, we obtain a rate of convergence estimate. By taking the limit in Eq. (3.11) as  $r \rightarrow \infty$ , we obtain

$$\max_i |h^k(i) - h(i)| \leq \frac{B(1-\epsilon)^{k/m}}{1-(1-\epsilon)^{1/m}}, \quad k = 0, 1, \dots,$$

so the bound on the error is reduced by  $(1-\epsilon)^{1/m}$  at each iteration. A sharper rate of convergence result can be obtained if we assume that there exists a unique optimal stationary policy  $\mu^*$ . Then, it is possible to show that the minimum in Eq. (3.12) is attained by  $\mu^*(i)$  for all  $i$  and all  $k$  after a certain index, so for such  $k$ , the relative value iteration takes the form  $h^{k+1} = T_{\mu^*} h^k - (T_{\mu^*} h^k)(t)c$ , and is governed by the largest eigenvalue modulus of the matrix  $P_{\mu^*}$  given by Eq. (3.6).

Note that contrary to the case of a discounted or a stochastic shortest path problem, the Gauss-Seidel version of the relative value iteration method need not converge. Indeed, the reader can construct examples of such behavior involving two-state systems and a single policy.

### Error Bounds

Similar to discounted problems, the relative value iteration method can be strengthened by the calculation of monotonic error bounds.

**Proposition 3.2:** Under the assumption of Prop. 3.1, the iterates  $h^k$  of the relative value iteration method (3.9) satisfy

$$c_k \leq c_{k+1} \leq \lambda \leq \bar{c}_{k+1} \leq \bar{c}_k, \quad (3.20)$$

where  $\lambda$  is the optimal average cost per stage for all initial states, and

$$c_k = \min_i [(Th^k)(i) - h^k(i)],$$

$$\bar{c}_k = \max_i [(Th^k)(i) - h^k(i)].$$

**Proof:** Let  $\mu_k(i)$  attain the minimum in

$$(Th^k)(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h^k(j) \right]$$

for each  $k$  and  $i$ . We have, using Eq. (3.9),

$$\begin{aligned} (Th^k)(i) &= g(i, \mu_k(i)) + \sum_{j=1}^n p_{ij}(\mu_k(i))h^k(j) \\ &= g(i, \mu_k(i)) + \sum_{j=1}^n p_{ij}(\mu_k(i))(Th^{k-1})(j) - (Th^{k-1})(t), \end{aligned}$$

and

$$h^k(i) \leq g(i, \mu_k(i)) + \sum_{j=1}^n p_{ij}(\mu_k(i))h^{k-1}(j) - (Th^{k-1})(t).$$

Subtracting the last two relations, we obtain

$$(Th^k)(i) - h^k(i) \geq \sum_{j=1}^n p_{ij}(\mu_k(i))((Th^{k-1})(j) - h^{k-1}(j)),$$

and it follows that

$$\min_i [(Th^k)(i) - h^k(i)] \geq \min_i [(Th^{k-1})(i) - h^{k-1}(i)],$$

or equivalently

$$\underline{c}_{k+1} \leq \underline{c}_k.$$

A similar argument shows that

$$\bar{c}_k \leq \bar{c}_{k+1}.$$

By Prop. 3.1 we have  $h^k(i) \rightarrow h(i)$  and  $(Th)(i) - h(i) = \lambda$  for all  $i$ , so that  $\underline{c}_k \rightarrow \lambda$ . Since  $\{\underline{c}_k\}$  is also nondecreasing, we must have  $\underline{c}_k \leq \lambda$  for all  $k$ . Similarly,  $\bar{c}_k \geq \lambda$  for all  $k$ . Q.E.D.

We now demonstrate the relative value iteration method and the error bounds (3.20) by means of an example.

### Example 3.2:

Consider an undiscounted version of the example of Section 1.3. We have

$$S = \{1, 2\}, \quad C = \{u^1, u^2\},$$

$$P(u^1) = \begin{pmatrix} p_{11}(u^1) & p_{12}(u^1) \\ p_{21}(u^1) & p_{22}(u^1) \end{pmatrix} = \begin{pmatrix} 3/4 & 1/4 \\ 3/4 & 1/4 \end{pmatrix},$$

$$P(u^2) = \begin{pmatrix} p_{11}(u^2) & p_{12}(u^2) \\ p_{21}(u^2) & p_{22}(u^2) \end{pmatrix} = \begin{pmatrix} 1/4 & 3/4 \\ 1/4 & 3/4 \end{pmatrix},$$

and

$$g(1, u^1) = 2, \quad g(1, u^2) = 0.5, \quad g(2, u^1) = 1, \quad g(2, u^2) = 3.$$

The mapping  $T$  has the form

$$(Th)(i) \approx \min \left\{ g(i, u^1) + \sum_{j=1}^2 p_{ij}(u^1)h(j), g(i, u^2) + \sum_{j=1}^2 p_{ij}(u^2)h(j) \right\}.$$

Letting  $t = 1$  be the reference state, the relative value iteration (3.9) takes the form

$$h^{k+1}(1) = 0$$

$$h^{k+1}(2) = (Th^k)(2) - (Th^k)(1).$$

The results of the computation starting with  $h^0(1) = h^0(2) = 0$  are shown in the table of Fig. 4.3.1.

$k$	$h^k(1)$	$h^k(2)$	$\underline{c}_k$	$\bar{c}_k$
0	0	0		
1	0	0.500	0.625	0.875
2	0	0.250	0.687	0.812
3	0	0.375	0.719	0.781
4	0	0.312	0.734	0.765
5	0	0.344	0.742	0.758
6	0	0.328	0.746	0.751
7	0	0.336	0.748	0.752
8	0	0.332	0.749	0.751
9	0	0.334	0.749	0.750
10	0	0.333	0.750	0.750

Figure 4.3.1 Iterates and error bounds generated by the relative value iteration method for the problem of Example 3.2.

We note an interesting application of the error bounds of Prop. 3.2. Suppose that for some vector  $h$ , we calculate a  $\mu$  such that

$$T_\mu h = Th.$$

Then by applying Prop. 3.2 to the original problem and also to the modified problem where the only stationary policy is  $\mu$ , we obtain

$$\underline{c} \leq J^*(i) \leq J_\mu(i) \leq \bar{c},$$

where

$$\underline{c} = \min_i [(Th)(i) - h(i)], \quad \bar{c} = \max_i [(Th)(i) - h(i)].$$

We thus obtain a bound on the degree of suboptimality of  $\mu$ . This bound can be proved in a more general setting, where  $J^*(i)$  is not necessarily independent of the initial state  $i$  (see Exercise 4.10).

### Other Versions of the Relative Value Iteration Method

As mentioned earlier, the condition for convergence of the relative value iteration method given in Prop. 3.1 is stronger than the conditions of Prop. 2.6 for the optimal average cost per stage to be independent of the initial state. We now show that we can bypass this difficulty by modifying

The problem without affecting either the optimal cost or the optimal policies and by applying the relative value iteration method to the modified problem.

Let  $\tau$  be any scalar with

$$0 < \tau < 1,$$

and consider the problem that results when each transition matrix  $P_\mu$  corresponding to a stationary policy  $\mu$  is replaced by

$$\tilde{P}_\mu = \tau P_\mu + (1 - \tau)I, \quad (3.21)$$

where  $I$  is the identity matrix. Note that  $\tilde{P}_\mu$  is a transition probability matrix with the property that, at every state, a self-transition occurs with probability at least  $(1 - \tau)$ . This destroys any periodic character that  $P_\mu$  may have. For another view of the same point, note that each eigenvalue of  $\tilde{P}_\mu$  is of the form  $\tau\gamma + (1 - \tau)$ , where  $\gamma$  is an eigenvalue of  $P_\mu$ . Therefore, all eigenvalues  $\gamma \neq 1$  of  $P_\mu$  that lie on the unit circle are mapped into eigenvalues of  $\tilde{P}_\mu$  strictly inside the unit circle.

Bellman's equation for the modified problem is

$$\hat{\lambda}_\mu c + \hat{h}_\mu = g_\mu + \tilde{P}_\mu \hat{h}_\mu = g_\mu + (\tau P_\mu + (1 - \tau)I)\hat{h}_\mu,$$

which can be written as

$$\hat{\lambda}_\mu c + \tau \hat{h}_\mu = g_\mu + P_\mu(\tau \hat{h}_\mu),$$

We observe that this equation is the same as Bellman's equation for the original problem,

$$\lambda_\mu c + h_\mu = g_\mu + P_\mu h_\mu,$$

with the identification

$$h_\mu = \tau \hat{h}_\mu.$$

It follows from Cor. 2.1.1 that if the average cost per stage for the original problem is independent of  $i$  for every  $\mu$ , then the same is true for the modified problem. Furthermore, the costs of all stationary policies, as well as the optimal cost, are equal for both the original and the modified problem.

Consider now the relative value iteration method (3.9) for the modified problem. A straightforward calculation shows that it takes the form

$$\begin{aligned} h^{k+1}(i) &= (1 - \tau)h^k(i) + \min_{u \in U(i)} \left[ g(i, u) + \tau \sum_{j=1}^n p_{ij}(u)h^k(j) \right] \\ &= \min_{u \in U(i)} \left[ g(t, u) + \tau \sum_{j=1}^n p_{tj}(u)h^k(j) \right], \end{aligned} \quad (3.22)$$

where  $t$  is some fixed state with  $h^0(t) = 0$ . Note that this iteration is as easy to execute as the original version. It is convergent, however, under weaker conditions than those required in Prop. 3.1.

**Proposition 3.3:** Assume that each stationary policy is unchain. Then, for  $0 < \tau < 1$ , the sequences  $\{h^k(i)\}$  generated by the modified relative value iteration (3.22) satisfy

$$\lim_{k \rightarrow \infty} h^k(i) = \frac{h(i)}{\tau},$$

$$\lim_{k \rightarrow \infty} \min_{u \in U(i)} \left[ g(t, u) + \tau \sum_{j=1}^n p_{tj}(u)h^k(j) \right] = \lambda, \quad (3.23)$$

where  $\lambda$  is the optimal average cost per stage and  $h$  is a differential cost vector.

**Proof:** The proof consists of showing that the conditions of Prop. 3.1 are satisfied for the modified problem involving the transition probability matrices  $\tilde{P}_\mu$  of Eq. (3.21).

Indeed, let  $m > m_M$ , where  $n$  is the number of states and  $n_M$  is the number of distinct stationary policies. Consider a set of control functions  $\mu_0, \mu_1, \dots, \mu_m$ . Then at least one  $\mu$  is repeated  $n$  times within the subset  $\mu_1, \dots, \mu_{m-1}$ . Let  $s$  be a state belonging to the recurrent class of the Markov chain corresponding to  $\mu$ . Then the conditions

$$[\tilde{P}_{\mu_m} \cdots \tilde{P}_{\mu_1}]_{is} \geq \epsilon, \quad i = 1, \dots, n,$$

$$[\tilde{P}_{\mu_{m-1}} \cdots \tilde{P}_{\mu_0}]_{is} \geq \epsilon, \quad i = 1, \dots, n,$$

are satisfied for some  $\epsilon$  because, in view of Eq. (3.21), when there is a positive probability of reaching  $s$  from  $i$  at some stage, there is also a positive probability of reaching it at any subsequent stage. **Q.E.D.**

Note that, since the modified value iteration method is nothing but the ordinary method applied to a modified problem, the error bounds of Prop. 3.2 apply in appropriately modified form.

#### 4.3.2 Policy Iteration

The policy iteration algorithm for the average cost problem is similar to those described in Sections 1.3 and 2.2. Given a stationary policy, one obtains an improved policy by means of a minimization process until no further improvement is possible. *We will assume throughout this section*

that every stationary policy encountered in the course of the algorithm is unichain.

At the  $k$ th step of the policy iteration algorithm, we have a stationary policy  $\mu^k$ . We then perform a *policy evaluation* step; that is, we obtain corresponding average and differential costs  $\lambda^k$  and  $h^k(i)$  satisfying

$$\lambda^k + h^k(i) = g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i))h^k(j), \quad i = 1, \dots, n, \quad (3.24)$$

or equivalently

$$\lambda^k c + h^k = T_{\mu^k} h^k = g_{\mu^k} + P_{\mu^k} h^k,$$

Note that  $\lambda^k$  and  $h^k$  can be computed as the unique solution of the linear system of equations (3.24) together with the normalizing equation  $h^k(t) = 0$ , where  $t$  is any state (cf. Prop. 2.5). This system can be solved either directly or iteratively using the relative value iteration method or by an adaptive aggregation method, as discussed later.

We subsequently perform a *policy improvement* step; that is, we find a stationary policy  $\mu^{k+1}$ , where for all  $i$ ,  $\mu^{k+1}(i)$  is such that

$$g(i, \mu^{k+1}(i)) + \sum_{j=1}^n p_{ij}(\mu^{k+1}(i))h^k(j) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^n p_{ij}(u)h^k(j) \right], \quad (3.25)$$

or equivalently

$$T_{\mu^{k+1}} h^k = Th^k.$$

If  $\mu^{k+1} = \mu^k$ , the algorithm terminates; otherwise, the process is repeated with  $\mu^{k+1}$  replacing  $\mu^k$ .

There is an easy proof, given in Exercise 4.11, that the policy iteration algorithm terminates finitely if we assume that the Markov chain corresponding to each  $\mu^k$  is irreducible (is unichain and has no transient states). To prove the result without this assumption, we impose the following restriction in the way the algorithm is operated: *if  $\mu^k(i)$  attains the minimum in Eq. (3.25), we choose  $\mu^{k+1}(i) = \mu^k(i)$  even if there are other controls attaining the minimum in addition to  $\mu^k(i)$ .* We then have:

**Proposition 3.4:** If all the generated policies are unichain, the policy iteration algorithm terminates finitely with an optimal stationary policy.

It is convenient to state the main argument needed for the proof of Prop. 3.4 as a lemma:

**Lemma 3.1:** Let  $\mu$  be a unichain stationary policy, and let  $\lambda$  and  $h$  be corresponding average and differential costs satisfying

$$\lambda c + h = T_{\mu} h, \quad (3.26)$$

as well as the normalization condition

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P_{\mu}^k h = 0. \quad (3.27)$$

[The above limit and the limit in the following Eq. (3.29) are shown to exist in Prop. 1.1.] Let  $\{\bar{\mu}, \bar{\mu}, \dots\}$  be the policy obtained from  $\mu$  via the policy iteration step described previously, and let  $\bar{\lambda}$  and  $\bar{h}$  be corresponding average and differential cost satisfying

$$\bar{\lambda} c + \bar{h} = T_{\bar{\mu}} (\bar{h}) \quad (3.28)$$

and

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P_{\bar{\mu}}^k \bar{h} = 0. \quad (3.29)$$

Then if  $\bar{\mu} \neq \mu$ , we must have either (1)  $\bar{\lambda} < \lambda$ , or (2)  $\bar{\lambda} = \lambda$  and  $\bar{h}(i) \leq h(i)$  for all  $i = 1, \dots, n$ , with strict inequality for at least one state  $i$ .

We note that, once Lemma 3.1 is established, it can be shown that the policy iteration algorithm will terminate finitely. The reason is that the vector  $h$  corresponding to  $\mu$  via Eq. (3.26) and (3.27) is unique by Prop. 2.5(b), and therefore the conclusion of Lemma 3.1 guarantees that no policy will be encountered more than once in the course of the algorithm. Since the number of stationary policies is finite, the algorithm must terminate finitely. If the algorithm stops at the  $k$ th step with  $\mu^{k+1} = \mu^k$ , we see from Eqs. (3.24) and (3.25) that

$$\lambda^k c + h^k = Th^k,$$

which by Prop. 2.4 implies that  $\mu^k$  is an optimal stationary policy. So to prove Prop. 3.4 there remains to prove Lemma 3.1.

**Proof of Lemma 3.1:** For notational convenience, denote

$$P = P_{\mu}, \quad \bar{P} = P_{\bar{\mu}}, \quad P^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P^k, \quad \bar{P}^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} \bar{P}^k.$$

$$g = g_\mu, \quad \bar{g} = g_{\bar{\mu}}.$$

Define the vector  $\delta$  by

$$\delta = \lambda c + h - \bar{g} - \bar{P}h. \quad (3.30)$$

We have, by assumption,  $T_{\bar{\mu}}^*h = Th \leq T_\mu h = \lambda c + h$ , or equivalently

$$\bar{g} + \bar{P}h \leq g + Ph = \lambda c + h, \quad (3.31)$$

from which we obtain

$$\delta(i) \geq 0, \quad i = 1, \dots, n. \quad (3.32)$$

Define also

$$\Delta = h - \bar{h}.$$

By combining Eq. (3.30) with the equation  $\bar{\lambda}c + \bar{h} = \bar{g} + \bar{P}\bar{h}$ , we obtain

$$\delta = (\lambda - \bar{\lambda})c + \Delta - \bar{P}\Delta.$$

Multiplying this relation with  $\bar{P}^k$  and adding from 0 to  $N - 1$ , we obtain

$$\sum_{k=0}^{N-1} \bar{P}^k \delta = N(\lambda - \bar{\lambda})c + \Delta - \bar{P}^N \Delta. \quad (3.33)$$

Dividing by  $N$  and taking the limit as  $N \rightarrow \infty$ , we obtain

$$\bar{P}^* \delta = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} \bar{P}^k \delta = (\lambda - \bar{\lambda})c. \quad (3.34)$$

In view of the fact  $\delta \geq 0$  [cf. Eq. (3.32)], we see that

$$\lambda \geq \bar{\lambda}.$$

If  $\lambda > \bar{\lambda}$ , we are done, so assume that  $\lambda = \bar{\lambda}$ . A state  $i$  is called  $\bar{P}$ -recurrent ( $\bar{P}$ -transient) if  $i$  belongs (does not belong, respectively) to the single recurrent class of the Markov chain corresponding to  $\bar{P}^*$ . From Eq. (3.34),  $\bar{P}^* \delta = 0$  and since  $\delta \geq 0$  and the elements of  $\bar{P}^*$  that are positive are those columns corresponding to  $\bar{P}$ -recurrent states, we obtain

$$\delta(i) = 0, \quad \text{for all } i \text{ that are } \bar{P}\text{-recurrent.} \quad (3.35)$$

It follows by construction of the algorithm that if  $i$  is  $\bar{P}$ -recurrent, then the  $i$ th rows of  $P$  and  $\bar{P}$  are identical [since  $\bar{\mu}(i) = \mu(i)$  for all  $i$  with  $\delta(i) = 0$ ]. Since  $P$  and  $\bar{P}$  have a single recurrent class, it follows that this

class is identical for both  $P$  and  $\bar{P}$ . From the normalization conditions (3.27) and (3.29), we then obtain  $h(i) = \bar{h}(i)$  for all  $i$  that are  $\bar{P}$ -recurrent. Equivalently,

$$\Delta(i) = 0, \quad \text{for all } i \text{ that are } \bar{P}\text{-recurrent.} \quad (3.36)$$

From Eq. (3.33) we obtain

$$\lim_{N \rightarrow \infty} \bar{P}^N \Delta = \Delta - \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \bar{P}^k \delta \leq \Delta - \delta.$$

In view of Eq. (3.36), the coordinates of  $\bar{P}^N \Delta$  corresponding to  $P$ -transient states tend to zero. Therefore, we have

$$\delta(i) \leq \Delta(i), \quad \text{for all } i \text{ that are } \bar{P}\text{-transient.} \quad (3.37)$$

From Eqs. (3.32) and (3.35) to (3.37), we see how that either  $\delta = 0$ , in which case  $\mu = \bar{\mu}$ , or else  $\Delta \geq 0$  with strict inequality  $\Delta(i) > 0$  for at least one  $\bar{P}$ -transient state  $i$ . Q.E.D.

We now demonstrate the policy iteration algorithm by means of the example of the previous section.

### Example 3.2: (continued)

Let

$$\mu^0(1) = u^1, \quad \mu^0(2) = u^2.$$

We take  $t = 1$  as a reference state and obtain  $\lambda_{\mu^0}$ ,  $h_{\mu^0}(1)$ , and  $h_{\mu^0}(2)$  from the system of equations

$$\lambda_{\mu^0} + h_{\mu^0}(1) = g(1, u^1) + p_{11}(u^1)h_{\mu^0}(1) + p_{12}(u^1)h_{\mu^0}(2),$$

$$\lambda_{\mu^0} + h_{\mu^0}(2) = g(2, u^2) + p_{21}(u^2)h_{\mu^0}(1) + p_{22}(u^2)h_{\mu^0}(2),$$

$$h_{\mu^0}(1) = 0.$$

Substituting the data of the problem,

$$\lambda_{\mu^0} = 2 + \frac{1}{4}h_{\mu^0}(2), \quad \lambda_{\mu^0} + h_{\mu^0}(2) = 3 + \frac{3}{4}h_{\mu^0}(2),$$

from which

$$\lambda_{\mu^0} = \frac{5}{2}, \quad h_{\mu^0}(1) = 0, \quad h_{\mu^0}(2) = 2.$$

We now find  $\mu^1(1)$  and  $\mu^1(2)$  by the minimization indicated in Eq. (3.25). We determine

$$\begin{aligned} \min & [g(1, u^1) + p_{11}(u^1)h_{\mu^0}(1) + p_{12}(u^1)h_{\mu^0}(2), \\ & g(1, u^2) + p_{11}(u^2)h_{\mu^0}(1) + p_{12}(u^2)h_{\mu^0}(2)] \\ & = \min \left[ 2 + \frac{1}{4} \cdot 2, 0.5 + \frac{3}{4} \cdot 2 \right] \\ & = \min[2.5, 2] \end{aligned}$$

and

$$\begin{aligned} \min & [g(2, u^1) + p_{21}(u^1)h_{\mu^0}(1) + p_{22}(u^1)h_{\mu^0}(2), \\ & g(2, u^2) + p_{21}(u^2)h_{\mu^0}(1) + p_{22}(u^2)h_{\mu^0}(2)] \\ & = \min \left[ 1 + \frac{1}{4} \cdot 2, 3 + \frac{3}{4} \cdot 2 \right] \\ & = \min[1.5, 4.5]. \end{aligned}$$

The minimization yields

$$\mu^1(1) = u^2, \quad \mu^1(2) = u^1.$$

We obtain  $\lambda_{\mu^1}$ ,  $h_{\mu^1}(1)$ , and  $h_{\mu^1}(2)$  from the system of equations

$$\begin{aligned} \lambda_{\mu^1} + h_{\mu^1}(1) &= g(1, u^2) + p_{11}(u^2)h_{\mu^1}(1) + p_{12}(u^2)h_{\mu^1}(2), \\ \lambda_{\mu^1} + h_{\mu^1}(2) &= g(2, u^1) + p_{21}(u^1)h_{\mu^1}(1) + p_{22}(u^1)h_{\mu^1}(2), \\ h_{\mu^1}(1) &= 0. \end{aligned}$$

By substituting the data of the problem, we obtain

$$\lambda_{\mu^1} = \frac{3}{4}, \quad h_{\mu^1}(1) = 0, \quad h_{\mu^1}(2) = \frac{1}{3}.$$

We find  $\mu^2(1)$  and  $\mu^2(2)$  by determining the minimum in

$$\begin{aligned} \min & [g(1, u^1) + p_{11}(u^1)h_{\mu^1}(1) + p_{12}(u^1)h_{\mu^1}(2), \\ & g(1, u^2) + p_{11}(u^2)h_{\mu^1}(1) + p_{12}(u^2)h_{\mu^1}(2)] \\ & = \min \left[ 2 + \frac{1}{4} \cdot \frac{1}{3}, 0.5 + \frac{3}{4} \cdot \frac{1}{3} \right] \\ & = \min[2.08, 0.75], \end{aligned}$$

and

$$\begin{aligned} \min & [g(2, u^1) + p_{21}(u^1)h_{\mu^1}(1) + p_{22}(u^1)h_{\mu^1}(2), \\ & g(2, u^2) + p_{21}(u^2)h_{\mu^1}(1) + p_{22}(u^2)h_{\mu^1}(2)] \\ & = \min \left[ 1 + \frac{1}{4} \cdot \frac{1}{3}, 3 + \frac{3}{4} \cdot \frac{1}{3} \right] \\ & = \min[1.08, 3.25]. \end{aligned}$$

The minimization yields

$$\mu^2(1) = \mu^1(1) = u^2, \quad \mu^2(2) = \mu^1(2) = u^1,$$

and hence the preceding policy is optimal and the optimal average cost is  $\lambda_{\mu^1} = 3/4$ .

The algorithm of this section shares some of the features of other types of policy iteration algorithms. In particular, it is possible to carry out policy evaluation approximately by using a few relative value iterations; see [Put94] for an analysis. Note also that in specially structured problems one may be able to confine policy iteration within a convenient subset of policies, for which policy evaluation is facilitated.

### Adaptive Aggregation

Consider now an extension to the average cost problem of the aggregation method described in Section 1.3.3. For a given unichain stationary policy  $\mu$ , we want to calculate an approximation to the pair  $(\lambda_\mu, h_\mu)$  satisfying Bellman's equation  $\lambda_\mu c + h_\mu = T_\mu h_\mu$  and  $h_\mu(n) = 0$ , where the state  $n$  is viewed as the reference state. By expressing  $\lambda_\mu$  as  $\lambda_\mu = (T_\mu h_\mu)(n)$ , we can eliminate it from this system of equations, and obtain  $h_\mu = T_\mu h_\mu - (T_\mu h_\mu)(n)c$ . This equation is written compactly as

$$h_\mu = \hat{T}h_\mu,$$

where the mapping  $\hat{T}$  is defined by

$$\hat{T}h = g_r + P_r h,$$

and

$$g_r = (I - cc_n')g_\mu, \quad P_r = (I - cc_n')P_\mu,$$

with  $c_n = (0, 0, \dots, 0, 1)'$ .

We partition the set of states into disjoint subsets  $S_1, S_2, \dots, S_m$  that are viewed as aggregate states. We assume that one of the subsets, say  $S_m$ , consists of just the reference state  $n$ ; that is,  $S_m = \{n\}$ . As in Section 1.3.3, consider the  $n \times m$  matrix  $W$  whose  $i$ th column has unit entries at coordinates corresponding to states in  $S_i$  and all other entries equal to zero. Consider also an  $m \times n$  matrix  $Q$  such that the  $i$ th row of  $Q$  is a probability distribution  $(q_{is})$  with  $q_{is} = 0$  if  $s \notin S_i$ . Note that  $QW = I$ , and that the  $m \times m$  matrix  $R = QP_\mu W$  is the transition probability matrix of the aggregate Markov chain, whose states are the  $m$  aggregate states.

Suppose now that we have an estimate  $\hat{h}$  of  $h_\mu$  and that we postulate that over the states  $s$  of every aggregate state  $S_i$  the variation  $h_\mu(s) - \hat{h}(s)$

is constant. This amounts to hypothesizing that for some  $m$ -dimensional vector  $y$  we have

$$h_\mu - h = Wy.$$

By combining the equations  $\hat{T}h \approx g_r + P_r h$  and  $h_\mu \approx g_r + P_r h_\mu$ , we have

$$(I - P_r)(h_\mu - h) = \hat{T}h - h.$$

By multiplying both sides of this equation with  $Q$ , and by using the relations  $h_\mu - h = Wy$  and  $QW = I$ , we obtain

$$(I - QP_r W)y = Q(\hat{T}h - h).$$

Assuming that the matrix  $I - QP_r W$  is invertible, this equation can be solved for  $y$ . Also, by applying  $\hat{T}$  to both sides of the equation  $h_\mu \approx h + Wy$ , we obtain

$$h_\mu = \hat{T}h_\mu = \hat{T}h + P_r Wy.$$

Thus the aggregation iteration for average cost problems is as follows:

### Aggregation Iteration

**Step 1:** Compute  $\hat{T}h = g_r + P_r h$ , where

$$g_r = (I - cc'_n)g_{\mu}, \quad P_r = (I - cc'_n)P_\mu.$$

**Step 2:** Delineate the aggregate states (i.e., define  $W$ ) and specify the matrix  $Q$ .

**Step 3:** Solve for  $y$  the system

$$(I - QP_r W)y = Q\hat{T}h,$$

and approximate  $h_\mu$  using

$$h := \hat{T}h + P_r Wy.$$

For the iteration to be valid, the matrix  $I - QP_r W$  must be invertible. We will show that this is guaranteed under an aperiodicity assumption such as the one used to prove convergence of the relative value iteration method (cf. Prop. 3.1). In particular, we assume that all the eigenvalues of the transition matrix  $R - QP_r W$  of the aggregate Markov chain, except for a single unity eigenvalue, lie strictly within the unit circle. Let us denote by  $e$  the  $m$ -dimensional vector of all 1's, and by  $e_m$  the  $m$ -dimensional vector

with last coordinate 1, and all other coordinates 0. Then using the easily verified relations  $Qe = e$  and  $e'_m Q = e'_m$ , we see that

$$QP_r W = (I - \bar{c}\bar{c}'_m)R.$$

From the analysis of Example 3.1 in Section 4.3, we have that  $QP_r W$  has  $m - 1$  eigenvalues that are equal to the  $m - 1$  nonunity eigenvalues of  $R$  and has 0 as its  $m$ th eigenvalue. Thus  $QP_r W$  must have all its eigenvalues strictly within the unit circle, and it follows that the matrix  $I - QP_r W$  is invertible.

In an adaptive aggregation method, a key issue is how to identify the aggregate states  $S_1, \dots, S_m$  in a way that the error  $h_\mu - h$  is of similar magnitude in each aggregate state. Similar to Section 4.3.3, one way to do this is to view  $\hat{T}h$  as an approximation to  $h_\mu$  and to group together states  $i$  with comparable magnitudes of  $(\hat{T}h)(i) - h(i)$ . As discussed in Section 4.3.3, this type of aggregation method can be greatly improved by interleaving each aggregation iteration with multiple relative value iterations (applications of the mapping  $\hat{T}$  on the current iterate). We refer to [BeC89] for further experimentation, analysis, and discussion.

### 4.3.3 Linear Programming

Let us now develop a linear programming-based solution method, assuming that any one of the conditions of Prop. 3.3 holds, so that the optimal average cost  $\lambda^*$  is independent of the initial state, and together with an associated differential cost vector  $h^*$ , satisfies  $\lambda^* c + h^* = \hat{T}h^*$ . Consider the following optimization problem in the variables  $\lambda$  and  $h(i)$ ,  $i = 1, \dots, n$ ,

$$\begin{aligned} & \text{maximize } \lambda \\ & \text{subject to } \lambda + h(i) \leq (\hat{T}h)(i), \quad i = 1, \dots, n, \end{aligned}$$

which is equivalent to the linear program

$$\begin{aligned} & \text{maximize } \lambda \\ & \text{subject to } \lambda + h(i) \leq g(i, u) + \sum_{j=1}^n p_{ij}(u)h(j), \quad i = 1, \dots, n, \quad u \in U(i). \end{aligned} \tag{3.38}$$

A nearly verbatim repetition of the proof of Prop. 2.1 shows that if  $(\lambda, h)$  is a feasible solution, that is,  $\lambda c + h \leq \hat{T}h$ , then  $\lambda \leq \lambda^*$ , which implies that  $(\lambda^*, h^*)$  is an optimal solution of the linear program (3.38). Furthermore, in any optimal solution  $(\bar{\lambda}, \bar{h})$  of the linear program (3.38), we have  $\bar{\lambda} = \lambda^*$ .

There is a linear program, which is dual to the above and which admits an interesting interpretation. In particular, the duality theory of

linear programming (see e.g., [Dam63]) asserts that the following (dual) linear program

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n \sum_{u \in U(i)} q(i, u) g(i, u) \\ & \text{subject to} \quad \sum_{u \in U(j)} q(j, u) = \sum_{i=1}^n \sum_{u \in U(i)} q(i, u) p_{ij}(u), \quad j = 1, \dots, n, \\ & \quad \sum_{i=1}^n \sum_{u \in U(i)} q(i, u) = 1, \\ & \quad q(i, u) \geq 0, \quad i = 1, \dots, n, \quad u \in U(i), \end{aligned} \quad (3.39)$$

has the same optimal value as the (primal) program (3.38). The variables  $q(i, u)$ ,  $i = 1, \dots, n$ ,  $u \in U(i)$ , of the dual program can be interpreted as the steady-state probabilities that state  $i$  will be visited at the typical transition and that control  $u$  will then be applied. The constraints of the dual program are the constraints that  $q(i, u)$  must satisfy in order to be feasible steady-state probabilities under some *randomized* stationary policy, that is, a policy that chooses at state  $i$  the control  $u$  probabilistically, by sampling the constraint set  $U(i)$  according to the probabilities  $q(i, u)$ ,  $u \in U(i)$ . The cost function

$$\sum_{i=1}^n \sum_{u \in U(i)} q(i, u) g(i, u)$$

of the dual problem is the steady-state average cost per transition. Duality theory asserts that the minimal value of this cost is  $\lambda^*$ , thus implying that the optimal average cost per stage that can be obtained using randomized stationary policies is no better than what can be achieved with ordinary (deterministic) stationary policies. Indeed, it can be verified that if  $\mu^*$  is an optimal (deterministic) stationary policy that is unichain, and  $p_i^*$  is the steady-state probability of state  $i$  in the corresponding Markov chain, then

$$q^*(i, u) = \begin{cases} p_i^* & \text{if } u = \mu^*(i), \\ 0 & \text{otherwise,} \end{cases}$$

is an optimal solution of the dual problem (3.39).

#### 4.3.4 Simulation-Based Methods

We now describe briefly how the simulation-based methods of Section 2.3 can be adapted to work for average cost problems. We make a slight change in the problem definition to make the notation better suited for

the simulation context. In particular, instead of considering the expected cost  $g(i, u)$  at state  $i$  under control  $u$ , we allow the cost  $g$  to depend on the next state  $j$ . Thus our notation for the cost per stage is now  $g(i, u, j)$ , as in the simulation-related material for stochastic shortest path and discounted problems (cf. Section 2.3). All the results and the entire analysis of the preceding sections can be rewritten in terms of the new notation by replacing  $g(i, u)$  with  $\sum_{j=1}^n p_{ij}(u)g(i, u, j)$ .

#### Policy Iteration

In order to implement a simulation-based policy iteration algorithm like the one of Section 2.3.1, we need to be able to carry out the policy evaluation step for a given unichain policy  $\mu$ . This can be done by using the connection with the stochastic shortest path formulation described in Section 4.1. We fix a state  $t$ , and we evaluate the cost of the given policy  $\mu$  for two stochastic shortest path problems whose termination state is (essentially)  $t$ . In particular, we evaluate by Monte-Carlo simulation or TD( $\lambda$ ) the expected cost  $C_t$  from each state  $i$  up to reaching  $t$  [cf. Eq. (2.15)]. This requires the generation of many trajectories terminating at state  $t$  and the corresponding sample costs. Simultaneously with the evaluation of the costs  $C_t$ , we evaluate the expected number of transitions  $N_t$  from each state  $i$  up to reaching  $t$  [cf. Eq. (2.16)]. Then the average cost  $\lambda_\mu$  of the policy is obtained as

$$\lambda_\mu = \frac{C_t}{N_t}, \quad (3.40)$$

[cf. Eq. 2.17)], and the associated differential costs are obtained as

$$h_\mu(i) = C_t - \lambda_\mu N_t, \quad i = 1, \dots, n, \quad (3.41)$$

[cf. Eq. (2.18)].

To implement a simulation-based approximate policy iteration algorithm, a similar procedure can be used. In particular, one can obtain by Monte-Carlo simulation or TD(1) functions  $\hat{C}_t(r)$  and  $\hat{N}_t(r)$  that depend on a parameter vector  $r$  and approximate the costs  $C_t$  and  $N_t$  of the corresponding stochastic shortest path problems, as described in Section 2.3.3. Then, one can use

$$\tilde{\lambda}_\mu(r) = \frac{\hat{C}_t(r)}{\hat{N}_t(r)}$$

as an approximation to the average cost per stage of the policy and also use

$$\tilde{h}_\mu(i) = \hat{C}_t(r) - \tilde{\lambda}_\mu(r)\hat{N}_t(r), \quad i = 1, \dots, n,$$

as approximations to the corresponding differential costs [cf. Eqs. (3.40) and (3.41)].

Note here that because the approximations  $\hat{C}_t(r)$  and  $\hat{N}_t(r)$  play an important role in the calculations, it may be worth doing some extra simulations starting from the reference state  $t$  to ensure that these approximations are nearly exact.

### Value Iteration and $Q$ -Learning

To derive the appropriate form of the  $Q$ -learning algorithm of Section 2.3.2, we form an auxiliary average cost problem by augmenting the original system with one additional state for each possible pair  $(i, u)$  with  $u \in U(i)$ . The probabilistic transition mechanism from the original states is the same as for the original problem, while the probabilistic transition mechanism from an auxiliary state  $(i, u)$  is that we move only to states  $j$  of the original problem with corresponding probabilities  $p_{ij}(u)$  and costs  $g(i, u, j)$ . It can be seen that the auxiliary problem has the same optimal average cost per stage  $\lambda$  as the original, and that the corresponding Bellman's equation is

$$\lambda + h(i) = \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + h(j)), \quad i = 1, \dots, n, \quad (3.42)$$

$$\lambda + Q(i, u) = \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + h(j)), \quad i = 1, \dots, n, \quad u \in U(i), \quad (3.43)$$

where  $Q(i, u)$  is the differential cost corresponding to  $(i, u)$ . Taking the minimum over  $u$  in Eq. (3.43) and substituting in Eq. (3.42), we obtain

$$h(i) = \min_{u \in U(i)} Q(i, u), \quad i = 1, \dots, n. \quad (3.44)$$

Substituting the above form of  $h(i)$  in Eq. (3.43), we obtain Bellman's equation in a form that exclusively involves the  $Q$ -factors:

$$\lambda + Q(i, u) = \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right), \quad i = 1, \dots, n, \quad u \in U(i). \quad (3.45)$$

Let us now apply to the auxiliary problem the following variant of the relative value iteration

$$h^{k+1} = Th^k - h^k(t)c,$$

(see Exercise 4.5 for the case where  $c = 0$ ,  $p_t = 1$ , and  $p_j = 0$  for  $j \neq t$ ). We then obtain the iteration

$$h^{k+1}(i) = \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + h^k(j)) - h^k(t), \quad i = 1, \dots, n, \quad (3.46)$$

$$Q^{k+1}(i, u) = \sum_{j=1}^n p_{ij}(u)(g(i, u, j) + h^k(j)) - h^k(t), \quad i = 1, \dots, n, \quad u \in U(i). \quad (3.47)$$

From these equations, we have that

$$h^k(i) = \min_{u \in U(i)} Q^k(i, u), \quad i = 1, \dots, n, \quad (3.48)$$

and by substituting the above form of  $h^k$  in Eq. (3.47), we obtain the following relative value iteration for the  $Q$ -factors

$$Q^{k+1}(i, u) = \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q^k(j, u') \right) - \min_{u' \in U(i)} Q^k(t, u'). \quad (3.49)$$

This iteration is analogous to the value iteration for  $Q$ -factors in the stochastic shortest path context. The sequence of values  $\min_{u \in U(i)} Q^k(i, u)$  is expected to converge to the optimal average cost per stage and the sequences of values  $\min_{u \in U(i)} Q(i, u)$  are expected to converge to differential costs  $h(i)$ .

An incremental version of the preceding iteration that involves a positive stepsize  $\gamma$  is given by

$$Q(i, u) := (1 - \gamma)Q(i, u) + \gamma \left( \sum_{j=1}^n p_{ij}(u) \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') \right) - \min_{u' \in U(i)} Q(t, u') \right), \quad (3.50)$$

[compare with Eq. (3.8) in Section 2.3]. The natural form of the  $Q$ -learning method for the average cost problem is an approximate version of this iteration, whereby the expected value is replaced by a single sample, i.e.,

$$Q(i, u) := Q(i, u) + \gamma \left( g(i, u, j) + \min_{u' \in U(j)} Q(j, u') - \min_{u' \in U(i)} Q(t, u') - Q(i, u) \right), \quad (3.51)$$

where  $j$  and  $g(i, u, j)$  are generated from the pair  $(i, u)$  by simulation.

### Minimization of the Bellman Equation Error

There is a straightforward extension of the method of Section 2.3.3 for obtaining an approximate representation of the average cost  $\lambda$  and associated differential costs  $h(i)$ , based on minimizing the squared error in Bellman's equation. Here we approximate  $\lambda$  by  $\hat{\lambda}(r)$  and  $h(i)$  by  $\hat{h}(i, r)$ ,

where  $r$  is a vector of unknown parameters/weights. We impose a normalization constraint such as  $\hat{h}(t, r) = 0$ , where  $t$  is a fixed state, and we minimize the error in Bellman's equation by solving the problem

$$\min_r \sum_{r \in S} \left| \hat{\lambda}(r) + \hat{h}(i, r) - \min_{u \in U(i)} \sum_{j=1}^n p_{ij}(u) (g(i, u, j) + \hat{h}(j, r)) \right|^2.$$

where  $S$  is a suitably chosen subset of "representative" states. This minimization may be attempted by using some gradient method of the type discussed in Section 2.3.3.

#### 4.4 INFINITE STATE SPACE

The standing assumption in the preceding sections has been that the state space is finite. Without finiteness of the state space, many of the results presented in the past three sections no longer hold. For example, whereas one could restrict attention to stationary policies for finite state systems, this is no longer true when the state space is infinite. The following example from [Ros83a] shows that if the state space is countable, there may not exist an optimal policy.

##### Example 4.1:

Let the state space be  $\{1, 1', 2, 2', 3, 3', \dots\}$ , and let there be two controls,  $u^1$  and  $u^2$ . The transition probabilities and costs per stage are

$$\begin{aligned} p_{i(i+1)}(u^1) &= 1, & p_{ii'}(u^2) &= 1, & i &= 1, 2, \dots, & c_i, f_i, \tilde{c}_i, \tilde{f}_i, \text{out}_i \\ p_{i'i'}(u^1) &= p_{ii'}(u^2) = 0, & i &= 1, 2, \dots, & & & \\ g(i, u^1) &= g(i, u^2) = 0, & i &= 1, 2, \dots, & & & \\ g(i', u^1) &= g(i', u^2) = -1 + \frac{1}{i}, & i &= 1, 2, \dots, & & & \end{aligned}$$

In words, at state  $i$  we may, at a cost 0, either move to state  $(i+1)$  or move to state  $i'$ , where we stay thereafter at a cost  $-1 + 1/i$  per stage.

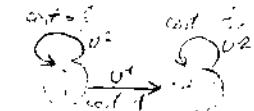
It can be seen that for every policy  $\pi$  and state  $i = 1, 2, \dots$ , we have  $J_\pi(i) > -1$ . However, for every state  $i$ , we can obtain an average cost per stage  $-1 + 1/j$ , where  $j \geq i$ , by moving to state  $j'$  once we get to state  $j$ . Hence, for every initial state  $i = 1, 2, \dots$ , an average cost per stage of  $-1$  can be approached arbitrarily closely, but cannot be attained by any policy.

Here is another example, from [Ros70], which shows that for a countable state space there may exist an optimal nonstationary policy, but not an optimal stationary policy.

##### Example 4.2:

Let the state space be  $\{1, 2, 3, \dots\}$ , and let there be two controls,  $u^1$  and  $u^2$ . The transition probabilities and costs per stage are

$$\begin{aligned} p_{i(i+1)}(u^1) &= p_{ii}(u^2) = 1, \\ g(i, u^1) &= 1, & g(i, u^2) &= \frac{1}{i}, & i &= 1, 2, \dots \end{aligned}$$



In words, at state  $i$  we may either move to state  $(i+1)$  at a cost 1 or stay at  $i$  at a cost  $1/i$ .

For any stationary policy  $\mu$  other than the policy for which  $\mu(i) = u^1$  for all  $i$ , let  $n(\mu)$  be the smallest integer for which

$$\mu(n(\mu)) = u^2.$$

Then the corresponding average cost per stage satisfies

$$J_\mu(i) = \frac{1}{n(\mu)} > 0, \quad \text{for all } i \text{ with } i \leq n(\mu).$$

For the policy where  $\mu(i) = u^1$  for all  $i$ , we have  $J_\mu(i) = 1$  for all  $i$ . Since the optimal cost per stage cannot be less than zero, it is clear that

$$\min_\pi J_\pi(i) = 0, \quad i = 1, 2, \dots$$

However, the optimal cost is not attained by any stationary policy, so no stationary policy is optimal. On the other hand, consider the nonstationary policy  $\pi^*$  that on entering state  $i$  chooses  $u^2$  for  $i$  consecutive times and then chooses  $u^1$ . If the starting state is  $i$ , the sequence of costs incurred is

$$\underbrace{\frac{1}{i}, \frac{1}{i}, \dots, \frac{1}{i}}_{i \text{ times}}, -1, \underbrace{\frac{1}{i+1}, \frac{1}{i+1}, \dots, \frac{1}{i+1}}_{(i+1) \text{ times}}, -1, \underbrace{\frac{1}{i+2}, \frac{1}{i+2}, \dots}_{\dots}$$

The average cost corresponding to this policy is

$$J_{\pi^*}(i) = \lim_{m \rightarrow \infty} \frac{2m}{\sum_{k=1}^m (i+k)} = 0, \quad i = 1, 2, 3, \dots$$

Hence the nonstationary policy  $\pi^*$  is optimal while, as shown previously, no stationary policy is optimal.

Generally, the analysis of average cost problems with an infinite state space is difficult, although there has been considerable progress (see the references). An important tool is Prop. 2.1, which admits a straightforward extension to the case where the state and control spaces are infinite. In particular, if we can find a scalar  $\lambda$  and a bounded function  $h$  such that Bellman's equation (2.1) holds, then by repeating the proof of Prop. 2.1, we can show that  $\lambda$  must be the optimal average cost per stage for all initial states. Among other situations, this result is useful when we can guess the right  $\lambda$  and  $h$ , and verify that they satisfy Bellman's equation. Some important special cases can be satisfactorily analyzed in this way (see the references). We describe one such case, the average cost version of the linear-quadratic problem examined in Chapters 4, and 5 of Vol. 1.

### Linear Systems with Quadratic Cost

Consider the linear-quadratic problem involving the system

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad k = 0, 1, \dots, \quad (4.1)$$

and the cost function

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} E_{\substack{x_k \\ u_k \\ w_k}} \left\{ \sum_{k=0}^{N-1} (x_k' Q x_k + \mu_k(x_k)' R \mu_k(x_k)) \right\}. \quad (4.2)$$

We make the same assumptions as in Section 8.1, that is,  $Q$  is positive semidefinite symmetric,  $R$  is positive definite symmetric, and  $w_k$  are independent, and have zero mean and finite second moments. We also assume that the pair  $(A, B)$  is controllable and that the pair  $(A, C)$ , where  $Q = C'C$ , is observable. Under these assumptions, it was shown in Section 4.1 of Vol. I that the Riccati equation

$$K_0 = 0, \quad (4.3)$$

$$K_{k+1} = A'(K_k - K_k B(B'K_k B + R)^{-1} B'K_k)A + Q \quad (4.4)$$

yields in the limit a matrix  $K$ ,

$$K = \lim_{k \rightarrow \infty} K_k, \quad (4.5)$$

which is the unique solution of the equation

$$K = A'(K - KB(B'KB + R)^{-1}B'K)A + Q \quad (4.6)$$

within the class of positive semidefinite symmetric matrices.

The optimal value of the  $N$ -stage costs

$$\frac{1}{N} E_{\substack{x_k \\ u_k \\ w_k}} \left\{ \sum_{k=0}^{N-1} (x_k' Q x_k + u_k' R u_k) \right\} \quad (4.7)$$

has been derived earlier and was seen to be equal to

$$\frac{1}{N} \left( x_0' K_N x_0 + \sum_{k=0}^{N-1} E\{w_k' K_k w_k\} \right).$$

Since  $K = \lim_{k \rightarrow \infty} K_k$  and

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} E\{w_k' K_k w_k\} = E\{w' K w\},$$

we see that the optimal  $N$ -stage costs tend to

$$\lambda = E\{w' K w\} \quad (4.8)$$

as  $N \rightarrow \infty$ . In addition, the  $N$ -stage optimal policy in its initial stages tends to the stationary policy

$$\mu^*(x) = -(B'KB + R)^{-1}B'KAx, \quad (4.9)$$

Furthermore, a simple calculation shows that, by the definition of  $\lambda$ ,  $K$ , and  $\mu^*(x)$ , we have

$$\lambda + x' K x = \min_u E\{x' Q x + u' R u + (Ax + Bu + w)' K (Ax + Bu + w)\},$$

while the minimum in the right-hand side of this equation is attained at  $u^* = \mu^*(x)$  as given by Eq. (4.9).

By repeating the proof of Prop. 2.1, we obtain

$$\begin{aligned} \lambda &\leq \frac{1}{N} E\{x_N' K x_N \mid x_0, \pi\} \\ &= \frac{1}{N} x_0' K x_0 + \frac{1}{N} E \left\{ \sum_{k=0}^{N-1} (x_k' Q x_k + u_k' R u_k) \mid x_0, \pi \right\}, \end{aligned}$$

with equality if  $\pi = \{\mu^*, \mu^*, \dots\}$ . Hence, if  $\pi$  is such that  $E\{x_N' K x_N \mid x_0, \pi\}$  is uniformly bounded over  $N$ , we have, by taking the limit as  $N \rightarrow \infty$  in the preceding relation,

$$\lambda \leq J_\pi(x), \quad x \in \mathbb{R}^n,$$

with equality if  $\pi = \{\mu^*, \mu^*, \dots\}$ . Thus the linear stationary policy given by Eq. (4.9) is optimal over all policies  $\pi$  with  $E\{x_N' K x_N \mid x_0, \pi\}$  bounded uniformly over  $N$ .

### 4.5 NOTES, SOURCES, AND EXERCISES

Several authors have contributed to the average cost problem ([How60], [Bro65], [Ros70], [Sch68], [Vei66], [Vei69]), most notably Blackwell ([Bla62]). An alternative detailed treatment to ours is given in [Put94]. An extensive survey containing many references is given in [ABF93].

The result of Prop. 2.6 under conditions (2) and (3) was shown in [Bat73] and [Ros70], respectively. The relative value iteration method of Section 4.3 is due to [Whi63], and its modified version of Eq. (3.22) is due

to [Sch71]. The error bounds of Prop. 3.2 are due to Odoni ([Odo69]). The value iteration method has been analyzed exhaustively in [Sch71], [ScF77], and [ScF78]. Convergence under slightly weaker conditions than those given here is shown in [Pla77]. The error bounds of Exercise 4.10 are due to Varaiya ([Var78]), who used them to construct a differential form of the value iteration method. Discrete-time versions of Varaiya's method are given in [PBW79]. The value iteration method based on stochastic shortest paths of Exercise 4.15 is new (see [Ber95c]).

The policy iteration algorithm can be generalized for problems where the optimal average cost per stage is not the same for every initial state (see [Bla62], [Put94], [Vei66], and [Der70]). The adaptive aggregation method is due to [BeC89].

The approximation procedures of Section 4.3.4, and the  $Q$ -learning algorithms of Section 4.3.4 and Exercise 4.16 are new. Alternative algorithms of the  $Q$ -learning type are given in [Sch93] and [Sin94].

For analysis of infinite horizon versions of inventory control problems, such as the ones of Section 4.2 of Vol. I, see [Igl63a], [Igl63b], and [Igl74]. Infinite state space models are discussed in [Kus78], [Sen86], [Las88], [Bor89], [Cav89a], [Cav89b], [Her89], [Sen89a], [Sen89b], [FAM90], [FAM91], [Cav91], [HHL91], [Sen91], [CaS92], [RIS92], [ABF93], [Sen93a], [Sen93b], and [Put94].

## EXERCISES

### 4.1

Solve the average cost version ( $\alpha = 1$ ) of the computer manufacturer's problem (Exercise 7.3, Vol. I).

### 4.2

Consider a stationary inventory control problem of the type considered in Section 4.2 of Vol. I but with the difference that the stock  $x_k$  can only take integer values from 0 to some integer  $M$ . The order  $u_k$  can take integer values with  $0 \leq u_k \leq M - x_k$ , and the random demand  $w_k$  can only take nonnegative integer values with  $P(w_k = 0) > 0$  and  $P(w_k = 1) > 0$ . Unsatisfied demand is lost, so stock evolves according to the equation  $x_{k+1} = \max(0, x_k + u_k - w_k)$ . The problem is to find an inventory policy that minimizes the average cost per stage. Show that there exists an optimal stationary policy and that the optimal cost is independent of the initial stock  $x_0$ .

### 4.3 [LiR71]

Consider a person providing a certain type of service to customers. The person receives at the beginning of each time period with probability  $p_i$  a proposal by a customer of type  $i$ , where  $i = 1, 2, \dots, n$ , who offers an amount of money  $M_i$ . We assume that  $\sum_{i=1}^n p_i \leq 1$ . The person may reject the offer, in which case the customer leaves and the person remains idle during that period, or the person may accept the offer in which case the person spends some time with that customer determined according to a Markov process with transition probabilities  $\beta_{ik}$ , where, for  $k = 1, 2, \dots$ ,

$\beta_{ik}$  = probability that the type  $i$  customer will leave after  $k$  periods, given that the customer has already stayed with the person for  $k-1$  periods.

The problem is to determine an acceptance-rejection policy that maximizes

$$\lim_{N \rightarrow \infty} \frac{1}{N} \{\text{Expected payment over } N \text{ periods}\}.$$

Consider two cases:

1.  $\beta_{ik} = \beta_i \in (0, 1)$  for all  $k$ .
2. For each  $i$  there exists  $\bar{k}_i$  such that  $\beta_{i\bar{k}_i} = 1$ .
  - (a) Formulate the person's problem as an average cost per stage problem, and show that the optimal cost is independent of the initial state.
  - (b) Show that there exists a scalar  $\lambda^*$  and an optimal policy that accepts the offer of a type  $i$  customer if and only if

$$\lambda^* T_i \leq M_i,$$

where  $T_i$  is the expected time spent with the type  $i$  customer given by

$$T_i = \beta_{i1} + \sum_{k=2}^{\infty} k \beta_{ik} (1 - \beta_{ik-1}) \cdots (1 - \beta_{i2}).$$

### 4.4

Let  $h^0$  be an arbitrary vector in  $\mathbb{R}^n$ , and define for all  $i$  and  $k \geq 1$

$$h_i^k = T^k h^0 - (T^k h^0)(i)e,$$

$$h^k = T^k h^0 - \frac{1}{n} \sum_{i=1}^n (T^k h^0)(i)e,$$

$$\tilde{h}^k = T^k h^0 - \min_{i=1, \dots, n} (T^k h^0)(i)e.$$

Let also  $h_i^0 = \hat{h}^0 = \tilde{h}^0 = h^0$ .

- (a) Show that the sequences  $\{h_i^k\}$ ,  $\{\hat{h}^k\}$ , and  $\{\tilde{h}^k\}$  are generated by the algorithms

$$h_i^{k+1} = Th_i^k + (Th_i^k)(i)c,$$

$$\hat{h}_i^{k+1} = T\hat{h}^k - \frac{1}{n} \sum_{i=1}^n (Th_i^k)(i)c,$$

$$\tilde{h}_i^{k+1} = T\tilde{h}^k + \min_{i=1,\dots,n} (T\tilde{h}^k)(i)c.$$

- (b) Show that the convergence result of Prop. 3.1 holds for the algorithms of part (a). *Hint:* Proposition 3.1 applies to the algorithms that generate  $\{h_i^k\}$ . Express  $\hat{h}^k$  and  $\tilde{h}^k$  as continuous functions of  $\{h_i^k\}$ ,  $i = 1, \dots, n$ .

#### 4.5 (Variants of Relative Value Iteration)

Consider the following two variants of the relative value iteration algorithm:

$$h^{k+1}(i) = (Th^k)(i) - \lambda^k, \quad i = 1, \dots, n,$$

where

$$\lambda^k = c + \sum_{j=1}^n p_j h^k(j),$$

or

$$\lambda^k = c + \sum_{j=1}^n p_j h^{k+1}(j).$$

Here  $c$  is an arbitrary scalar and  $(p_1, \dots, p_n)$  is an arbitrary probability distribution over the states of the system. Under the assumptions of Prop. 3.1, show that the sequence  $\{h^k\}$  converges to a vector  $h$  and the sequence  $\{\lambda^k\}$  converges to a scalar  $\lambda$  satisfying  $\lambda c + h = Th$ , so that by Prop. 2.1,  $\lambda$  is equal to the optimal average cost per stage for all initial states and  $h$  is an associated differential cost vector. *Hint:* Modify the problem by introducing an artificial state  $t'$  from which the system moves at a cost  $c$  to state  $j$  with probability  $p_j$ , for all  $u$ . Apply Prop. 3.1.

#### 4.6

Consider a deterministic system with two states 0 and 1. Upon entering state 0, the system stays there permanently at no cost. In state 1 there is a choice of staying there at no cost or moving to state 0 at cost 1. Show that every policy is average cost optimal, but the only stationary policy that is Blackwell optimal is the one that keeps the system in the state it currently is.

#### 4.7

Show that a Blackwell optimal policy is optimal over all policies (not just those that are stationary). *Hint:* Use the following result: If  $\{c_n\}$  is a nonnegative bounded sequence, then

$$\begin{aligned} \liminf_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} c_k &\leq \liminf_{\alpha \downarrow 1} (1-\alpha) \sum_{k=0}^{\infty} \alpha^k c_k \\ &\leq \limsup_{\alpha \uparrow 1} (1-\alpha) \sum_{k=0}^{\infty} \alpha^k c_k \\ &\leq \limsup_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} c_k. \end{aligned}$$

A proof of this result can be found in [Put94], p. 417.

#### 4.8 (Reduction to the Discounted Case)

For the finite-state average cost problem suppose there is a state  $t$  such that for some  $\beta > 0$  we have  $p_{it}(u) \geq \beta$  for all states  $i$  and controls  $u$ . Consider the  $(1-\beta)$ -discounted problem with the same state space, control space, and transition probabilities

$$\bar{p}_{ij}(u) = \begin{cases} (1-\beta)^{-1} p_{ij}(u) & \text{if } j \neq t, \\ (1-\beta)^{-1} (p_{jt}(u) - \beta) & \text{if } j = t. \end{cases}$$

Show that  $\beta \bar{J}(t)$  and  $\bar{J}(i)$  are optimal average and differential costs, respectively, where  $\bar{J}$  is the optimal cost function of the  $(1-\beta)$ -discounted problem.

#### 4.9 (Deterministic Finite-State Systems)

Consider a deterministic finite-state system. Suppose that the system is controllable in the sense that given any two states  $i$  and  $j$ , there exists a sequence of admissible controls that drives the state of the system from  $i$  to  $j$ . Consider the problem of finding an admissible control sequence  $\{u_0, u_1, \dots\}$  that minimizes

$$J_k(t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} g(x_k, u_k).$$

Show that the optimal cost is independent of the initial state, and that there exist optimal control sequences, which after a certain time index are periodic.

### 4.10 (Generalized Error Bounds)

Let  $h$  be any  $n$ -dimensional vector and  $\mu$  be such that

$$T_\mu h = Th.$$

Show that, for all  $i$ ,

$$\min_j [(Th)(j) - h(j)] \leq J^*(i) \leq J_\mu(i) \leq \max_j [(Th)(j) - h(j)],$$

regardless of whether  $J^*(i)$  is independent of the initial state  $i$ . Hint: Complete the details of the following argument. Let

$$\delta(i) = (Th)(i) - h(i), \quad i = 1, \dots, n,$$

and let  $\delta$  be the vector with coordinates  $\delta(i)$ . We have

$$T_\mu h = \delta + h, \quad T_\mu^2 h = T_\mu h + P_\mu \delta = \delta + P_\mu \delta + h$$

and, continuing in the same manner,

$$T_\mu^N h = \sum_{k=0}^{N-1} P_\mu^k \delta + h, \quad N = 1, 2, \dots$$

Hence

$$J_\mu = \lim_{N \rightarrow \infty} \frac{1}{N} T_\mu^N h = P_\mu^* \delta,$$

where

$$P_\mu^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} P_\mu^k,$$

proving the right-hand side of the desired relation. Also, let  $\pi = \{\mu_0, \mu_1, \dots\}$  be any admissible policy. We have

$$T_{\mu_N} h \geq \delta + h$$

from which we obtain

$$T_{\mu_{N+1}} T_{\mu_N} h \geq P_{\mu_{N+1}} \delta + T_{\mu_N} h \geq P_{\mu_{N+1}} \delta + \delta + h \geq 2 \min_j \delta(j) e + h.$$

Thus, for all  $i$ ,

$$\frac{1}{N+1} (T_{\mu_0} \cdots T_{\mu_N} h)(i) \geq \min_j \delta(j) + \frac{h(i)}{N+1}$$

and, taking the limit as  $N \rightarrow \infty$ , we obtain

$$J_\pi(i) \geq \min_j \delta(j).$$

Since  $\pi$  is arbitrary, we obtain the left-hand side of the desired relation.

### 4.11

Use Prop. 4.1 to show that in the policy iteration algorithm we have for all  $k$ ,

$$\lambda^{k+1} e = \lambda^k e + P_{\mu^{k+1}}^*(Th^k + h^k - \lambda^k e),$$

where

$$P_{\mu^{k+1}}^* = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} P_{\mu^{k+1}}^m.$$

Use this fact to show that if the Markov chain corresponding to  $\mu^{k+1}$  has no transient states and  $\mu^{k+1}$  is not optimal, then  $\lambda^{k+1} < \lambda^k$ .

### 4.12 (Policy Iteration for Linear-Quadratic Problems)

The purpose of this problem is to show that policy iteration works for linear-quadratic problems (even though neither the state space nor the control space are finite). Consider the problem of Section 4.4 under the usual controllability, observability, and positive (semi)definiteness assumptions. Let  $L_0$  be an  $m \times n$  matrix such that the matrix  $(A + BL_0)$  is stable.

- (a) Show that the average cost per stage corresponding to the stationary policy  $\mu^0$ , where  $\mu^0(x) = L_0 x$ , is of the form

$$J_{\mu^0} = E\{w^T K_0 w\},$$

where  $K_0$  is a positive semidefinite symmetric matrix satisfying the (linear) equation

$$K_0 = (A + BL_0)^T K_0 (A + BL_0) + Q + L_0^T R L_0,$$

- (b) Let  $\mu^1(x) = L_1 x = (R + B^T K_0 B)^{-1} B^T K_0 A x$  be the control function attaining the minimum for each  $x$  in the expression

$$\min_u \{u^T R u + (A x + B u)^T K_0 (A x + B u)\}.$$

Show that

$$J_{\mu^1} = E\{w^T K_1 w\} \leq J_{\mu^0},$$

where  $K_1$  is some positive semidefinite symmetric matrix.

- (c) Consider repeating the (policy iteration) process described in parts (a) and (b), thereby obtaining a sequence of positive semidefinite symmetric matrices  $\{K_k\}$ . Show that

$$K_k \rightarrow K,$$

where  $K$  is the optimal cost matrix of the problem.

### 4.13 (Alternative Analysis for the Unichain Case)

The purpose of this exercise is to show how to extend the average cost problem analysis based on the connection with the stochastic shortest path problem, which is given in Section 7.4 of Vol. 1. In particular, here this connection is used to show that there exists a solution  $(\lambda, h)$  to Bellman's equation  $\lambda c + h = Th$  if every policy that is optimal within the class of stationary policies is unichain, without resorting to the use of Blackwell optimal policies (cf. Prop. 2.6). For this we will use the stochastic shortest path theory of Section 2.1, and from the present chapter, Prop. 2.1 and Prop. 2.5 (which is proved using a stochastic shortest path argument). Complete the details of the following proof:

For any stationary policy  $\mu$ , let  $\lambda_\mu$  be the average cost per stage as defined by Eq. (2.17), let  $\lambda = \min_\mu \lambda_\mu$ , and let  $M = \{\mu \mid \lambda_\mu = \lambda\}$ . Suppose that there is a state  $s$  that is simultaneously recurrent in the Markov chains corresponding to all  $\mu \in M$ . Similar to Section 7.4 in Vol. 1, consider an associated stochastic shortest path problem with states  $1, 2, \dots, n$  and an artificial termination state  $t$  to which we move from state  $i$  with transition probability  $p_{is}(u)$ . The stage costs in this problem are  $g(i, u) - \lambda$  for  $i = 1, \dots, n$ , and the transition probabilities from a state  $i$  to a state  $j \neq s$  are the same as those of the original problem, while  $p_{js}(u)$  is zero. Show that in this stochastic shortest path problem, every improper policy has infinite cost for some initial state, and use this fact to conclude that if  $h(i)$  is the optimal cost starting at state  $i = 1, \dots, n$ , then  $\lambda$  and  $h$  satisfy  $\lambda c + h = Th$ . If there is no state  $s$  that is simultaneously recurrent for all  $\mu \in M$ , select a  $\bar{\mu} \in M$  such that there is no  $\mu \in M$  whose recurrent class is a strict subset of the recurrent class of  $\bar{\mu}$  (it is sufficient that  $\bar{\mu}$  has minimal number of recurrent states over all  $\mu \in M$ ), change the stage cost of all states  $i$  that are not recurrent under  $\bar{\mu}$  to  $g(i, u) + \epsilon$ , where  $\epsilon > 0$ , use as state  $s$  in the preceding argument any state that is recurrent under  $\bar{\mu}$ , and take  $\epsilon \rightarrow 0$ .

### 4.14 (Stochastic Shortest Path Solution Method)

The purpose of this exercise is to show how the average cost problem can be solved by solving a finite sequence of stochastic shortest path problems. As in Section 7.4 of Vol. 1, we assume that a special state, by convention state  $n$ , is recurrent in the Markov chain corresponding to each stationary policy. For a stationary policy  $\mu$ , let

$C_\mu(i)$ : expected cost starting from  $i$  up to the first visit to  $n$ ,

$N_\mu(i)$ : expected number of stages starting from  $i$  up to the first visit to  $n$ .

The proof of Prop. 2.5 shows that  $\lambda_\mu = C_\mu(n)/N_\mu(n)$ . Let  $\lambda^* = \min_\mu \lambda_\mu$  be the corresponding optimal cost.

Consider the stochastic shortest path problem obtained by leaving unchanged all transition probabilities  $p_{ij}(u)$  for  $j \neq n$ , by setting all transition probabilities  $p_{nn}(u)$  to 0, and by introducing an artificial termination state  $t$  to which we move from each state  $i$  with probability  $p_{in}(u)$ . The expected stage

cost at state  $i$  is  $g(i, u) - \lambda$ , where  $\lambda$  is a scalar parameter. Let  $h_{\mu, \lambda}(i)$  be the cost of stationary policy  $\mu$  for this stochastic shortest path problem, starting from state  $i$ , and let  $h_\lambda(i) = \min_\mu h_{\mu, \lambda}(i)$  be the corresponding optimal cost.

(a) Show that for all scalars  $\lambda$  and  $\lambda'$ , we have

$$h_{\mu, \lambda}(i) = h_{\mu, \lambda'}(i) + (\lambda' - \lambda)N_\mu(i), \quad i = 1, \dots, n.$$

(b) Define

$$h_\lambda(i) = \min_\mu h_{\mu, \lambda}(i), \quad i = 1, \dots, n.$$

Show that  $h_\lambda(i)$  is concave, monotonically decreasing, and piecewise linear as a function of  $\lambda$ , and that

$$h_\lambda(n) = 0 \quad \text{if and only if} \quad \lambda = \lambda^*.$$

Figure 4.5.1 illustrates these relations.

(c) Consider a one-dimensional search procedure that finds a zero of the function  $h_\lambda(n)$  of  $\lambda$ . This procedure brackets  $\lambda^*$  from above and below, and is illustrated in Fig. 4.5.2. Show that this procedure solves the average cost problem by solving a finite number of stochastic shortest path problems.

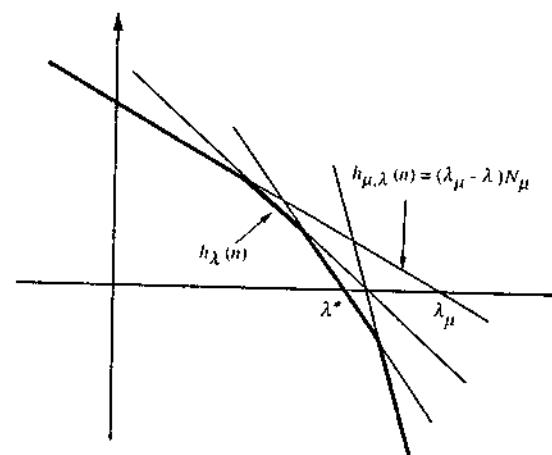
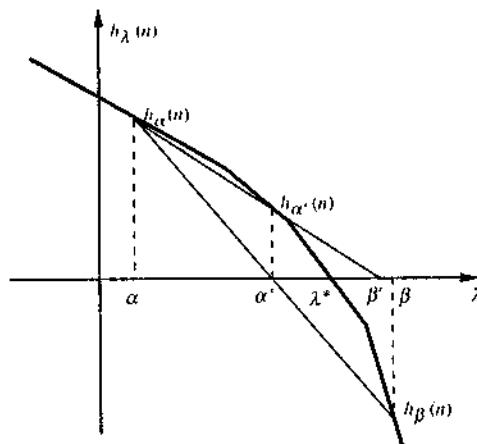


Figure 4.5.1 Relation of the costs of stationary policies in the average cost problem and the associated stochastic shortest path problem.



**Figure 4.5.2** One dimensional iterative search procedure to find  $\lambda$  such that  $h_\lambda(n) = 0$  [cf. Exercise 4.14(c)]. Each value  $h_\lambda(n)$  is obtained by solving the associated stochastic shortest path problem with stage cost  $g(i, u) - \lambda$ . At the start of the typical iteration, we have scalars  $\alpha$  and  $\beta$  such that  $\alpha < \lambda^* < \beta$ , together with the corresponding nonzero values  $h_\alpha(n)$  and  $h_\beta(n)$ . We find  $\alpha'$  such that

$$\frac{\alpha' - \alpha}{\beta' - \alpha} = \frac{h_{\alpha'}(n)}{h_{\beta}(n)},$$

and we calculate  $h_{\alpha'}(n)$ . Let  $\beta'$  be such that

$$\frac{\beta' - \alpha'}{\beta' - \alpha} = \frac{h_{\alpha'}(n)}{h_{\alpha}(n)}.$$

We then replace  $\alpha$  by  $\alpha'$ , and if  $\beta' < \beta$ , we also calculate  $h_{\beta'}(n)$  and we replace  $\beta$  by  $\beta'$ . We then perform another iteration. The algorithm stops if either  $h_{\alpha}(n) = 0$  or  $h_{\beta}(n) = 0$ .

#### 4.15 (Stochastic Shortest Path-Based Value Iteration [Ber95c])

The purpose of this exercise is to provide a value iteration method for average cost problems, which is based on the connection with the stochastic shortest path problem. Let the assumptions of Exercise 4.14 hold. Consider the algorithm

$$h^{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^{n-1} p_{ij}(u) h^k(j) \right] - \lambda^k, \quad i = 1, \dots, n,$$

$$\lambda^{k+1} = \lambda^k + \delta^k h^{k+1}(n),$$

where  $n$  is the special state that is recurrent for all unichain policies and  $\delta^k$  is a positive stepsize.

- (a) Interpret the algorithm as a value iteration algorithm for a slowly varying stochastic shortest path problem of the type considered in Exercise 4.14. Given that, for small  $\delta$ , the iteration of  $h$  is faster than the iteration of  $\lambda$ , speculate on the convergence properties of the algorithm. [It can be proved that there exists a positive constant  $\tilde{\delta}$  such that we have  $h^k(n) \rightarrow 0$  and  $\lambda^k \rightarrow \lambda^*$  if  $\underline{\delta} \leq \delta^k \leq \tilde{\delta}$ , where  $\underline{\delta}$  is some positive constant. Another interesting possibility for which convergence can be proved is to select  $\delta^k$  as a constant divided by 1 plus the number of times that  $h^k(n)$  has changed sign.]

- (b) Use the error bounds of Prop. 3.2 to justify the iteration

$$h^{k+1}(i) = \min_{u \in U(i)} \left[ g(i, u) + \sum_{j=1}^{n-1} p_{ij}(u) h^k(j) \right] - \lambda^k, \quad i = 1, \dots, n,$$

$$\lambda^{k+1} = [\lambda^k + \delta^k h^{k+1}(n)]^+,$$

where  $[c]^+$  denotes the projection of a scalar  $c$  on the interval

$$\left[ \max_{m=0, \dots, k} \beta^m, \min_{m=0, \dots, k} \bar{\beta}^m \right],$$

with

$$\underline{\beta}^k = \lambda^k + \min \left[ \min_{i \neq n} [h^{k+1}(i) - h^k(i)], h^{k+1}(n) \right],$$

$$\bar{\beta}^k = \lambda^k + \max \left[ \max_{i \neq n} [h^{k+1}(i) - h^k(i)], h^{k+1}(n) \right].$$

#### 4.16 (*Q*-Learning Based on Stochastic Shortest Paths)

The purpose of this exercise is to provide a *Q*-learning method for average cost problems, which is based on the value iteration method of Exercise 4.15. Let the assumptions of Exercise 4.14 hold. Speculate on the convergence properties of the following *Q*-learning algorithm

$$Q(i, u) := Q(i, u) + \gamma \left( g(i, u, j) + \min_{u' \in U(j)} \hat{Q}(j, u') - Q(i, u) \right) - \lambda,$$

$$i = 1, \dots, n, \quad u \in U(i),$$

$$\lambda := \lambda + \delta \min_{u' \in U(n)} Q(n, u'),$$

where

$$\hat{Q}(j, u') = \begin{cases} Q(j, u') & \text{if } j \neq n, \\ 0 & \text{otherwise,} \end{cases}$$

and  $j$  and  $g(i, u, j)$  are generated from the pair  $(i, u)$  by simulation. Here the stepsizes  $\gamma$  and  $\delta$  should be diminishing, but  $\delta$  should diminish "faster" than  $\gamma$  [for example  $\gamma = c_1/k$  and  $\delta = c_2/k \log k$ , where  $c_1$  and  $c_2$  are positive constants and  $k$  is the number of iterations performed on the corresponding pair  $(i, u)$  or  $\lambda$ ].

*Continuous-Time Problems***Contents**

5.1. Uniformization . . . . .	p. 242
5.2. Queueing Applications . . . . .	p. 250
5.3. Semi-Markov Problems . . . . .	p. 261
5.4. Notes, Sources, and Exercises . . . . .	p. 273

We have considered so far problems where the cost per stage does not depend on the time required for transition from one state to the next. Such problems have a natural discrete-time representation. On the other hand, there are situations where controls are applied at discrete times but cost is continuously accumulated. Furthermore, the time between successive control choices is variable; it may be random or it may depend on the current state and the choice of control. For example, in queueing systems state transitions correspond to arrivals or departures of customers, and the corresponding times of transition are random. This chapter primarily discusses problems of this type. We restrict attention to continuous-time systems with a finite or countable number of states. Many of the practical systems of this type involve the Poisson process, so for many of the examples discussed, we assume that the reader is familiar with this process at the level of textbooks such as [Ros83b] and [Gal95].

In Section 5.1, we concentrate on an important class of continuous-time optimization models of the discounted type, where the times between successive transitions have an *exponential probability distribution*. We show that by using a conversion process called *uniformization*, discounted versions of these models can be analyzed within the discrete-time framework discussed up to now.

In Section 5.2, we discuss applications of uniformization. We concentrate on queueing models arising in various communications and scheduling contexts.

In Section 5.3, we discuss more general continuous-time models, called *semi-Markov problems*, where the times between successive transitions need not have an exponential distribution.

## 5.1 UNIFORMIZATION

In this chapter, we restrict ourselves to continuous-time systems with a finite or a countable number of states. Here state transitions and control selections take place at discrete times, but the time from one transition to the next is random. In this section, we assume that:

1. If the system is in state  $i$  and control  $u$  is applied, the next state will be  $j$  with probability  $p_{ij}(u)$ .
2. The time interval  $\tau$  between the transition to state  $i$  and the transition to the next state is exponentially distributed with parameter  $\nu_i(u)$ ; that is,

$$P\{\text{transition time interval } \leq \tau \mid i, u\} \leq 1 - e^{-\nu_i(u)\tau},$$

or equivalently, the probability density function of  $\tau$  is

$$p(\tau) = \nu_i(u)e^{-\nu_i(u)\tau}, \quad \tau \geq 0.$$

Furthermore,  $\tau$  is independent of earlier transition times, states, and controls. The parameters  $\nu_i(u)$  are uniformly bounded in the sense that for some  $\nu$  we have

$$\nu_i(u) \leq \nu, \quad \text{for all } i, u \in U(i).$$

The parameter  $\nu_i(u)$  is referred to as the *rate of transition* associated with state  $i$  and control  $u$ . It can be verified that the corresponding average transition time is

$$E\{\tau\} = \int_0^\infty \tau \nu_i(u) e^{-\nu_i(u)\tau} d\tau = \frac{1}{\nu_i(u)},$$

so  $\nu_i(u)$  can be interpreted as the average number of transitions per unit time.

The state and control at any time  $t$  are denoted by  $x(t)$  and  $u(t)$ , respectively, and stay constant between transitions. We use the following notation:

$t_k$ : The time of occurrence of the  $k$ th transition. By convention, we denote  $t_0 = 0$ .

$\tau_k = t_k - t_{k-1}$ : The  $k$ th transition time interval.

$x_k = x(t_k)$ : We have  $x(t) = x_k$  for  $t_k \leq t < t_{k+1}$ .

$u_k = u(t_k)$ : We have  $u(t) = u_k$  for  $t_k \leq t < t_{k+1}$ .

We consider a cost function of the form

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} g(x(t), u(t)) dt \right\}, \quad (1.1)$$

where  $g$  is a given function and  $\beta$  is a given positive discount parameter. Similar to discrete-time problems, an admissible policy is a sequence  $\pi = \{\mu_0, \mu_1, \dots\}$ , where each  $\mu_k$  is a function mapping states to controls with  $\mu_k(i) \in U(i)$  for all states  $i$ . Under  $\pi$ , the control applied in the interval  $[t_k, t_{k+1})$  is  $\mu_k(x_k)$ . Because states stay constant between transitions, the cost function of  $\pi$  is given by

$$J_\pi(x_0) = \sum_{k=0}^{\infty} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} g(x_k, \mu_k(x_k)) \Big| x_0 \right\}.$$

We first consider the case where *the rate of transition is the same for all states and controls*; that is,

$$\nu_i(u) = \nu, \quad \text{for all } i, u.$$

A little thought shows that the problem is then essentially the same as the one where transition times are fixed, because the control cannot influence the cost of a stage by affecting the length of the next transition time interval.

Indeed, the cost (1.1) corresponding to a sequence  $\{(x_k, u_k)\}$  can be expressed as

$$\sum_{k=0}^{\infty} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} g(x(t), u(t)) dt \right\} = \sum_{k=0}^{\infty} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} dt \right\} E \{g(x_k, u_k)\} \quad (1.2)$$

We have (using the independence of the transition time intervals)

$$\begin{aligned} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} dt \right\} &= \frac{E\{e^{-\beta t_k}\}(1 - E\{e^{-\beta t_{k+1}}\})}{\beta} \\ &= \frac{E\{e^{-\beta(\tau_1 + \dots + \tau_k)}\}(1 - E\{e^{-\beta\tau_{k+1}}\})}{\beta} \\ &= \frac{\alpha^k(1 - \alpha)}{\beta}, \end{aligned} \quad (1.3)$$

where

$$\alpha = E\{e^{-\beta\tau}\} = \int_0^\infty e^{-\beta\tau} \nu e^{-\nu\tau} d\tau = \frac{\nu}{\beta + \nu}.$$

The above expression for  $\alpha$  yields  $(1 - \alpha)/\beta = 1/(\beta + \nu)$ , so that from Eq. (1.3), we have

$$E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} dt \right\} = \frac{\alpha^k}{\beta + \nu}.$$

From this equation together with Eq. (1.2) it follows that the cost of the problem can be expressed as

$$\frac{1}{\beta + \nu} \sum_{k=0}^{\infty} \alpha^k E \{g(x_k, u_k)\}.$$

Thus we are faced in effect with an ordinary discrete-time problem where expected total cost is to be minimized. The effect of randomness of the transition times has been simply to appropriately scale the cost per stage.

To summarize, a continuous-time Markov chain problem with cost

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} g(x(t), u(t)) dt \right\}$$

and rate of transition  $\nu$  that is independent of state and control is equivalent to a discrete-time Markov chain problem with discount factor

$$\alpha = \frac{\nu}{\beta + \nu}, \quad (1.4)$$

and cost per stage given by

$$\hat{g}(i, u) = \frac{g(i, u)}{\beta + \nu}. \quad (1.5)$$

In particular, Bellman's equation takes the form

$$J(i) = \min_{u \in U(i)} \left[ \frac{g(i, u)}{\beta + \nu} + \alpha \sum_j p_{ij}(u) J(j) \right], \quad (1.6)$$

or equivalently,

$$J(i) = \frac{1}{\beta + \nu} \min_{u \in U(i)} \left[ g(i, u) + \nu \sum_j p_{ij}(u) J(j) \right]. \quad (1.7)$$

In some problems, in addition to the cost (1.1), there is an extra expected stage cost  $\hat{g}(i, u)$  that is incurred at the time the control  $u$  is chosen at state  $i$ , and is independent of the length of the transition interval. In that case the expected stage cost (1.5) should be changed to

$$\hat{g}(i, u) + \frac{g(i, u)}{\beta + \nu}, \quad (1.8)$$

and Bellman's equation (1.6) becomes

$$J(i) = \min_{u \in U(i)} \left[ \hat{g}(i, u) + \frac{g(i, u)}{\beta + \nu} + \alpha \sum_j p_{ij}(u) J(j) \right]. \quad (1.9)$$

### Example 1.1

A manufacturer of a specialty item processes orders in batches. Orders arrive according to a Poisson process with rate  $\nu$  per unit time; that is, the successive interarrival intervals are independent and exponentially distributed with parameter  $\nu$ . For each order there is a positive cost  $c$  per unit time that the order is unfilled. Costs are discounted with a discount parameter  $\beta > 0$ . The setup cost for processing the orders is  $K$ . Upon arrival of a new order, the manufacturer must decide whether to process the current batch or to wait for the next order.

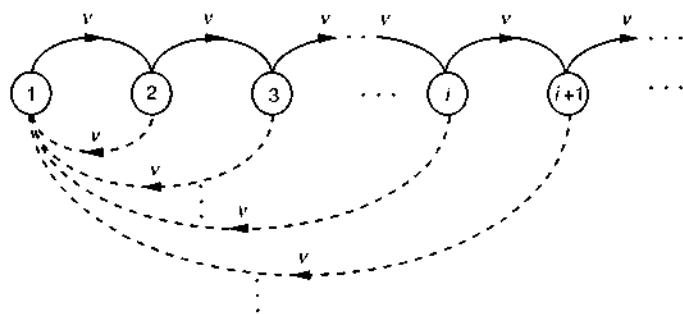
Here the state is the number  $i$  of unfilled orders. If the decision to fill the orders at state  $i$  is made, the cost is  $K$  and the next transition will be to state 1. Otherwise, there will be an average cost  $(ci)/(\beta + \nu)$  up to the transition to the next state  $i + 1$  [cf. Eq. (1.5)], as shown in Fig. 5.1.1. [Note that the setup cost  $K$  is incurred immediately after a decision to process the orders is made, so  $K$  is not discounted over the time interval up to the next

transition; cf. Eq. (1.9).] We are in effect faced with a discounted discrete-time problem with positive but unbounded cost per stage. (We could also consider an alternative model where an upper bound is placed on the number of unfilled orders. We would then have a discounted discrete-time problem with bounded cost per stage.)

Since Assumption P is satisfied (cf. Section 3.1), Bellman's equation holds and takes the form

$$J(i) = \min \left[ K + \alpha J(1), \frac{\alpha i}{\beta + \nu} + \alpha J(i+1) \right], \quad i = 1, 2, \dots, \quad (1.10)$$

where  $\alpha = \nu/(\beta + \nu)$  is the effective discount factor [cf. Eq. (1.1)]. Reasoning from first principles, we see that  $J(i)$  is a monotonically nondecreasing function of  $i$ , so from Bellman's equation it follows that there exists a threshold  $i^*$  such that it is optimal to process the orders if and only if their number exceeds  $i^*$ .



**Figure 5.1.1** Transition diagram for the continuous-time Markov chain of Example 1.1. The transitions associated with the first control (do not fill the orders) are shown with solid lines, and the transitions associated with the second control (fill the orders) are shown with broken lines.

### Nonuniform Transition Rates

We now argue that the more general case where the transition rate  $\nu_i(u)$  depends on the state and the control can be converted to the previous case of uniform transition rate by using the trick of *allowing fictitious transitions from a state to itself*. Roughly, transitions that are slow on the average are speeded up with the understanding that sometimes after a transition the state may stay unchanged. To see how this works, let  $\nu$  be a new uniform transition rate with  $\nu_i(u) \leq \nu$  for all  $i$  and  $u$ , and define new

transition probabilities by

$$\hat{p}_{ij}(u) = \begin{cases} \frac{\nu_i(u)}{\nu} p_{ij}(u) & \text{if } i \neq j, \\ \frac{\nu_i(u)}{\nu} p_{ii}(u) + 1 - \frac{\nu_i(u)}{\nu} & \text{if } i = j. \end{cases}$$

We refer to this process as the *uniform* version of the original (see Fig. 5.1.2). We argue now that leaving state  $i$  at a rate  $\nu_i(u)$  in the original process is statistically identical to leaving state  $i$  at the faster rate  $\nu$ , but returning back to  $i$  with probability  $1 - \nu_i(u)/\nu$  in the new process. Equivalently, transitions are real (lead to a different state) with probability  $\nu_i(u)/\nu < 1$ . By statistical equivalence, we mean that, for any given policy  $\pi$ , initial state  $x_0$ , time  $t$ , and state  $i$ , the probability  $P\{x(t) = i \mid x_0, \pi\}$  is identical for the original process and its uniform version. We give a proof of this fact in Exercise 5.1 for the case of a finite number of states (see also [Lip75], [Ser79], and [Ros83b] for further discussion).

To summarize, we can convert a continuous-time Markov chain problem with transition rates  $\nu_i(u)$ , transition probabilities  $p_{ij}(u)$ , and cost

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} g(x(t), u(t)) dt \right\},$$

into a discrete-time Markov chain problem with discount factor

$$\alpha = \frac{\nu}{\beta + \nu}, \quad (1.11)$$

where  $\nu$  is a uniform transition rate chosen so that

$$\nu_i(u) \leq \nu, \quad \text{for all } i, u. \quad (1.12)$$

The transition probabilities are

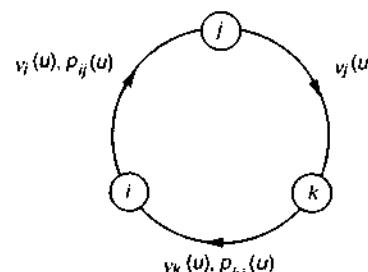
$$\hat{p}_{ij}(u) = \begin{cases} \frac{\nu_i(u)}{\nu} p_{ij}(u) & \text{if } i \neq j, \\ \frac{\nu_i(u)}{\nu} p_{ii}(u) + 1 - \frac{\nu_i(u)}{\nu} & \text{if } i = j, \end{cases} \quad (1.13)$$

and the cost per stage is

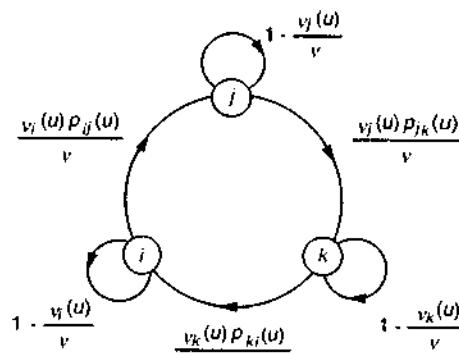
$$\hat{g}(i, u) = \frac{g(i, u)}{\beta + \nu}, \quad \text{for all } i, u.$$

In particular, Bellman's equation takes the form

$$J(i) = \min_{u \in U(i)} \left[ \hat{g}(i, u) + \alpha \sum_j \hat{p}_{ij}(u) J(j) \right].$$



Transition rates and probabilities  
for continuous-time chain



Transition probabilities for uniform version

**Figure 5.1.2** Transforming a continuous-time Markov chain into its uniform version through the use of fictitious self-transitions. The uniform version has a uniform transition rate  $\nu$ , which is an upper bound for all transition rates  $\nu_i(u)$  of the original, and transition probabilities  $\bar{p}_{ij}(u) = (\nu_i(u)/\nu)p_{ij}(u)$  for  $i \neq j$ , and  $\bar{p}_{ii}(u) = (\nu_i(u)/\nu)p_{ii}(u) + 1 - \nu_i(u)/\nu$  for  $j = i$ . In the example of the figure we have  $p_{ii}(u) = 0$  for all  $i$  and  $u$ .

which, after some calculation using the preceding definitions, can be written as

$$J(i) = \frac{1}{\beta + \nu} \min_{u \in U(i)} \left[ g(i, u) + (\nu - \nu_i(u))J(i) + \nu_i(u) \sum_j p_{ij}(u)J(j) \right]. \quad (1.14)$$

In the case where there is an extra expected stage cost  $\hat{g}(i, u)$  that is incurred at the time the control  $u$  is chosen at state  $i$ , Bellman's equation

becomes [cf. Eq. (1.9)]

$$\begin{aligned} J(i) = \frac{1}{\beta + \nu} \min_{u \in U(i)} & \left[ (\beta + \nu)\hat{g}(i, u) + g(i, u) \right. \\ & \left. + (\nu - \nu_i(u))J(i) + \nu_i(u) \sum_j p_{ij}(u)J(j) \right]. \end{aligned} \quad (1.15)$$

### Undiscounted and Average Cost Problems

When the discount parameter  $\beta$  is zero in the preceding problem formulation, we obtain a continuous-time version of the undiscounted cost problem considered in Chapter 3. If in addition, the number of states is finite and there is a cost-free and absorbing state, we obtain a continuous-time analog of the stochastic shortest path problem considered of Chapter 2. However, when  $\beta = 0$ , it is unnecessary to resort to uniformization. It can be seen that the problem is essentially the same as the discrete-time problem with the same transition probabilities but where the average transition cost at state  $i$  under  $u$  is the average cost per unit time  $g(i, u)$  multiplied with the expected length  $1/\nu_i(u)$  of the transition interval. Thus Bellman's equation has the form

$$J(i) = \min_{u \in U(i)} \left[ \frac{g(i, u)}{\nu_i(u)} + \sum_j p_{ij}(u)J(j) \right]. \quad (1.16)$$

After some calculation, it can be seen that the above equation can also be obtained from Eq. (1.14) by setting  $\beta = 0$ .

In fact for undiscounted problems, the preceding argument does not depend on the character of the probability distributions of the transition times. Regardless of whether these distributions are exponential or not, one simply needs to multiply  $g(i, u)$  with the average transition time corresponding to  $(i, u)$  and then treat the problem as if it were a discrete-time problem.

There is also a continuous-time version of the average cost per stage problem of Chapter 4. The cost function has the form

$$\lim_{N \rightarrow \infty} \frac{1}{E\{t_N\}} E \left\{ \int_0^{t_N} g(x(t), u(t)) dt \right\}.$$

We will consider this problem in Section 5.3 in a more general context where the probability distributions of the transition times need not be exponential.

## 5.2 QUEUEING APPLICATIONS

We now illustrate the theory of the preceding section through some applications involving the control of queues.

### Example 2.1 (M/M/1 Queue with Controlled Service Rate)

Consider a single-server queueing system where customers arrive according to a Poisson process with rate  $\lambda$ . The service time of a customer is exponentially distributed with parameter  $\mu$  (called the service rate). Service times of customers are independent and are also independent of customer interarrival times. The service rate  $\mu$  can be selected from a closed subset  $M$  of an interval  $[0, \bar{\mu}]$  and can be changed at the times when a customer arrives or when a customer departs from the system. There is a cost  $q(\mu)$  per unit time for using rate  $\mu$  and a waiting cost  $c(i)$  per unit time when there are  $i$  customers in the system (waiting in queue or undergoing service). The idea is that one should be able to cut down on the customer waiting costs by choosing a faster service rate, which presumably costs more. The problem, roughly, is to select the service rate so that the service cost is optimally traded off with the customer waiting cost.

We assume the following:

1. For some  $\mu \in M$  we have  $\mu > \lambda$ . (In words, there is available a service rate that is fast enough to keep up with the arrival rate, thereby maintaining the queue length bounded.)
2. The waiting cost function  $c$  is nonnegative, monotonically nondecreasing, and "convex" in the sense

$$c(i+2) - c(i+1) \geq c(i+1) - c(i), \quad i = 0, 1, \dots$$

3. The service rate cost function  $q$  is nonnegative, and continuous on  $[0, \bar{\mu}]$ , with  $q(0) = 0$ .

The problem fits the framework of this section. The state is the number of customers in the system, and the control is the choice of service rate following a customer arrival or departure. The transition rate at state  $i$  is

$$\nu_i(\mu) = \begin{cases} \lambda & \text{if } i = 0, \\ \lambda + \mu & \text{if } i \geq 1. \end{cases}$$

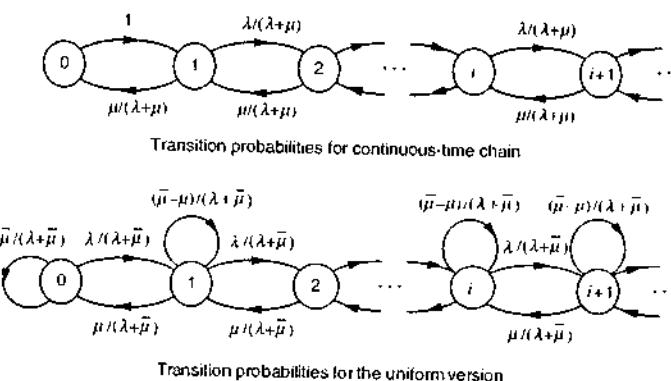
The transition probabilities of the Markov chain and its uniform version for the choice

$$\nu = \lambda + \bar{\mu}$$

are shown in Fig. 5.2.1.

The effective discount factor is

$$\alpha = \frac{\nu}{\beta + \nu}$$



**Figure 5.2.1** Continuous-time Markov chain and uniform version for Example 2.1 when the service rate is equal to  $\mu$ . The transition rates of the original Markov chain are  $\nu_i(\mu) = \lambda + \mu$  for states  $i \geq 1$ , and  $\nu_0(\mu) = \lambda$  for state 0. The transition rate for the uniform version is  $\nu = \lambda + \bar{\mu}$ .

and the cost per stage is

$$\frac{1}{\beta + \nu} (c(i) + q(\mu)).$$

The form of Bellman's equation is [cf. Eq. (1.14)]

$$J(0) = \frac{1}{\beta + \nu} (c(0) + (\nu - \lambda)J(0) + \lambda J(1))$$

and for  $i = 1, 2, \dots$ ,

$$J(i) = \frac{1}{\beta + \nu} \min_{\mu \in M} [c(i) + q(\mu) + \mu J(i-1) + (\nu - \lambda - \mu)J(i) + \lambda J(i+1)]. \quad (2.1)$$

An optimal policy is to use at state  $i$  the service rate that minimizes the expression on the right. Thus it is optimal to use at state  $i$  the service rate

$$\mu^*(i) = \arg \min_{\mu \in M} \{q(\mu) - \mu \Delta(i)\}, \quad (2.2)$$

where  $\Delta(i)$  is the differential of the optimal cost

$$\Delta(i) = J(i) - J(i-1), \quad i = 1, 2, \dots$$

[When the minimum in Eq. (2.2) is attained by more than one service rate  $\mu$  we choose by convention the smallest.] We will demonstrate shortly that  $\Delta(i)$  is monotonically nondecreasing. It will then follow from Eq. (2.2) (see

Fig. 5.2.2) that the optimal service rate  $\mu^*(i)$  is monotonically nondecreasing. Thus, as the queue length increases, it is optimal to use a faster service rate.

To show that  $\Delta(i)$  is monotonically nondecreasing, we use the value iteration method to generate a sequence of functions  $J_k$  from the starting function

$$J_0(i) = 0, \quad i = 0, 1, \dots$$

For  $k = 0, 1, \dots$ , [cf. Eq. (2.1)], we have

$$J_{k+1}(0) = \frac{1}{\beta + \nu} (c(0) + (\nu - \lambda) J_k(0) + \lambda J_k(1)),$$

and for  $i = 1, 2, \dots$ ,

$$J_{k+1}(i) = \frac{1}{\beta + \nu} \min_{\mu \in M} [c(i) + q(\mu) + \mu J_k(i-1) + (\nu - \lambda - \mu) J_k(i) + \lambda J_k(i+1)]. \quad (2.3)$$

For  $k = 0, 1, \dots$  and  $i = 1, 2, \dots$ , let

$$\Delta_k(i) = J_k(i) - J_k(i-1).$$

For completeness of notation, define also  $\Delta_k(0) = 0$ . From the theory of Section 3.1 (see Prop. 1.7 of that section), we have  $J_k(i) \rightarrow J(i)$  as  $k \rightarrow \infty$ . It follows that we have

$$\lim_{k \rightarrow \infty} \Delta_k(i) = \Delta(i), \quad i = 1, 2, \dots$$

Therefore, it will suffice to show that  $\Delta_k(i)$  is monotonically nondecreasing for every  $k$ . For this we use induction. The assertion is trivially true for  $k = 0$ . Assuming that  $\Delta_k(i)$  is monotonically nondecreasing, we show that the same is true for  $\Delta_{k+1}(i)$ . Let

$$\mu^k(0) = 0,$$

$$\mu^k(i) = \arg \min_{\mu \in M} [q(\mu) - \mu \Delta_k(i)], \quad i = 1, 2, \dots$$

From Eq. (2.3) we have, for all  $i = 0, 1, \dots$ ,

$$\begin{aligned} \Delta_{k+1}(i+1) &= J_{k+1}(i+1) - J_{k+1}(i) \\ &= \frac{1}{\beta + \nu} (c(i+1) + q(\mu^k(i+1)) + \mu^k(i+1) J_k(i) \\ &\quad + (\nu - \lambda - \mu^k(i+1)) J_k(i+1) \\ &\quad - \lambda J_k(i+2) - c(i) - q(\mu^k(i+1)) - \mu^k(i+1) J_k(i-1) \quad (2.4) \\ &\quad - (\nu - \lambda - \mu^k(i+1)) J_k(i) - \lambda J_k(i+1)) \\ &= \frac{1}{\beta + \nu} (c(i+1) - c(i) + \lambda \Delta_k(i+2) + (\nu - \lambda) \Delta_k(i+1) \\ &\quad - \mu^k(i+1) (\Delta_k(i+1) - \Delta_k(i))). \end{aligned}$$

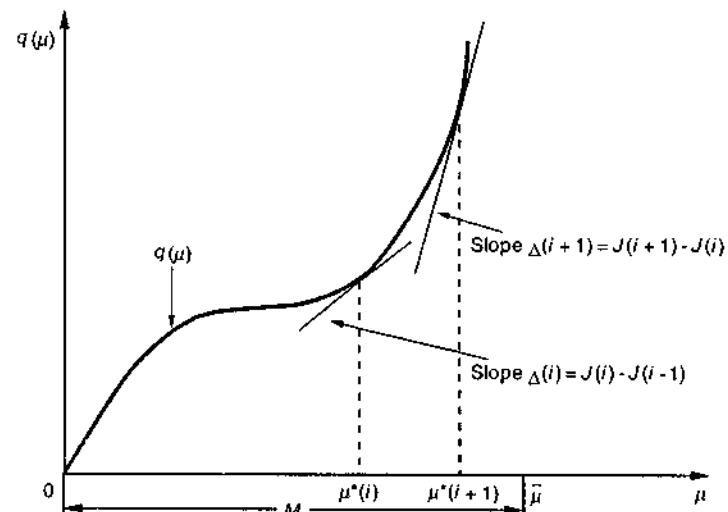
Similarly, we obtain, for  $i = 1, 2, \dots$

$$\begin{aligned} \Delta_{k+1}(i) &\leq \frac{1}{\beta + \nu} (c(i) - c(i-1) + \lambda \Delta_k(i+1) + (\nu - \lambda) \Delta_k(i) \\ &\quad - \mu^k(i-1) (\Delta_k(i) - \Delta_k(i-1))). \end{aligned}$$

Subtracting the last two inequalities, we obtain, for  $i = 1, 2, \dots$ ,

$$\begin{aligned} (\beta + \nu) (\Delta_{k+1}(i+1) - \Delta_{k+1}(i)) &\geq (c(i+1) - c(i)) - (c(i) - c(i-1)) \\ &\quad + \lambda (\Delta_k(i+2) - \Delta_k(i+1)) \\ &\quad + (\nu - \lambda - \mu^k(i+1)) (\Delta_k(i+1) - \Delta_k(i)) \\ &\quad + \mu^k(i-1) (\Delta_k(i) - \Delta_k(i-1)). \end{aligned}$$

Using our convexity assumption on  $c(i)$ , the fact  $\nu - \lambda - \mu^k(i+1) = \bar{\mu} - \mu^k(i+1) \geq 0$ , and the induction hypothesis, we see that every term on the right-hand side of the preceding inequality is nonnegative. Therefore,  $\Delta_{k+1}(i+1) \geq \Delta_{k+1}(i)$  for  $i = 1, 2, \dots$  From Eq. (2.4) we can also show that  $\Delta_{k+1}(1) \geq 0 = \Delta_{k+1}(0)$ , and the induction proof is complete.



**Figure 5.2.2** Determining the optimal service rate at states  $i$  and  $(i+1)$  in Example 2.1. The optimal service rate  $\mu^*(i)$  tends to increase as the system becomes more crowded ( $i$  increases).

To summarize, the optimal service rate  $\mu^*(i)$  is given by Eq. (2.2) and tends to become faster as the system becomes more crowded ( $i$  increases).

### Example 2.2 (M/M/1 Queue with Controlled Arrival Rate)

Consider the same queueing system as in the previous example with the difference that the service rate  $\mu$  is fixed, but the arrival rate  $\lambda$  can be controlled. We assume that  $\lambda$  is chosen from a closed subset  $\Lambda$  of an interval  $[0, \bar{\lambda}]$ , and there is a cost  $q(\lambda)$  per unit time. All other assumptions of Example 2.1 are also in effect. What we have here is a problem of flow control, whereby we want to trade off optimally the cost for throttling the arrival process with the customer waiting cost.

This problem is very similar to the one of Example 2.1. We choose as uniform transition rate

$$\nu = \bar{\lambda} + \mu$$

and construct the uniform version of the Markov chain. Bellman's equation takes the form

$$\begin{aligned} J(0) &= \frac{1}{\beta + \nu} \min_{\lambda \in \Lambda} [c(0) + q(\lambda) + (\nu - \lambda)J(0) + \lambda J(1)], \\ J(i) &= \frac{1}{\beta + \nu} \min_{\lambda \in \Lambda} [c(i) + q(\lambda) + \mu J(i-1) + (\nu - \lambda - \mu)J(i) + \lambda J(i+1)]. \end{aligned}$$

An optimal policy is to use at state  $i$  the arrival rate

$$\lambda^*(i) = \arg \min_{\lambda \in \Lambda} [q(\lambda) + \lambda \Delta(i+1)], \quad (2.5)$$

where, as before,  $\Delta(i)$  is the differential of the optimal cost

$$\Delta(i) = J(i) - J(i-1), \quad i = 1, 2, \dots$$

As in Example 2.1, we can show that  $\Delta(i)$  is monotonically nondecreasing; so from Eq. (2.5) we see that *the optimal arrival rate tends to decrease as the system becomes more crowded* ( $i$  increases).

### Example 2.3 (Priority Assignment and the $\mu c$ Rule)

Consider  $n$  queues that share a single server. There is a positive cost  $c_i$  per unit time and per customer in each queue  $i$ . The service time of a customer of queue  $i$  is exponentially distributed with parameter  $\mu_i$ , and all customer service times are independent. Assuming that we start with a given number of customers in each queue and no further arrivals occur, what is the optimal order for serving the customers? The cost here is

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} \sum_{i=1}^n c_i x_i(t) dt \right\},$$

where  $x_i(t)$  is the number of customers in the  $i$ th queue at time  $t$ , and  $\beta$  is a positive discount parameter.

We first construct the uniform version of the problem. The construction is shown in Fig. 5.2.3. The discount factor is

$$\alpha = \frac{\mu}{\beta + \mu}, \quad (2.6)$$

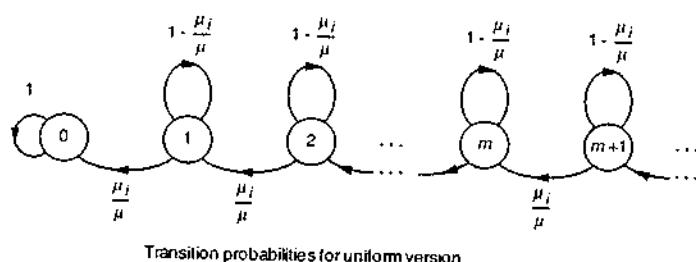
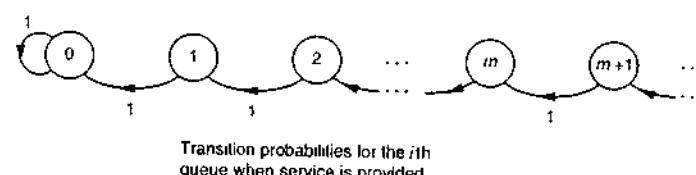
where

$$\mu = \max_i \{\mu_i\},$$

and the corresponding cost is

$$\frac{1}{\beta + \mu} \sum_{k=0}^{\infty} \alpha^k E \left\{ \sum_{i=1}^n c_i x'_k \right\}, \quad (2.7)$$

where  $x'_k$  is the number of customers in the  $i$ th queue after the  $k$ th transition (real or fictitious).



**Figure 5.2.3** Continuous-time Markov chain and uniform version for the  $i$ th queue of Example 2.3 when service is provided. The transition rate for the uniform version is  $\mu = \max_i \{\mu_i\}$ .

We now rewrite the cost in a way that is more convenient for analysis. The idea is to transform the problem from one of minimizing waiting costs to one of maximizing savings in waiting costs through customer service. For  $k = 0, 1, \dots$ , define

$$i_k = \begin{cases} i & \text{if the } k\text{th transition corresponds to a departure from queue } i, \\ 0 & \text{if the } k\text{th transition is fictitious.} \end{cases}$$

Denote also

$$c_{i0} = 0,$$

$x'_0$ : the initial number of customers in queue  $i$ .

Then the cost (2.7) can also be written as

$$\begin{aligned} & \frac{1}{\beta + \mu} \left[ \sum_{i=1}^n c_i x'_0 + \sum_{k=1}^{\infty} \alpha^k E \left\{ \sum_{i=1}^n c_i x'_0 + \sum_{m=0}^{k-1} c_{im} \right\} \right] \\ &= \frac{1}{\beta + \mu} \left[ \sum_{k=0}^{\infty} \alpha^k \left( \sum_{i=1}^n c_i x'_0 \right) - E \left\{ \sum_{m=0}^{\infty} \sum_{k=m+1}^{\infty} \alpha^k c_{im} \right\} \right] \\ &= \frac{1}{(\beta + \mu)(1 - \alpha)} \sum_{i=1}^n c_i x'_0 - \frac{\alpha}{(\beta + \mu)(1 - \alpha)} \sum_{k=0}^{\infty} \alpha^k E\{c_{ik}\} \\ &= \frac{1}{\beta} \sum_{i=1}^n c_i x'_0 - \frac{\alpha}{\beta} \sum_{k=0}^{\infty} \alpha^k E\{c_{ik}\}. \end{aligned}$$

Therefore, instead of minimizing the cost (2.7), we can equivalently

$$\text{maximize } \sum_{k=0}^{\infty} \alpha^k E\{c_{ik}\}, \quad (2.8)$$

where  $c_{ik}$  can be viewed as the *savings in waiting cost rate* obtained from the  $k$ th transition.

We now recognize problem (2.8) as a *multiarmed bandit problem*. The  $n$  queues can be viewed as separate projects. At each time, a nonempty queue, say  $i$ , is selected and served. Since a customer departure occurs with probability  $\mu_i/\mu$ , and a fictitious transition that leaves the state unchanged occurs with probability  $1 - \mu_i/\mu$ , the corresponding expected reward is

$$\frac{\mu_i}{\mu} c_i. \quad (2.9)$$

It is evident that the problem falls in the deteriorating case examined at the end of Section 1.5. Therefore, after each customer departure, it is optimal to serve the queue with maximum expected reward per stage (i.e., engage the project with maximal index; cf. the end of Section 1.5). Equivalently [cf. Eq. (2.9)], it is optimal to serve the nonempty queue  $i$  for which  $\mu_i c_i$  is maximum. This policy is known as the *μc rule*. It plays an important role in several other formulations of the priority assignment problem (see [BDM83], [Har75a], and [Har75b]). We can view  $\mu_i c_i$  as the ratio of the waiting cost rate  $c_i$  by the average time  $1/\mu_i$  needed to serve a customer. Therefore, the *μc rule* amounts to serving the queue for which the savings in waiting cost rate per unit average service time are maximized.

### Example 2.4 (Routing Policies for a Two-Station System)

Consider the system consisting of two queues shown in Fig. 5.2.4. Customers arrive according to a Poisson process with rate  $\lambda$  and are routed upon arrival to one of the two queues. Service times are independent and exponentially distributed with parameter  $\mu_1$  in the first queue and  $\mu_2$  in the second queue. The cost is

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} (c_1 x_1(t) + c_2 x_2(t)) dt \right\},$$

where  $\beta$ ,  $c_1$ , and  $c_2$  are given positive scalars, and  $x_1(t)$  and  $x_2(t)$  denote the number of customers at time  $t$  in queues 1 and 2, respectively.

As earlier, we construct the uniform version of this problem with uniform rate

$$\nu = \lambda + \mu_1 + \mu_2 \quad (2.10)$$

and the transition probabilities shown in Fig. 5.2.5. We take as state space the set of pairs  $(i, j)$  of customers in queues 1 and 2. Bellman's equation takes the form

$$\begin{aligned} J(i, j) = & \frac{1}{\beta + \nu} (c_1 i + c_2 j + \mu_1 J((i-1)^+, j) + \mu_2 J(i, (j-1)^+)) \\ & + \frac{\lambda}{\beta + \nu} \min[J(i+1, j), J(i, j+1)], \end{aligned} \quad (2.11)$$

where for any  $x$  we denote

$$(x)^+ = \max(0, x).$$

From this equation we see that an optimal policy is to route an arriving customer to queue 1 if and only if the state  $(i, j)$  at the time of arrival belongs to the set

$$S_1 = \{(i, j) \mid J(i+1, j) \leq J(i, j+1)\}. \quad (2.12)$$

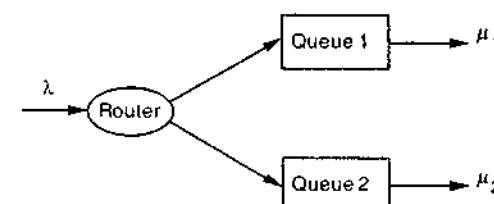
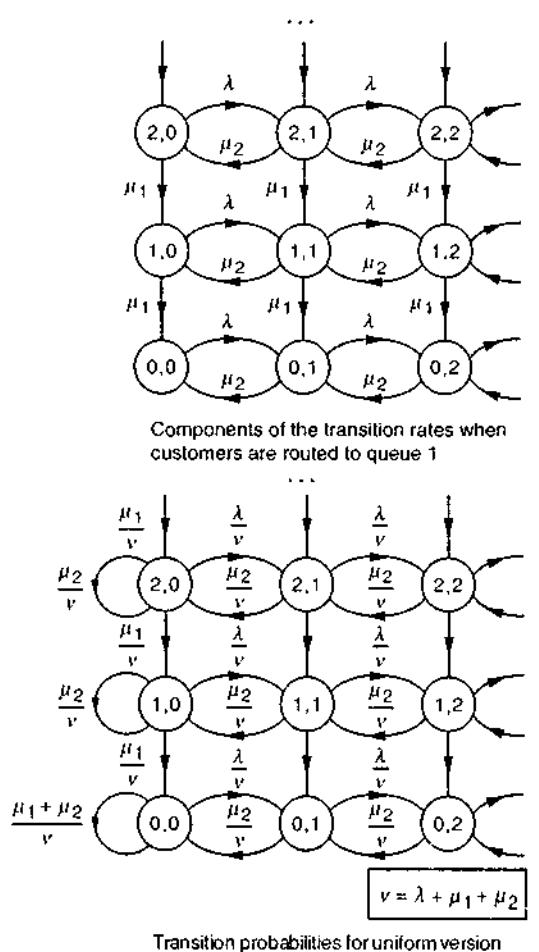


Figure 5.2.4 Queueing system of Example 2.4. The problem is to route each arriving customer to queue 1 or 2 so as to minimize the total average discounted waiting cost.



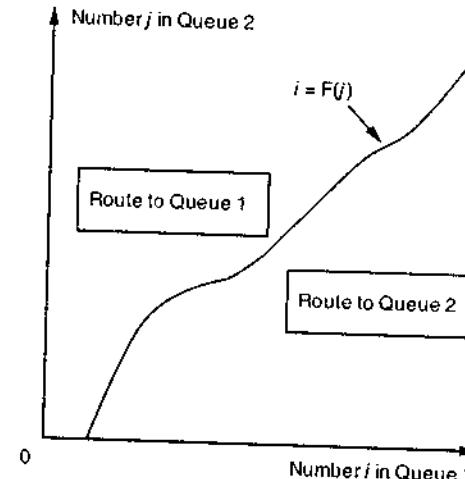
**Figure 5.2.5** Continuous-time Markov chain and uniform version for Example 2.1 when customers are routed to the first queue. The states are the pairs of customer numbers in the two queues.

This optimal policy can be characterized better by some further analysis. Intuitively, one expects that optimal routing can be achieved by sending a customer to the queue that is "less crowded" in some sense. It is therefore natural to conjecture that, if it is optimal to route to the first queue when the state is  $(i, j)$ , it must be optimal to do the same when the first queue is even less crowded; that is, the state is  $(i - m, j)$  with  $m \geq 1$ . This is equivalent

to saying that the set of states  $S_1$  for which it is optimal to route to the first queue is characterized by a monotonically nondecreasing *threshold function*  $F$  by means of

$$S_1 = \{(i, u) \mid i = F(j)\} \quad (2.13)$$

(see Fig. 5.2.6). Accordingly, we call the corresponding optimal policy a *threshold policy*.



**Figure 5.2.6** Typical threshold policy characterized by a threshold function  $F$ .

We will demonstrate the existence of a threshold optimal policy by showing that the functions

$$\Delta_1(i, j) = J(i+1, j) - J(i, j+1),$$

$$\Delta_2(i, j) = J(i, j+1) - J(i+1, j)$$

are monotonically nondecreasing in  $i$  for each fixed  $j$ , and in  $j$  for each fixed  $i$ , respectively. We will show this property for  $\Delta_1$ ; the proof for  $\Delta_2$  is analogous. It will be sufficient to show that for all  $k = 0, 1, \dots$  the functions

$$\Delta_1^k(i, j) = J_k(i+1, j) - J_k(i, j+1) \quad (2.14)$$

are monotonically nondecreasing in  $i$  for each fixed  $j$ , where  $J_k$  is generated by the value iteration method starting from the zero function; that is,  $J_{k+1}(i, j) = (TJ_k)(i, j)$ , where  $T$  is the DP mapping defining Bellman's equation (2.11) and  $J_0 = 0$ . This is true because  $J_k(i, j) \rightarrow J(i, j)$  for all  $i, j$  as  $k \rightarrow \infty$  (Prop. 1.6 in Section 3.1). To prove that  $\Delta_1^k(i, j)$  has the desired

property, it is useful to first verify that  $J_k(i, j)$  is monotonically nondecreasing in  $i$  (or  $j$ ) for fixed  $j$  (or  $i$ ). This is simple to show by induction or by arguing from first principles using the fact that  $J_k(i, j)$  has a  $k$ -stage optimal cost interpretation. Next we use Eqs. (2.11) and (2.14) to write

$$\begin{aligned} (\beta + \nu)\Delta_1^{k+1}(i, j) &= c_1 - c_2 \\ &\quad + \mu_1(J_k(i, j) - J_k((i-1)^+, j+1)) \\ &\quad + \mu_2(J_k(i+1, (j-1)^+) - J_k(i, j)) \\ &\quad + \lambda(\min[J_k(i+2, j), J_k(i+1, j+1)] \\ &\quad - \min[J_k(i+1, j+1), J_k(i, j+2)]). \end{aligned} \quad (2.15)$$

We now argue by induction. We have  $\Delta_1^0(i, j) = 0$  for all  $(i, j)$ . We assume that  $\Delta_1^k(i, j)$  is monotonically nondecreasing in  $i$  for fixed  $j$ , and show that the same is true for  $\Delta_1^{k+1}(i, j)$ . This can be verified by showing that each of the terms in the right-hand side of Eq. (2.15) is monotonically nondecreasing in  $i$  for fixed  $j$ . Indeed, the first term is constant, and the second and third terms are seen to be monotonically nondecreasing in  $i$  using the induction hypothesis for the case where  $i, j > 0$  and the earlier shown fact that  $J_k(i, j)$  is monotonically nondecreasing in  $i$  for the case where  $i = 0$  or  $j = 0$ . The last term on the right-hand side of Eq. (2.15) can be written as

$$\begin{aligned} &\lambda(J_k(i+1, j+1) + \min[J_k(i+2, j) - J_k(i+1, j+1), 0] \\ &\quad - J_k(i+1, j+1) - \min[0, J_k(i, j+2) - J_k(i+1, j+1)]) \\ &= \lambda(\min[0, J_k(i+1, j) - J_k(i+1, j+1)] \\ &\quad + \max[0, J_k(i+1, j+1) - J_k(i, j+2)]) \\ &= \lambda(\min[0, \Delta_1^k(i+1, j)] + \max[0, \Delta_1^k(i, j+1)]). \end{aligned}$$

Since  $\Delta_1^k(i+1, j)$  and  $\Delta_1^k(i, j+1)$  are monotonically nondecreasing in  $i$  by the induction hypothesis, the same is true for the preceding expression. Therefore, each of the terms on the right-hand side of Eq. (2.15) is monotonically nondecreasing in  $i$ , and the induction proof is complete. Thus the existence of an optimal threshold policy is established.

There are a number of generalizations of the routing problem of this example that admit a similar analysis and for which there exist optimal policies of the threshold type. For example, suppose that there are additional Poisson arrival processes with rates  $\lambda_1$  and  $\lambda_2$  at queues 1 and 2, respectively. The existence of an optimal threshold policy can be shown by a nearly verbatim repetition of our analysis. A more substantive extension is obtained when there is additional service capacity  $\mu$  that can be switched at the times of transition due to an arrival or service completion to serve a customer in queue 1 or 2. Then we can similarly prove that it is optimal to route to queue 1 if and only if  $(i, j) \in S_1$  and to switch the additional service capacity to queue 2 if and only if  $(i+1, j+1) \in S_1$ , where  $S_1$  is given by Eq. (2.12) and is characterized by a threshold function as in Eq. (2.13). For a proof of this and further extensions, we refer to [Haj84], which generalizes and unifies several earlier results on the subject.

### 5.3 SEMI-MARKOV PROBLEMS

We now consider a more general version of the continuous-time problem of Section 5.1. We still have a finite or a countable number of states, but we replace transition probabilities with *transition distributions*  $Q_{ij}(\tau, u)$  that, for a given pair  $(i, u)$ , specify the joint distribution of the transition interval and the next state:

$$Q_{ij}(\tau, u) = P\{t_{k+1} - t_k \leq \tau, x_{k+1} = j \mid x_k = i, u_k = u\}.$$

We assume that for all states  $i$  and  $j$ , and controls  $u \in U(i)$ ,  $Q_{ij}(\tau, u)$  is known and that the average transition time is finite:

$$\int_0^\infty \tau Q_{ij}(\tau, u) d\tau < \infty.$$

Note that the transition distributions specify the ordinary transition probabilities via

$$p_{ij}(u) = P\{x_{k+1} = j \mid x_k = i, u_k = u\} = \lim_{\tau \rightarrow \infty} Q_{ij}(\tau, u).$$

The difference from the model of Section 5.1 is that  $Q_{ij}(\tau, u)$  need not be an exponential distribution.

Continuous-time problems with general transition distributions as described above are called *semi-Markov problems* because, for any given policy, while at a transition time  $t_k$  the future of the system statistically depends only on the current state, at other times it may depend in addition on the time elapsed since the preceding transition. By contrast, when the transition distributions are exponential, the future of the system depends only on its current state at all times. This is a consequence of the so-called *memoryless property* of the exponential distribution. In our context, this property implies that, for any time  $t$  between the transition times  $t_k$  and  $t_{k+1}$ , the additional time  $t_{k+1} - t$  needed to effect the next transition is independent of the time  $t - t_k$  that the system has been in the current state [to see this, use the following generic calculation]

$$\begin{aligned} P\{\tau > r_1 + r_2 \mid \tau > r_1\} &= \frac{P\{\tau > r_1 + r_2\}}{P\{\tau > r_1\}} \\ &= \frac{e^{-\nu(r_1+r_2)}}{e^{-\nu r_1}} \\ &= e^{-\nu r_2} \\ &= P\{\tau > r_2\}, \end{aligned}$$

where  $r_1 = t - t_k$ ,  $r_2 = t_{k+1} - t$ , and  $\nu$  is the transition rate]. Thus, when the transition distributions are exponential, the state evolves in continuous time as a Markov process, but this need not be true for a more general distribution.

### Discounted Problems

Let us first consider a cost function of the form

$$\lim_{N \rightarrow \infty} E \left\{ \int_0^{t_N} e^{-\beta t} g(x(t), u(t)) dt \right\}, \quad (3.1)$$

where  $t_N$  is the completion time of the  $N$ th transition, and the function  $g$  and the positive discount parameter  $\beta$  are given. The cost function of an admissible  $N$ -stage policy  $\pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$  is given by

$$J_\pi^N(i) = \sum_{k=0}^{N-1} E \left\{ \int_{t_k}^{t_{k+1}} e^{-\beta t} g(x_k, \mu_k(x_k)) dt \mid x_0 = i \right\}.$$

We see that for all states  $i$  we have

$$J_\pi^N(i) = G(i, \mu_0(i)) + \sum_j \int_0^\infty e^{-\beta \tau} Q_{ij}(d\tau, \mu(i)) J_{\pi_1}^{N-1}(j), \quad (3.2)$$

where  $J_{\pi_1}^{N-1}(j)$  is the  $(N-1)$ -stage cost of the policy  $\pi_1 = \{\mu_1, \mu_2, \dots, \mu_{N-1}\}$  that is used after the first stage, and  $G(i, u)$  is the expected single stage cost corresponding to  $(i, u)$ . This latter cost is given by

$$G(i, u) = g(i, u) \sum_j \int_u^\infty \left( \int_0^\tau e^{-\beta t} dt \right) Q_{ij}(d\tau, u),$$

or equivalently, since  $\int_0^\tau e^{-\beta t} dt = (1 - e^{-\beta \tau})/\beta$ ,

$$G(i, u) = g(i, u) \sum_j \int_0^\infty \frac{1 - e^{-\beta \tau}}{\beta} Q_{ij}(d\tau, u). \quad (3.3)$$

If we denote

$$m_{ij}(u) = \int_0^\infty e^{-\beta \tau} Q_{ij}(d\tau, u), \quad (3.4)$$

we see that Eq. (3.2) can be written in the form

$$J_\pi^N(i) = G(i, \mu_0(i)) + \sum_j m_{ij}(\mu_0(i)) J_{\pi_1}^{N-1}(j), \quad (3.5)$$

which is similar to the corresponding equation for discounted discrete-time problems [we have  $m_{ij}(u)$  in place of  $\alpha p_{ij}(u)$ ].

The expression (3.5) motivates the use of mappings  $T$  and  $T_\mu$  that are similar to those used in Chapter 1 for discounted problems. Let us define for a function  $J$  and a stationary policy  $\mu$ ,

$$(T_\mu J)(i) = G(i, \mu(i)) + \sum_j m_{ij}(\mu(i)) J(j), \quad (3.6)$$

$$(TJ)(i) = \min_{u \in U(i)} \left[ G(i, u) + \sum_j m_{ij}(u) J(j) \right]. \quad (3.7)$$

Then by using Eq. (3.5), it can be seen that the cost function  $J_\pi$  of an infinite horizon policy  $\pi = \{\mu_0, \mu_1, \dots\}$  can be expressed as

$$J_\pi(i) = \lim_{N \rightarrow \infty} J_\pi^N(i) = \lim_{N \rightarrow \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} J_0)(i),$$

where  $J_0$  is the zero function [ $J_0(i) = 0$  for all  $i$ ]. The cost of a stationary policy  $\mu$  can be expressed as

$$J_\mu(i) = \lim_{N \rightarrow \infty} (T_\mu^N J_0)(i).$$

The discounted cost analysis of Section 1.2 carries through in its entirety, provided we assume that:

- (a)  $g(i, u)$  [and hence also  $G(i, u)$ ] is a bounded function of  $i$  and  $u$ .
- (b) The maximum over  $(i, u)$  of the sum  $\sum_j m_{ij}(u)$  is less than one; that is,

$$\rho = \max_{i, u \in U(i)} \sum_j m_{ij}(u) < 1. \quad (3.8)$$

Under these circumstances, the mappings  $T$  and  $T_\mu$  can be shown to be contraction mappings with modulus of contraction  $\rho$  [compare also with Prop. 2.4 in Section 1.2]. Using this fact, analogs of Props. 2.1-2.3 of Section 1.2 can be readily shown. In particular, the optimal cost function  $J^*$  is the unique bounded solution of Bellman's equation  $J = TJ$  or

$$J(i) = \min_{u \in U(i)} \left[ G(i, u) + \sum_j m_{ij}(u) J(j) \right].$$

In addition, there are analogs of several of the computational methods of Section 1.3, including policy iteration and linear programming.

What is happening here is that essentially we have the equivalent of a discrete-time discounted problem where the discount factor depends on  $i$  and  $u$ . In fact, in Exercise 1.12 of Chapter 1, a data transformation is given, which converts such a problem to an ordinary discrete-time discounted problem where the discount factor is the same for all  $i$  and  $u$ . With a little thought it can be seen that this data transformation is very similar to the uniformization process we discussed in Section 5.1.

We note that for the contraction property  $\rho < 1$  [cf. Eq. (3.8)] to hold, it is sufficient that there exist  $\bar{\tau} > 0$  and  $\epsilon > 0$  such that the transition time is greater than  $\bar{\tau}$  with probability greater than  $\epsilon > 0$ ; that is, we have for all  $i$  and  $u \in U(i)$ ,

$$1 - \sum_j Q_{ij}(\bar{\tau}, u) = \sum_j P\{\tau \geq \bar{\tau} \mid i, u, j\} \geq \epsilon. \quad (3.9)$$

In the case where the state space is countably infinite and the function  $g(i, u)$  is not bounded, the mappings  $T$  and  $T_\mu$  are not contraction mappings, and a discounted cost analysis that parallels the one of Section 1.2 is not possible. Even in this case, however, analogs of the results of Section 3.1 can often be shown under appropriate conditions that parallel Assumptions P and N of that section.

We finally note that in some problems, in addition to the cost (3.1), there is an extra expected stage cost  $\hat{g}(i, u)$  that is incurred at the time the control  $u$  is chosen at state  $i$ , and is independent of the length of the transition interval. In that case the mappings  $T$  and  $T_\mu$  should be changed to

$$(T_\mu J)(i) = \hat{g}(i, \mu(i)) + G(i, \mu(i)) + \sum_j m_{ij}(\mu(i))J(j),$$

$$(TJ)(i) = \min_{u \in U(i)} \left[ \hat{g}(i, u) + G(i, u) + \sum_j m_{ij}(u)J(j) \right]. \quad (3.10)$$

Another problem variation arises when the cost per unit time  $g$  depends on the next state  $j$ . In this problem formulation, once the system goes into state  $i$ , a control  $u \in U(i)$  is selected, the next state is determined to be  $j$  with probability  $p_{ij}(u)$ , and the cost of the next transition is  $g(i, u, j)\tau_{ij}(u)$  where  $\tau_{ij}(u)$  is random with distribution  $Q_{ij}(\tau, u)/p_{ij}(u)$ . In this case,  $G(i, u)$  should be defined by

$$G(i, u) = \sum_j \int_0^\infty g(i, u, j) \frac{1 - e^{-\beta\tau}}{\beta} Q_{ij}(d\tau, u),$$

[cf. Eq. (3.3)] and the preceding development goes through without modification.

### Example 3.1

Consider the manufacturer's problem of Example 1.1, with the only difference that the times between the arrivals of successive orders are uniformly distributed in a given interval  $[0, \tau_{\max}]$  instead of being exponentially distributed. Let  $F$  and  $NF$  denote the choices of filling and not filling the orders, respectively. The transition distributions are

$$Q_{ij}(\tau, F) = \begin{cases} \min\left[1, \frac{\tau}{\tau_{\max}}\right] & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$Q_{ij}(\tau, NF) = \begin{cases} \min\left[1, \frac{\tau}{\tau_{\max}}\right] & \text{if } j = i+1, \\ 0 & \text{otherwise.} \end{cases}$$

The effective cost per stage  $G$  of Eq. (3.3) is given by

$$G(i, F) = 0, \quad G(i, NF) = \gamma ci,$$

### Sec. 5.3 Semi-Markov Problems

where

$$\gamma = \int_0^{\tau_{\max}} \frac{1 - e^{-\beta\tau}}{\beta\tau_{\max}} d\tau.$$

The scalars  $m_{ij}$  of Eq. (3.4) that are nonzero are

$$m_{ii}(F) = m_{i(i+1)}(NF) = \alpha,$$

where

$$\alpha = \int_0^{\tau_{\max}} \frac{e^{-\beta\tau}}{\tau_{\max}} d\tau = \frac{1 - e^{-\beta\tau_{\max}}}{\beta\tau_{\max}},$$

Bellman's equation has the form

$$J(i) = \min [K + \alpha J(1), \gamma ci + \alpha J(i+1)], \quad i = 1, 2, \dots$$

As in Example 1.1, we can conclude that there exists a threshold  $i^*$  such that it is optimal to fill the orders if and only if their number  $i$  exceeds  $i^*$ .

### Example 3.2 (Control of an M/D/1 Queue)

Consider a single server queue where customers arrive according to a Poisson process with rate  $\lambda$ . The service time of a customer is deterministic and is equal to  $1/\mu$  where  $\mu$  is the service rate provided. The arrival and service rates  $\lambda$  and  $\mu$  can be selected from given subsets  $\Lambda$  and  $M$ , and can be changed only when a customer departs from the system. There are costs  $q(\lambda)$  and  $r(\mu)$  per unit time for using rates  $\lambda$  and  $\mu$ , respectively, and there is a waiting cost  $c(i)$  per unit time when there are  $i$  customers in the system (waiting in queue or undergoing service). We wish to find a rate-setting policy that minimizes the total cost when there is a positive discount parameter  $\beta$ .

This problem bears similarity with Examples 2.1 and 2.2 of Section 5.2. Note, however, that while in those examples the rates can be changed both when a customer arrives and when a customer departs, here the rates can be changed only when a customer departs. Because the service time distribution is not exponential, it is necessary to make this restriction in order to be able to use as state the number of customers in the system; if we allowed the arrival rate to also change when a customer arrives, the time already spent in service by the customer found in service by the arriving customer would have to be part of the state.

The transition distributions are given by

$$Q_{0j}(\tau, \lambda, \mu) = \begin{cases} 1 - e^{-\lambda\tau} & \text{if } j = 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$Q_{ij}(\tau, \lambda, \mu) = \begin{cases} p_{ij}(\lambda, \mu) & \text{if } 1/\mu \leq \tau, \\ 0 & \text{otherwise,} \end{cases} \quad i \geq 1,$$

where  $p_{ij}(\lambda, \mu)$  are the state transition probabilities. It can be seen that for  $i \geq 1$  and  $j \geq i-1$ ,  $p_{ij}(\lambda, \mu)$  can be calculated as the probability that  $j-i+1$  arrivals will occur in an interval of length  $[0, 1/\mu]$ . In particular, we have

$$p_{ij}(\lambda, \mu) = \begin{cases} \frac{e^{-\lambda/\mu} (\lambda/\mu)^{j-i+1}}{(j-i+1)!} & \text{if } j \geq i-1, \\ 0 & \text{otherwise,} \end{cases} \quad i \geq 1.$$

Using the above formulas and Eqs. (3.3)-(3.4) and (3.6)-(3.7), one can write Bellman's equation and solve the problem as if it were essentially a discrete-time discounted problem.

### Average Cost Problems

A natural cost function for the continuous-time average cost problem would be

$$\lim_{T \rightarrow \infty} \frac{1}{T} E \left\{ \int_0^T g(x(t), u(t)) dt \right\}. \quad (3.11)$$

However, we will use instead the cost function

$$\lim_{N \rightarrow \infty} \frac{1}{E\{t_N\}} E \left\{ \int_0^{t_N} g(x(t), u(t)) dt \right\}, \quad (3.12)$$

where  $t_N$  is the completion time of the  $N$ th transition. This cost function is also reasonable and turns out to be analytically convenient. We note, however, that the cost functions (3.11) and (3.12) are equivalent under the conditions of the subsequent analysis, although a rigorous justification of this is beyond our scope (see [Ros70], p. 52 and p. 160 for related analysis).

We assume that there are  $n$  states, denoted  $1, \dots, n$ , and that the control constraint set  $U(i)$  is finite for each state  $i$ . For each pair  $(i, u)$ , we denote by  $G(i, u)$  the single stage expected cost corresponding to state  $i$  and control  $u$ . We have

$$G(i, u) = g(i, u) \bar{\tau}_i(u), \quad (3.13)$$

where  $\bar{\tau}_i(u)$  is the expected value of the transition time corresponding to  $(i, u)$ :

$$\bar{\tau}_i(u) = \sum_{j=1}^n \int_0^\infty \tau Q_{ij}(d\tau, u). \quad (3.14)$$

If the cost per unit time  $g$  depends on the next state  $j$ , the expected transition cost  $G(i, u)$  should be defined by

$$G(i, u) = \sum_{j=1}^n \int_0^\infty g(i, u, j) \tau Q_{ij}(d\tau, u).$$

and the following analysis and results go through without modification.] We assume throughout the remainder of this section that

$$0 < \bar{\tau}_i(u) < \infty, \quad i = 1, \dots, n, \quad u \in U(i). \quad (3.15)$$

The cost function of an admissible policy  $\pi = \{\mu_0, \mu_1, \dots\}$  is given by

$$J_\pi(i) = \lim_{N \rightarrow \infty} \frac{1}{E\{t_N \mid x_0 = i, \pi\}} E \left\{ \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} g(x_k, \mu_k(x_k)) dt \mid x_0 = i \right\}.$$

Our earlier analysis of the discrete-time average cost problem in Chapter 4 suggests that under assumptions similar to those of Section 4.2, the cost  $J_\mu(i)$  of a stationary policy  $\mu$ , as well as the optimal average cost per stage  $J^*(i)$ , are independent of the initial state  $i$ . Indeed, we will see that the character of the solution of the problem is determined by the structure of the *embedded Markov chain*, which is the controlled discrete-time Markov chain whose transition probabilities are

$$p_{ij}(u) = \lim_{\tau \rightarrow \infty} Q_{ij}(\tau, u).$$

In particular, we will show that  $J_\mu(i)$  and  $J^*(i)$  are independent of  $i$  if and only if the same is true for the embedded Markov chain problem. For example, we will show that  $J_\mu(i)$  and  $J^*(i)$ , are independent of  $i$  if all stationary policies  $\mu$  are unichain; that is, the Markov chain with transition probabilities  $p_{ij}(\mu(i))$  has a single recurrent class.

We will also show that Bellman's equation for average cost semi-Markov problems resembles the corresponding discrete-time equation, and takes the form

$$h(i) = \min_{u \in U(i)} \left[ G(i, u) + \lambda \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h(j) \right]. \quad (3.16)$$

As a special case, when  $\bar{\tau}_i(u) = 1$  for all  $(i, u)$ , we obtain the corresponding discrete-time equation of Chapter 4. We illustrate Bellman's equation (3.16) for the case of a single unichain policy with the stochastic shortest path argument that we used to prove Prop. 2.5 in Section 4.2.

Consider a unichain policy  $\mu$  and without loss of generality assume that state  $n$  is a recurrent state in the Markov chain corresponding to  $\mu$ . For each state  $i \neq n$  let  $C_i$  and  $T_i$  be the expected cost and the expected time, respectively, up to reaching state  $n$  for the first time starting from  $i$ . Let also  $C_n$  and  $T_n$  be the expected cost and the expected time, respectively, up to returning to  $n$  for the first time starting from  $n$ . We can view  $C_i$  as the costs corresponding to  $\mu$  in a stochastic shortest path problem where  $n$  is a termination state and the costs are  $G(i, \mu(i))$ . Since  $\mu$  is a proper policy for this problem, from Prop. 1.1 in Section 2.1, we have that the scalars  $C_i$  solve uniquely the system of equations

$$C_i = G(i, \mu(i)) + \sum_{j=1, j \neq n}^n p_{ij}(\mu(i)) C_j, \quad i = 1, \dots, n. \quad (3.17)$$

Similarly, we can view  $T_i$  as the costs corresponding to  $\mu$  in a stochastic shortest path problem where  $n$  is a termination state and the costs are  $\bar{\tau}_i(\mu(i))$ , so that the  $T_i$  solve uniquely the system of equations

$$T_i = \bar{\tau}_i(\mu(i)) + \sum_{j=1, j \neq n}^n p_{ij}(\mu(i)) T_j, \quad i = 1, \dots, n. \quad (3.18)$$

Denote

$$\lambda_\mu = \frac{C_n}{T_n}. \quad (3.19)$$

Multiplying Eq. (3.18) by  $\lambda_\mu$  and subtracting it from Eq. (3.17), we obtain for all  $i = 1, \dots, n$ ,

$$C_i - \lambda_\mu T_i = G(i, \mu(i)) - \lambda_\mu \bar{\tau}_i(\mu(i)) + \sum_{j=1, j \neq n}^n p_{ij}(\mu(i))(C_j - \lambda_\mu T_j).$$

By defining

$$h_\mu(i) = C_i - \lambda_\mu T_i, \quad i = 1, \dots, n, \quad (3.20)$$

and by noting that from Eq. (3.19) we have

$$h_\mu(n) = 0,$$

we obtain for all  $i = 1, \dots, n$ ,

$$h_\mu(i) = G(i, \mu(i)) - \lambda_\mu \bar{\tau}_i(\mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) h_\mu(j), \quad (3.21)$$

which is Bellman's equation (3.16) for the case of a single stationary policy  $\mu$ .

We have not yet proved that the scalar  $\lambda_\mu$  of Eq. (3.19) is the average cost per stage corresponding to  $\mu$ . This fact will follow from the following proposition, which parallels Prop. 2.1 in Section 4.2 and shows that if Bellman's equation (3.16) has a solution  $(\lambda, h)$ , then the optimal average cost is equal to  $\lambda$  and is independent of the initial state.

**Proposition 3.1:** If a scalar  $\lambda$  and an  $n$ -dimensional vector  $h$  satisfy

$$h(i) = \min_{u \in U(i)} \left[ G(i, u) - \lambda \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h(j) \right], \quad i = 1, \dots, n, \quad (3.22)$$

then  $\lambda$  is the optimal average cost per stage  $J^*(i)$  for all  $i$ ,

$$\lambda = \min_{\pi} J_{\pi}(i) = J^*(i), \quad i = 1, \dots, n. \quad (3.23)$$

Furthermore, if  $\mu^*(i)$  attains the minimum in Eq. (3.22) for each  $i$ , the stationary policy  $\mu^*$  is optimal; that is,  $J_{\mu^*}(i) = \lambda$  for all  $i$ .

**Proof:** For any  $\mu$  consider the mapping  $T_\mu : \mathbb{R}^n \mapsto \mathbb{R}^n$  given by

$$(T_\mu h)(i) = G(i, \mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) h(j), \quad i = 1, \dots, n,$$

and the vector  $\bar{\tau}(\mu)$  and matrix  $P_\mu$  given by

$$\bar{\tau}(\mu) = \begin{pmatrix} \bar{\tau}_1(\mu(1)) \\ \vdots \\ \bar{\tau}_n(\mu(n)) \end{pmatrix}, \quad P_\mu = \begin{pmatrix} p_{11}(\mu(1)) & \dots & p_{1n}(\mu(1)) \\ \dots & \dots & \dots \\ p_{n1}(\mu(n)) & \dots & p_{nn}(\mu(n)) \end{pmatrix}.$$

Let  $\pi = \{\mu_0, \mu_1, \dots\}$  be any admissible policy and  $N$  be a positive integer. We have from Eq. (3.22),

$$T_{\mu_{N-1}} h \geq \lambda \bar{\tau}(\mu_{N-1}) + h.$$

By applying  $T_{\mu_{N-2}}$  to both sides of this relation, and by using the monotonicity of  $T_{\mu_{N-2}}$  and Eq. (3.22), we see that

$$\begin{aligned} T_{\mu_{N-2}} T_{\mu_{N-1}} h &\geq T_{\mu_{N-2}} (\lambda \bar{\tau}(\mu_{N-1}) + h) \\ &= \lambda P_{\mu_{N-2}} \bar{\tau}(\mu_{N-1}) + T_{\mu_{N-2}} h \\ &\geq \lambda P_{\mu_{N-2}} \bar{\tau}(\mu_{N-1}) + \lambda \bar{\tau}(\mu_{N-2}) + h. \end{aligned}$$

Continuing in the same manner, we finally obtain

$$T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} h \geq \bar{\lambda}_N(\pi) + h, \quad (3.24)$$

where  $\bar{\lambda}_N(\pi)$  is given by

$$\begin{aligned} \bar{\lambda}_N(\pi) &= P_{\mu_0} \cdots P_{\mu_{N-2}} \bar{\tau}(\mu_{N-1}) \\ &\quad + P_{\mu_0} \cdots P_{\mu_{N-3}} \bar{\tau}(\mu_{N-2}) + \cdots + \bar{\tau}(\mu_0). \end{aligned}$$

Note that the  $i$ th component of the vector  $\bar{\lambda}_N(\pi)$  is  $E\{t_N \mid x_0 = i, \pi\}$ , the expected value of the completion time of the  $N$ th transition when the initial state is  $i$  and  $\pi$  is used. Note also that equality holds in Eq. (3.24) if  $\mu_k(i)$  attains the minimum in Eq. (3.22) for all  $k$  and  $i$ . It can be seen that

$$(T_{\mu_0} T_{\mu_1} \cdots T_{\mu_{N-1}} h)(i) = E \left\{ h(x_N) + \int_0^{t_N} g(x(t), u(t)) dt \mid x_0 = i, \pi \right\}.$$

Using this relation in Eq. (3.24) and dividing by  $E\{t_N \mid x_0 = i, \pi\}$ , we obtain for all  $i$

$$\begin{aligned} \frac{E\{h(x_N) \mid x_0 = i, \pi\}}{E\{t_N \mid x_0 = i, \pi\}} &+ \frac{E\left\{ \int_0^{t_N} g(x(t), u(t)) dt \mid x_0 = i, \pi \right\}}{E\{t_N \mid x_0 = i, \pi\}} \\ &\geq \lambda + \frac{h(i)}{E\{t_N \mid x_0 = i, \pi\}}. \end{aligned}$$

Taking the limit as  $N \rightarrow \infty$  and using the fact  $\lim_{N \rightarrow \infty} E\{t_N \mid x_0 = i, \pi\} = \infty$  [cf. Eq. (3.15)], we see that

$$\lim_{N \rightarrow \infty} \frac{E \left\{ \int_0^{t_N} g(x(t), u(t)) dt \mid x_0 = i, \pi \right\}}{E\{t_N \mid x_0 = i, \pi\}} = J_\mu(i) \geq \lambda, \quad i = 1, \dots, n,$$

with equality if  $\mu_k(i)$  attains the minimum in Eq. (3.22) for all  $k$  and  $i$ . **Q.E.D.**

By combining Prop. 3.1 with Eq. (3.21), we obtain the following:

**Proposition 3.2:** Let  $\mu$  be a unichain policy. Then:

- (a) There exists a scalar  $\lambda_\mu$  and a vector  $h_\mu$  such that

$$J_\mu(i) = \lambda_\mu, \quad i = 1, \dots, n, \quad (3.25)$$

and

$$h_\mu(i) = G(i, \mu(i)) - \lambda_\mu \bar{\tau}_i(\mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) h_\mu(j), \quad i = 1, \dots, n. \quad (3.26)$$

- (b) Let  $t$  be a fixed state. The system of the  $n+1$  linear equations

$$h(i) = G(i, \mu(i)) - \lambda_\mu \bar{\tau}_i(\mu(i)) + \sum_{j=1}^n p_{ij}(\mu(i)) h(j), \quad i = 1, \dots, n, \quad (3.27)$$

$$h(t) = 0, \quad (3.28)$$

in the  $n+1$  unknowns  $\lambda, h(1), \dots, h(n)$  has a unique solution.

**Proof:** Part (a) follows from Prop. 3.1 and Eq. (3.21). The proof of part (b) is identical to the proof of Prop. 2.5(b) in Section 4.2. **Q.E.D.**

To establish conditions under which there exists a solution  $(\lambda, h)$  to Bellman's equation (3.22), we formulate a corresponding discrete-time average cost problem. Let  $\gamma$  be any scalar such that

$$0 < \gamma < \frac{\bar{\tau}_i(u)}{1 - p_{ii}(u)}$$

for all  $i$  and  $u \in U(i)$  with  $p_{ii}(u) < 1$ . Define also for all  $i$  and  $u \in U(i)$ ,

$$\tilde{p}_{ij}(u) = \begin{cases} \frac{\gamma p_{ij}(u)}{\bar{\tau}_i(u)} & \text{if } j \neq i, \\ 1 - \frac{\gamma(1-p_{ii}(u))}{\bar{\tau}_i(u)} & \text{if } j = i. \end{cases} \quad (3.29)$$

It can be seen that we have for all  $i$  and  $j$

$$0 \leq \tilde{p}_{ij}(u), \quad \sum_{j=1}^n \tilde{p}_{ij}(u) = 1, \\ \tilde{p}_{ij}(u) = 0 \quad \text{if and only if} \quad p_{ij}(u) = 0. \quad (3.30)$$

We view  $\tilde{p}_{ij}(u)$  as the transition probabilities of the discrete-time average cost problem whose expected stage cost corresponding to  $(i, u)$  is

$$\tilde{G}(i, u) := \frac{G(i, u)}{\bar{\tau}_i(u)}. \quad (3.31)$$

We call this the *auxiliary discrete time average cost problem*. The following proposition shows that this problem is essentially equivalent with our original semi-Markov average cost problem.

**Proposition 3.3** If the scalar  $\lambda$  and the vector  $\tilde{h}$  satisfy

$$\tilde{h}(i) = \min_{u \in U(i)} \left[ \tilde{G}(i, u) - \lambda + \sum_{j=1}^n \tilde{p}_{ij}(u) \tilde{h}(j) \right], \quad i = 1, \dots, n, \quad (3.32)$$

then  $\lambda$  and the vector  $h$  with components

$$h(i) = \gamma \tilde{h}(i), \quad i = 1, \dots, n, \quad (3.33)$$

satisfy Bellman's equation

$$h(i) = \min_{u \in U(i)} \left[ G(i, u) - \lambda \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h(j) \right], \quad i = 1, \dots, n, \quad (3.34)$$

for the semi-Markov average cost problem.

**Proof:** By substituting Eqs. (3.29), (3.31), and (3.33) in Eq. (3.32), we obtain after a straightforward calculation

$$0 = \min_{u \in U(i)} \frac{1}{\bar{\tau}_i(u)} \left[ G(i, u) - \lambda \bar{\tau}_i(u) + \sum_{j=1}^n p_{ij}(u) h(j) - h(i) \right], \quad i = 1, \dots, n.$$

This implies that the minimum of the expression within brackets in the right-hand side above is zero, which is equivalent to Bellman's equation (3.34). **Q.E.D.**

Note that in view of Eq. (3.30), the auxiliary discrete-time average cost problem and the semi-Markov average cost problem have the same probabilistic structure. In particular, if all stationary policies are unichain for one problem, the same is true for the other. Thus, the results and algorithms of Sections 4.2 and 4.3, when applied to the auxiliary discrete-time problem, yield results and algorithms for the semi-Markov problem. For example, value iteration, policy iteration, and linear programming can be applied to the auxiliary problem in order to solve the semi-Markov problem. We state a partial analog of Prop. 2.6 from Section 4.2.

**Proposition 3.4:** Consider the semi-Markov average cost problem, and assume either one of the following two conditions:

- (1) Every policy that is optimal within the class of stationary policies is unichain.
- (2) For every two states  $i$  and  $j$ , there exists a stationary policy  $\pi$  (depending on  $i$  and  $j$ ) such that, for some  $k$ ,

$$P(x_k = j \mid x_0 = i, \pi) > 0.$$

Then the optimal average cost per stage has the same value  $\lambda$  for all initial states  $i$ . Furthermore,  $\lambda$  together with a vector  $h$  satisfies Bellman's equation (3.34) for the semi-Markov average cost problem.

**Proof:** By Prop. 2.6 in Section 4.2, under either one of the conditions stated, Bellman's equation (3.32) for the auxiliary discrete-time average cost problem has a solution  $(\lambda, h)$ , from which a solution to Bellman's equation (3.34) can be extracted according to Prop. 3.3. **Q.E.D.**

### Example 3.3:

Consider the average cost version of the manufacturer's problem of Example 3.1. Here we have

$$\bar{\tau}_i(F) = \bar{\tau}_i(NF) = \frac{\bar{T}_{\max}}{2},$$

$$G(i, F) = K, \quad G(i, NF) = \frac{c_i T_{\max}}{2},$$

where  $F$  and  $NF$  denote the decisions to fill and not fill the orders, respectively. Bellman's equation takes the form

$$h(i) = \min \left[ K - \lambda \frac{\bar{T}_{\max}}{2} + h(1), c_i \frac{\bar{T}_{\max}}{2} - \lambda \frac{\bar{T}_{\max}}{2} + h(i+1) \right].$$

We leave it as an exercise for the reader to show that there exists a threshold  $i^*$  such that it is optimal to fill the orders if and only if  $i$  exceeds  $i^*$ .

### Example 3.4: [LiR71]

Consider a person providing a certain type of service to customers. Potential customers arrive according to a Poisson process with rate  $r$ ; that is the customer's interarrival times are independent and exponentially distributed with parameter  $r$ . Each customer offers one of  $n$  pairs  $(m_i, T_i)$ ,  $i = 1, \dots, n$ , where  $m_i$  is the amount of money offered for the service and  $T_i$  is the average amount of time that will be required to perform the service. Successive offers are independent and offer  $(m_i, T_i)$  occurs with probability  $p_i$ , where  $\sum_{i=1}^n p_i = 1$ . An offer may be rejected, in which case the customer leaves, or may be accepted in which case all offers that arrive while the customer is being served are lost. The problem is to determine the acceptance-rejection policy that maximizes the service provider's average income per unit time.

Let us denote by  $i$  the state corresponding to the offer  $(m_i, T_i)$ , and let  $A$  and  $R$  denote the accept and reject decision, respectively. We have

$$\bar{\tau}_i(A) = T_i + \frac{1}{r}, \quad \bar{\tau}_i(R) = \frac{1}{r},$$

$$G(i, A) = -m_i, \quad G(i, R) = 0,$$

$$p_{ij}(A) = p_{ij}(R) = p_j.$$

Bellman's equation is given by

$$h(i) = \min \left[ -m_i - \lambda \left( T_i + \frac{1}{r} \right) + \sum_{j=1}^n p_j h(j), -\lambda \frac{1}{r} + \sum_{j=1}^n p_j h(j) \right].$$

It follows that an optimal policy is to accept offer  $(i, T_i)$  if

$$\frac{m_i}{T_i} \geq -\lambda,$$

where  $-\lambda$  is the optimal average income per unit time.

## 5.4 NOTES, SOURCES, AND EXERCISES

The idea of using uniformization to convert continuous-time stochastic control problems involving Markov chains into discrete-time problems gained wide attention following [Lip75]; see also [BeR87].

Control of queueing systems has been researched extensively. For additional material on the problem of control of arrival rate or service rate (cf. Examples 2.1 and 2.2 in Section 5.2), see [BWN92], [CoR87], [CoV84], [RVW82], [Sob82], [STP74], and [Sti85]. For more on priority assignment and routing (cf. Examples 2.3, 2.4 in Section 5.2), see [BDM83], [BaD81], [BeT89b], [BhE91], [CoV84], [Har75a], [Har75b], [PaK81], [SuC91], and [AyR91], [CrC91], [EVW80], [EpV89], [Haj84], [LiK84], [TSC92], [ViE88], respectively.

Semi-Markov decision models were introduced in [Jew63] and are also discussed in [Ros70].

---

### EXERCISES

---

#### 5.1 (Proof of Validity of Uniformization)

Complete the details of the following argument, showing the validity of the uniformization procedure for the case of a finite number of states  $i = 1, \dots, n$ . We fix a policy, and for notational simplicity we do not show the dependence of transition rates on the control. Let  $p(t)$  be the row vector with coordinates

$$p_i(t) = P\{x(t) = i \mid x_0\}, \quad i = 1, \dots, n.$$

We have

$$dp(t)/dt = p(t)A,$$

where  $p(0)$  is the row vector with  $i$ th coordinate equal to one if  $x_0 = i$  and zero otherwise, and the matrix  $A$  has elements

$$a_{ij} = \begin{cases} \nu_i p_{ij} & \text{if } i \neq j, \\ -\nu_i & \text{if } i = j. \end{cases}$$

From this we obtain

$$p(t) = p(0)e^{At},$$

where

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!}.$$

Consider the transition probability matrix  $B$  of the uniform version

$$B = I + \frac{A}{\nu},$$

where  $\nu \geq \nu_i$ ,  $i = 1, \dots, n$ . Consider also the following equation:

$$e^{At} = e^{-\nu t} e^{B\nu t} = e^{-\nu t} \sum_{k=0}^{\infty} \frac{(B\nu t)^k}{k!}.$$

Use these relations to write

$$p(t) = p(0) \sum_{k=0}^{\infty} \Gamma(k, t) B^k,$$

where

$$\Gamma(k, t) = \frac{(\nu t)^k}{k!} e^{-\nu t} = \text{Prob}\{k \text{ transitions occur between 0 and } t \text{ in the uniform Markov chain}\}.$$

Verify that for  $i = 1, \dots, n$  we have

$$p_i(t) = \text{Prob}\{x(t) = i \text{ in the uniform Markov chain}\}.$$

#### 5.2

Consider the  $M/M/1$  queueing problem with variable service rate (Example 2.1 in Section 5.2). Assume that no arrivals are allowed ( $\lambda = 0$ ), and one can either serve a customer at rate  $\mu$  or refuse service ( $M = \{0, \mu\}$ ). Let the cost rates for customer waiting and service be  $c(i) = ci$  and  $q(\mu)$ , respectively, with  $q(0) = 0$ .

(a) Show that an optimal policy is to always serve an available customer if

$$\frac{q(\mu)}{\mu} \leq \frac{c}{\beta},$$

and to always refuse service otherwise.

(b) Analyze the problem when the cost rate for waiting is instead  $c(i) = ci^2$ .

#### 5.3

A person has an asset to sell for which she receives offers that can take one of  $n$  values. The times between successive offers are random, independent, and identically distributed with given distribution. Find the offer acceptance policy that maximizes  $E\{\alpha^T s\}$ , where  $T$  is the time of sale,  $s$  is the sale price, and  $\alpha \in (0, 1)$  is a discount factor.

## 5.4

Analyze the priority assignment problem of Example 2.3 in Section 5.2 within the semi-Markov context of Section 5.3, assuming that the customer service times are independent but not exponentially distributed. Consider both the discounted and the average cost cases.

## 5.5

An unemployed worker receives job offers according to a Poisson process with rate  $r$ , which she may accept or reject. The offered salary (per unit time) takes one of  $n$  possible values  $w_1, \dots, w_n$  with given probabilities, independently of preceding offers. If she accepts an offer at salary  $w_i$ , she keeps the job for a random amount of time that has expected value  $t_i$ . If she rejects the offer, she receives unemployment compensation  $c$  (per unit time) and is eligible to accept future offers. Solve the problem of maximizing the worker's average income per unit time.

## 5.6

Consider a single server queueing system where the server may be either on or off. Customers arrive according to a Poisson process with rate  $\lambda$ , and their service times are independent, identically distributed with given distribution. Each time a customer departs, the server may switch from on to off at a fixed cost  $C_0$  or from off to on at a fixed cost  $C_1$ . There is a cost  $c$  per unit time and customer residing in the system. Analyze this problem as a semi-Markov problem for the discounted and the average cost cases. In the latter case, assume that the queue has limited storage, and that customers arriving when the queue is full are lost.

## 5.7

Consider a semi-Markov version of the machine replacement problem of Example 2.4 in Section 1.2. Here, the transition times are random, independent, and have given distributions. Also  $g(i)$  is the cost per unit time of operating the machine at state  $i$ . Assume that  $p_{i(i+1)} > 0$  for all  $i < n$ . Derive Bellman's equation and analyze the problem.

## References

- [ABF93] Arapostathis, A., Borkar, V., Fernandez-Gaucherand, E., Ghosh, M., and Marcus, S., 1993. "Discrete-Time Controlled Markov Processes with Average Cost Criterion: A Survey," *SIAM J. on Control and Optimization*, Vol. 31, pp. 282-344.
- [AMT93] Archibald, T. W., McKinnon, K. I. M., and Thomas, L. C., 1993. "Serial and Parallel Value Iteration Algorithms for Discounted Markov Decision Processes," *Eur. J. Operations Research*, Vol. 67, pp. 188-203.
- [Ash70] Ash, R. B., 1970. *Basic Probability Theory*, Wiley, N. Y.
- [AyR91] Ayouni, S., and Rosberg, Z., 1991. "Optimal Routing to Two Parallel Heterogeneous Servers with Resequencing," *IEEE Trans. on Automatic Control*, Vol. 36, pp. 1436-1449.
- [BBS93] Barto, A. G., Bradtko, S. J., and Singh, S. P., 1993. "Real-Time Learning and Control Using Asynchronous Dynamic Programming," Comp. Science Dept. Tech. Report 91-57, Univ. of Massachusetts, Artificial Intelligence, Vol. 72, 1995, pp. 81-138.
- [BDM83] Baras, J. S., Dorsey, A. J., and Makowski, A. M., 1983. "Two Competing Queues with Linear Costs: The  $\mu$ -Rule is Often Optimal," Report SRR 83-1, Department of Electrical Engineering, University of Maryland.
- [BMP90] Benveniste, A., Metivier, M., and Prourier, P., 1990. *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, N. Y.
- [BPT94a] Bertsimas, D., Paschalidis, I. C., and Tsitsiklis, J. N., 1994. "Optimization of Multiclass Queueing Networks: Polyhedral and Nonlinear Characterizations of Achievable Performance," *Annals of Applied Probability*, Vol. 4, pp. 43-75.
- [BPT94b] Bertsimas, D., Paschalidis, I. C., and Tsitsiklis, J. N., 1994. "Branching Bandits and Klimov's Problem: Achievable Region and Side Constraints," Proc. of the 1994 IEEE Conference on Decision and Control, pp. 174-180, *IEEE Trans. on Automatic Control*, to appear.

- [BWN92] Blaue, J. P. C., de Waal, P. R., Nain, P., and Towsley, D., 1992. "Optimal Control of Admission to a Multiserver Queue with Two Arrival Streams," *IEEE Trans. on Automatic Control*, Vol. 37, pp. 785-797.
- [BaD81] Baras, J. S., and Dorsey, A. J., 1981. "Stochastic Control of Two Partially Observed Competing Queues," *IEEE Trans. Automatic Control*, Vol. AC-26, pp. 1106-1117.
- [Bai93] Baird, L. C., 1993. "Advantage Updating," Report WL-TR-93-1146, Wright Patterson AFB, OH.
- [Bai94] Baird, L. C., 1994. "Reinforcement Learning in Continuous Time: Advantage Updating," International Conf. on Neural Networks, Orlando, Fla.
- [Bai95] Baird, L. C., 1995. "Residual Algorithms: Reinforcement Learning with Function Approximation," Dept. of Computer Science Report, U.S. Air Force Academy, CO.
- [Bat73] Bather, J., 1973. "Optimal Decision Procedures for Finite Markov Chains," *Advances in Appl. Probability*, Vol. 5, pp. 328-339, pp. 521-540, 541-553.
- [BeC89] Bertsekas, D. P., and Castanon, D. A., 1989. "Adaptive Aggregation Methods for Infinite Horizon Dynamic Programming," *IEEE Trans. on Automatic Control*, Vol. AC-34, pp. 589-598.
- [BeN93] Bertsimas, D., and Nino-Mora, J., 1993. "Conservation Laws, Extended Polymatroids, and the Multiarmed Bandit Problem: A Unified Polyhedral Approach," *Mathematics of Operations Research*, to appear.
- [BeR87] Bentler, F. J., and Ross, K. W., 1987. "Uniformization for Semi-Markov Decision Processes Under Stationary Policies," *J. Appl. Prob.*, Vol. 24, pp. 399-420.
- [BeS78] Bertsekas, D. P., and Shreve, S. E., 1978. *Stochastic Optimal Control: The Discrete Time Case*, Academic Press, N. Y.
- [BeS79] Bertsekas, D. P., and Shreve, S. E., 1979. "Existence of Optimal Stationary Policies in Deterministic Optimal Control," *J. Math. Anal. and Appl.*, Vol. 69, pp. 607-620.
- [BeT89a] Bertsekas, D. P., and Tsitsiklis, J. N., 1989. *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J.
- [BeT89b] Bentler, F. J., and Teneketzis, D., 1989. "Routing in Queueing Networks Under Imperfect Information: Stochastic Dominance and Thresholds," *Stochastics and Stochastics Reports*, Vol. 26, pp. 81-100.
- [BeT91a] Bertsekas, D. P., and Tsitsiklis, J. N., 1991. "A Survey of Some Aspects of Parallel and Distributed Iterative Algorithms," *Automatica*,

- Vol. 27, pp. 3-21.
- [BeT91b] Bertsekas, D. P., and Tsitsiklis, J. N., 1991. "An Analysis of Stochastic Shortest Path Problems," *Math. Operations Research*, Vol. 16, pp. 580-595.
- [Bel57] Bellman, R., 1957. *Applied Dynamic Programming*, Princeton University Press, Princeton, N. J.
- [Ber71] Bertsekas, D. P., 1971. "Control of Uncertain Systems With a Set-Membership Description of the Uncertainty," Ph.D. Dissertation, Massachusetts Institute of Technology.
- [Ber72] Bertsekas, D. P., 1972. "Infinite Time Reachability of State Space Regions by Using Feedback Control," *IEEE Trans. Automatic Control*, Vol. AC-17, pp. 604-613.
- [Ber73a] Bertsekas, D. P., 1973. "Stochastic Optimization Problems with Nondifferentiable Cost Functionals," *J. Optimization Theory Appl.*, Vol. 12, pp. 218-231.
- [Ber73b] Bertsekas, D. P., 1973. "Linear Convex Stochastic Control Problems Over an Infinite Time Horizon," *IEEE Trans. Automatic Control*, Vol. AC-18, pp. 314-315.
- [Ber75] Bertsekas, D. P., 1975. "Convergence of Discretization Procedures in Dynamic Programming," *IEEE Trans. Automatic Control*, Vol. AC-20, pp. 415-419.
- [Ber76] Bertsekas, D. P., 1976. "On Error Bounds for Successive Approximation Methods," *IEEE Trans. Automatic Control*, Vol. AC-21, pp. 394-396.
- [Ber77] Bertsekas, D. P., 1977. "Monotone Mappings with Application in Dynamic Programming," *SIAM J. on Control and Optimization*, Vol. 15, pp. 438-464.
- [Ber82a] Bertsekas, D. P., 1982. "Distributed Dynamic Programming," *IEEE Trans. Automatic Control*, Vol. AC-27, pp. 610-616.
- [Ber82b] Bertsekas, D. P., 1982. *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, N. Y.
- [Ber93] Bertsekas, D. P., 1993. "A Generic Rank One Correction Algorithm for Markovian Decision Problems," Lab. for Info. and Decision Systems Report LIDS-P-2212, Massachusetts Institute of Technology, Operations Research Letters, to appear.
- [Ber95a] Bertsekas, D. P., 1995. *Nonlinear Programming*, Athena Scientific, Belmont, MA, to appear.
- [Ber95b] Bertsekas, D. P., 1995. "A Counterexample to Temporal Differences Learning," *Neural Computation*, Vol. 7, pp. 270-279.

- [Ber95c] Bertsekas, D. P., 1995. "A New Value Iteration Method for the Average Cost Dynamic Programming Problem," Lab. for Info. and Decision Systems Report, Massachusetts Institute of Technology.
- [BhE91] Bhattacharya, P. P., and Ephremides, A., 1991. "Optimal Allocations of a Server Between Two Queues with Due Times," IEEE Trans. on Automatic Control, Vol. 36, pp. 1417-1423.
- [Bill83] Billingsley, P., 1983. "The Singular Function of Bold Play," American Scientist, Vol. 71, pp. 392-397.
- [Blat62] Blackwell, D., 1962. "Discrete Dynamic Programming," Ann. Math. Statist., Vol. 33, pp. 719-726.
- [Blat65] Blackwell, D., 1965. "Discounted Dynamic Programming," Ann. Math. Statist., Vol. 36, pp. 226-235.
- [Bla70] Blackwell, D., 1970. "On Stationary Policies," J. Roy. Statist. Soc.: Ser. A, Vol. 133, pp. 33-38.
- [Bor89] Borkar, V. S., 1989. "Control of Markov Chains with Long-Run Average Cost Criterion: The Dynamic Programming Equations," SIAM J. on Control and Optimization, Vol. 27, pp. 642-657.
- [Bro65] Brown, B. W., 1965. "On the Iterative Method of Dynamic Programming on a Finite Space Discrete Markov Process," Ann. Math. Statist., Vol. 36, pp. 1279-1286.
- [CaS92] Cavazos-Cadena, R., and Sennott, L. I., 1992. "Comparing Recent Assumptions for the Existence of Optimal Stationary Policies," Operations Research Letters, Vol. 11, pp. 33-37.
- [Cav89a] Cavazos-Cadena, R., 1989. "Necessary Conditions for the Optimality Equations in Average-Reward Markov Decision Processes," Sys. Control Letters, Vol. 11, pp. 65-71.
- [Cav89b] Cavazos-Cadena, R., 1989. "Weak Conditions for the Existence of Optimal Stationary Policies in Average Markov Decisions Chains with Unbounded Costs," Kybernetika, Vol. 25, pp. 145-156.
- [Cav91] Cavazos-Cadena, R., 1991. "Recent Results on Conditions for the Existence of Average Optimal Stationary Policies," Annals of Operations Research, Vol. 28, pp. 3-28.
- [ChT89] Chow, C.-S., and Tsitsiklis, J. N., 1989. "The Complexity of Dynamic Programming," Journal of Complexity, Vol. 5, pp. 466-488.
- [ChT91] Chow, C.-S., and Tsitsiklis, J. N., 1991. "An Optimal One Way Multigrid Algorithm for Discrete Time Stochastic Control," IEEE Trans. on Automatic Control, Vol. AC-36, pp. 898-914.
- [CoR87] Courcoubetis, C. A., and Reiman, M. I., 1987. "Optimal Control of a Queueing System with Simultaneous Service Requirements," IEEE Trans. on Automatic Control, Vol. AC-32, pp. 717-727.

- [CoV84] Courcoubetis, C., and Varaiya, P. P., 1984. "The Service Process with Least Thinking Time Maximizes Resource Utilization," IEEE Trans. Automatic Control, Vol. AC-29, pp. 1005-1008.
- [CrC91] Cruz, R. L., and Chualh, M. C., 1991. "A Minimax Approach to a Simple Routing Problem," IEEE Trans. on Automatic Control, Vol. 36, pp. 1424-1435.
- [D'Ep60] D'Epenoux, F., 1960. "Sur un Probleme de Production et de Stockage Dans l'Aleatoire," Rev. Francaise Aut. Indus. Recherche Operationnelle, Vol. 14. (English Transl.: Management Sci., Vol. 10, 1963, pp. 98-108).
- [Dan63] Dantzig, G. B., 1963. Linear Programming and Extensions, Princeton Univ. Press, Princeton, N. J.
- [Den67] Denardo, E. V., 1967. "Contraction Mappings in the Theory Underlying Dynamic Programming," SIAM Review, Vol. 9, pp. 165-177.
- [Der70] Derman, C., 1970. Finite State Markovian Decision Processes, Academic Press, N. Y.
- [DuS65] Dubins, L., and Savage, L. M., 1965. How to Gamble If You Must, McGraw-Hill, N. Y.
- [DyY79] Dynkin, E. B., and Yuskevich, A. A., 1979. Controlled Markov Processes, Springer-Verlag, N. Y.
- [EVW80] Ephremides, A., Varaiya, P. P., and Walrand, J. C., 1980. "A Simple Dynamic Routing Problem," IEEE Trans. Automatic Control, Vol. AC-25, pp. 690-693.
- [EaZ62] Eaton, J. H., and Zadeh, L. A., 1962. "Optimal Pursuit Strategies in Discrete State Probabilistic Systems," Trans. ASME Ser. D. J. Basic Eng., Vol. 84, pp. 23-29.
- [EpV89] Ephremides, A., and Verdú, S., 1989. "Control and Optimization Methods in Communication Network Problems," IEEE Trans. Automatic Control, Vol. AC-34, pp. 930-942.
- [FAM90] Fernández-Gaucherand, E., Arapostathis, A., and Marcus, S. I., 1990. "Remarks on the Existence of Solutions to the Average Cost Optimality Equation in Markov Decision Processes," Systems and Control Letters, Vol. 15, pp. 425-432.
- [FAM91] Fernández-Gaucherand, E., Arapostathis, A., and Marcus, S. I., 1991. "On the Average Cost Optimality Equation and the Structure of Optimal Policies for Partially Observable Markov Decision Processes," Annals of Operations Research, Vol. 29, pp. 439-470.
- [FeS94] Feinberg, E. A., and Schwartz, A., 1994. "Markov Decision Models

- with Weighted Discounted Criteria," *Mathematics of Operations Research*, Vol. 19, pp. 1-17.
- [Fei78] Feinberg, E. A., 1978. "The Existence of a Stationary  $\epsilon$ -Optimal Policy for a Finite-State Markov Chain," *Theor. Prob. Appl.*, Vol. 23, pp. 207-313.
- [Fei92a] Feinberg, E. A., 1992. "Stationary Strategies in Borel Dynamic Programming," *Mathematics of Operations Research*, Vol. 125, pp. 87-96.
- [Fei92b] Feinberg, E. A., 1992. "A Markov Decision Model of a Search Process," *Contemporary Mathematics*, Vol. 125, pp. 87-96.
- [Fox71] Fox, B. L., 1971. "Finite State Approximations to Denumerable State Dynamic Programs," *J. Math. Anal. Appl.*, Vol. 34, pp. 665-670.
- [Gal95] Gallager, R. G., 1995. *Discrete Stochastic Processes*, Kluwer, N. Y.
- [Gho90] Ghosh, M. K., 1990. "Markov Decision Processes with Multiple Costs," *Operations Research Letters*, Vol. 9, pp. 257-260.
- [GJ74] Gittins, J. C., and Jones, D. M., 1974. "A Dynamic Allocation Index for the Sequential Design of Experiments," *Progress in Statistics* (J. Gani, ed.), North-Holland, Amsterdam, pp. 241-266.
- [Git79] Gittins, J. C., 1979. "Bandit Processes and Dynamic Allocation Indices," *J. Roy. Statist. Soc.*, Vol. B, No. 41, pp. 148-164.
- [HBK94] Harmon, M. E., Baird, L. C., and Klopf, A. H., 1994. "Advantage Updating Applied to a Differential Game," Presented at NIPS Conf., Denver, Colo.
- [HHL91] Hernandez-Lerma, O., Henet, J. C., and Lasserre, J. B., 1991. "Average Cost Markov Decision Processes: Optimality Conditions," *J. Math. Anal. Appl.*, Vol. 158, pp. 396-406.
- [Hal86] Haurie, A., and L'Ecuyer, P., 1986. "Approximation and Bounds in Discrete Event Dynamic Programming," *IEEE Trans. Automatic Control*, Vol. AC-31, pp. 227-235.
- [Haj84] Hajek, B., 1984. "Optimal Control of Two Interacting Service Stations," *IEEE Trans. Automatic Control*, Vol. AC-29, pp. 491-499.
- [Har75a] Harrison, J. M., 1975. "A Priority Queue with Discounted Linear Costs," *Operations Research*, Vol. 23, pp. 260-269.
- [Har75b] Harrison, J. M., 1975. "Dynamic Scheduling of a Multiclass Queue: Discount Optimality," *Operations Research*, Vol. 23, pp. 270-282.
- [Has68] Hastings, N. A. J., 1968. "Some Notes on Dynamic Programming and Replacement," *Operational Research Quart.*, Vol. 19, pp. 453-464.
- [HeS84] Heyman, D. P., and Sobel, M. J., 1984. *Stochastic Models in Operations Research*, Vol. II, McGraw-Hill, N. Y.

- [Her89] Hernandez-Lerma, O., 1989. *Adaptive Markov Control Processes*, Springer-Verlag, N. Y.
- [HoT74] Hordijk, A., and Tijms, H., 1974. "Convergence Results and Approximations for Optimal  $(s, S)$  Policies," *Management Sci.*, Vol. 20, pp. 1431-1438.
- [How60] Howard, R., 1960. *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA.
- [Igl63a] Iglehart, D. L., 1963. "Optimality of  $(S, s)$  Policies in the Infinite Horizon Dynamic Inventory Problem," *Management Sci.*, Vol. 9, pp. 259-267.
- [Igl63b] Iglehart, D. L., 1963. "Dynamic Programming and Stationary Analysis of Inventory Problems," in Scarf, H., Gillard, D., and Shelly, M., (eds.), *Multistage Inventory Models and Techniques*, Stanford University Press, Stanford, CA, 1963.
- [JJS94] Jaakkola, T., Jordan, M. I., and Singh, S. P., 1994. "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms," *Neural Computation*, Vol. 6, pp. 1185-1201.
- [Jew63] Jewell, W., 1963. "Markov Renewal Programming I and II," *Operations Research*, Vol. 2, pp. 938-971.
- [KaV87] Katehakis, M., and Veinott, A. F., 1987. "The Multi-Armed Bandit Problem: Decomposition and Computation," *Math. of Operations Research*, Vol. 12, pp. 262-268.
- [Kal83] Kallenberg, L. C. M., 1983. *Linear Programming and Finite Markov Control Problems*, Mathematical Centre Report, Amsterdam.
- [Kel81] Kelly, F. P., "Multi-Armed Bandits with Discount Factor Near One: The Bernoulli Case," *The Annals of Statistics*, Vol. 9, pp. 987-1001.
- [Kle68] Kleinman, D. L., 1968. "On an Iterative Technique for Riccati Equation Computations," *IEEE Trans. Automatic Control*, Vol. AC-13, pp. 114-115.
- [KuV86] Kumar, P. R., and Varaiya, P. P., 1986. *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, Englewood Cliffs, N. J.
- [Kum85] Kumar, P. R., 1985. "A Survey of Some Results in Stochastic Adaptive Control," *SIAM J. on Control and Optimization*, Vol. 23, pp. 329-380.
- [Kus71] Kushner, H. J., 1971. *Introduction to Stochastic Control*, Holt, Rinehart and Winston, N. Y.
- [Kus78] Kushner, H. J., 1978. "Optimality Conditions for the Average Cost per Unit Time Problem with a Diffusion Model," *SIAM J. Control Opti-*

- imization, Vol. 16, pp. 330-346.
- [Las88] Lasserre, J. B., 1988. Conditions for Existence of Average and Blackwell Optimal Stationary Policies in Denumerable Markov Decision Processes," *J. Math. Anal. Appl.*, Vol. 136, pp. 479-490.
- [LiK84] Liu, W., and Kumar, P. R., 1984. "Optimal Control of a Queueing System with Two Heterogeneous Servers," *IEEE Trans. Automatic Control*, Vol. AC-29, pp. 696-703.
- [LiR71] Lippman, S. A., and Ross, S. M., 1971. "The Streetwalker's Dilemma: A Job-Shop Model," *SIAM J. of Appl. Math.*, Vol. 20, pp. 336-342.
- [LiS61] Lusternik, L., and Sobolev, V., 1961. *Elements of Functional Analysis*, Ungar, N. Y.
- [Lip75] Lippman, S. A., 1975. "Applying a New Device in the Optimization of Exponential Queueing Systems," *Operations Research*, Vol. 23, pp. 687-710.
- [LjS83] Ljung, L., and Soderstrom, T., 1983. *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA.
- [Lue69] Luenberger, D. G., 1969. *Optimization by Vector Space Methods*, Wiley, N. Y.
- [McQ66] MacQueen, J., 1966. "A Modified Dynamic Programming Method for Markovian Decision Problems," *J. Math. Anal. Appl.*, Vol. 14, pp. 38-43.
- [MoW77] Morton, T. E., and Wecker, W., 1977. "Discounting, Ergodicity and Convergence for Markov Decision Processes," *Management Sci.*, Vol. 23, pp. 890-900.
- [Mor71] Morton, T. E., 1971. "On the Asymptotic Convergence Rate of Cost Differences for Markovian Decision Processes," *Operations Research*, Vol. 19, pp. 244-248.
- [NTW89] Nain, P., Tsoucas, P., and Walrand, J., 1989. "Interchanging Arguments in Stochastic Scheduling," *J. of Appl. Prob.*, Vol. 27, pp. 815-826.
- [NgP86] Nguyen, S., and Pallottino, S., 1986. "Hyperpaths and Shortest Hyperpaths," in *Combinatorial Optimization* by B. Simeone (ed.), Springer-Verlag, N. Y., pp. 258-271.
- [Odo69] Odoni, A. R., 1969. "On Finding the Maximal Gain for Markov Decision Processes," *Operations Research*, Vol. 17, pp. 857-860.
- [OrR70] Ortega, J. M., and Rheinboldt, W. C., 1970. *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, N. Y.
- [Orn69] Ornstein, D., 1969. "On the Existence of Stationary Optimal Strategies," *Proc. Amer. Math. Soc.*, Vol. 20, pp. 563-569.

- [PBT95] Polymenakos, L. C., Bertsekas, D. P., and Tsitsiklis, J. N., 1995. "Efficient Algorithms for Continuous-Space Shortest Path Problems," Lab. for Info. and Decision Systems Report LIDS-P-2292, Massachusetts Institute of Technology.
- [PBW79] Popvack, J. L., Brown, R. L., and White, C. C., III, 1969. "Discrete Versions of an Algorithm due to Varaiya," *IEEE Trans. Aut. Control*, Vol. 24, pp. 503-504.
- [PaK81] Pattipati, K. R., and Kleinman, D. L., 1981. "Priority Assignment Using Dynamic Programming for a Class of Queueing Systems," *IEEE Trans. on Automatic Control*, Vol. AC-26, pp. 1095-1106.
- [PaT87] Papadimitriou, C. H., and Tsitsiklis, J. N., 1987. "The Complexity of Markov Decision Processes," *Math. Operations Research*, Vol. 12, pp. 441-450.
- [Pla77] Platzman, L., 1977. "Improved Conditions for Convergence in Undiscounted Markov Renewal Programming," *Operations Research*, Vol. 25, pp. 529-533.
- [PoA69] Pollatschek, M., and Avi-Itzhak, B., 1969. "Algorithms for Stochastic Games with Geometrical Interpretation," *Man. Sci.*, Vol. 15, pp. 399-413.
- [PoT78] Portens, E., and Totten, J., 1978. "Accelerated Computation of the Expected Discounted Return in a Markov Chain," *Operations Research*, Vol. 26, pp. 350-358.
- [PoT92] Polychronopoulos, G. H., and Tsitsiklis, J. N., 1992. "Stochastic Shortest Path Problems with Recourse," Lab. for Info. and Decision Systems Report LIDS-P-2183, Massachusetts Institute of Technology.
- [Por71] Porteus, E., 1971. "Some Bounds for Discounted Sequential Decision Processes," *Man. Sci.*, Vol. 18, pp. 7-11.
- [Por75] Porteus, E., 1975. "Bounds and Transformations for Finite Markov Decision Chains," *Operations Research*, Vol. 23, pp. 761-784.
- [Por81] Porteus, E., 1981. "Improved Conditions for Convergence in Undiscounted Markov Renewal Programming," *Operations Research*, Vol. 25, pp. 529-533.
- [PsT93] Psaraftis, H. N., and Tsitsiklis, J. N., 1993. "Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs," *Operations Research*, Vol. 41, pp. 91-101.
- [PuB78] Puterman, M. L., and Brumelle, S. L., 1978. "The Analytic Theory of Policy Iteration," in *Dynamic Programming and Its Applications*, M. L. Puterman (ed.), Academic Press, N. Y.
- [PuS78] Puterman, M. L., and Shin, M. C., 1978. "Modified Policy Iteration

- Algorithms for Discounted Markov Decision Problems," *Management Sci.*, Vol. 24, pp. 1127-1137.
- [Put82] Puterman, M. L., and Shiu, M. C., 1982. "Action Elimination Procedures for Modified Policy Iteration Algorithms," *Operations Research*, Vol. 30, pp. 301-318.
- [Put78] Puterman, M. L. (ed.), 1978. *Dynamic Programming and its Applications*, Academic Press, N. Y.
- [Put94] Puterman, M. L., 1994. "Markovian Decision Problems," *J. Wiley*, N. Y.
- [RVW82] Rosberg, Z., Varaiya, P. P., and Walrand, J. C., 1982. "Optimal Control of Service in Tandem Queues," *IEEE Trans. Automatic Control*, Vol. AC-27, pp. 600-609.
- [RaF91] Raghavan, T. E. S., and Filar, J. A., 1991. "Algorithms for Stochastic Games - A Survey," *ZOR - Methods and Models of Operations Research*, Vol. 35, pp. 437-472.
- [RGS92] Ritt, R. K., and Sennott, L. I., 1992. "Optimal Stationary Policies in General State Markov Decision Chains with Finite Action Set," *Math. Operations Research*, Vol. 17, pp. 901-909.
- [Roc70] Rockafellar, R. T., 1970. *Convex Analysis*, Princeton University Press, Princeton, N. J.
- [Ros70] Ross, S. M., 1970. *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, CA.
- [Ros83a] Ross, S. M., 1983. *Introduction to Stochastic Dynamic Programming*, Academic Press, N. Y.
- [Ros83b] Ross, S. M., 1983. *Stochastic Processes*, Wiley, N. Y.
- [Ros89] Ross, K. W., 1989. "Randomized and Past-Dependent Policies for Markov Decision Processes with Multiple Constraints," *Operations Research*, Vol. 37, pp. 474-477.
- [Rus94] Rust, J., 1994. "Using Randomization to Break the Curse of Dimensionality," Unpublished Report, Dept. of Economics, University of Wisconsin.
- [Rus95] Rust, J., 1995. "Numerical Dynamic Programming in Economics," in *Handbook of Computational Economics*, H. Amman, D. Kendrick, and J. Rust (eds.).
- [SPK85] Schweitzer, P. J., Puterman, M. L., and Kindle, K. W., 1985. "Iterative Aggregation-Disaggregation Procedures for Solving Discounted Semi-Markovian Reward Processes," *Operations Research*, Vol. 33, pp. 589-605.

- [SeF77] Schweitzer, P. J., and Federgruen, A., 1977. "The Asymptotic Behavior of Value Iteration in Markov Decision Problems," *Math. Operations Research*, Vol. 2, pp. 360-381.
- [SeF78] Schweitzer, P. J., and Federgruen, A., 1978. "The Functional Equations of Undiscounted Markov Renewal Programming," *Math. Operations Research*, Vol. 3, pp. 308-321.
- [SeS85] Schweitzer, P. J., and Seidman, A., 1985. "Generalized Polynomial Approximations in Markovian Decision Problems," *J. Math. Anal. and Appl.*, Vol. 110, pp. 568-582.
- [Sch68] Schweitzer, P. J., 1968. "Perturbation Theory and Finite Markov Chains," *J. Appl. Prob.*, Vol. 5, pp. 401-413.
- [Sch71] Schweitzer, P. J., 1971. "Iterative Solution of the Functional Equations of Undiscounted Markov Renewal Programming," *J. Math. Anal. Appl.*, Vol. 34, pp. 495-501.
- [Sch72] Schweitzer, P. J., 1972. "Data Transformations for Markov Renewal Programming," talk at National ORSA Meeting, Atlantic City, N. J.
- [Sch75] Schäl, M., 1975. "Conditions for Optimality in Dynamic Programming and for the Limit of  $n$ -Stage Optimal Policies to be Optimal," *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, Vol. 32, pp. 179-196.
- [Sch81] Schweitzer, P. J., 1981. "Bottleneck Determination in a Network of Queues," Graduate School of Management Working Paper No. 8107, University of Rochester, Rochester, N. Y.
- [Sch93] Schwartz, A., 1993. "A Reinforcement Learning Method for Maximizing Undiscounted Rewards," Proc. of the Tenth Machine Learning Conference.
- [Sen86] Sennott, L. I., 1986. "A New Condition for the Existence of Optimum Stationary Policies in Average Cost Markov Decision Processes," *Operations Research Lett.*, Vol. 5, pp. 17-23.
- [Sen89a] Sennott, L. I., 1989. "Average Cost Optimal Stationary Policies in Infinite State Markov Decision Processes with Unbounded Costs," *Operations Research*, Vol. 37, pp. 626-633.
- [Sen89b] Sennott, L. I., 1989. "Average Cost Semi-Markov Decision Processes and the Control of Queueing Systems," *Prob. Eng. Info. Sci.*, Vol. 3, pp. 247-272.
- [Sen91] Sennott, L. I., 1991. "Value Iteration in Countable State Average Cost Markov Decision Processes with Unbounded Cost," *Annals of Operations Research*, Vol. 28, pp. 261-272.
- [Sen93a] Sennott, L. I., 1993. "The Average Cost Optimality Equation and Critical Number Policies," *Prob. Eng. Info. Sci.*, Vol. 7, pp. 47-67.

- [Sen93b] Sennott, L. I., 1993. "Constrained Average Cost Markov Decision Chains," *Prob. Eng. Info. Sci.*, Vol. 7, pp. 69-83.
- [Ser79] Serfozo, R., 1979. "An Equivalence Between Discrete and Continuous Time Markov Decision Processes," *Operations Research*, Vol. 27, pp. 616-620.
- [Sha53] Shapley, L. S., 1953. "Stochastic Games," *Proc. Nat. Acad. Sci. U.S.A.*, Vol. 39.
- [Sin94] Singh, S. P., 1994. "Reinforcement Learning Algorithms for Average-Payoff Markovian Decision Processes," *Proc. of 12th National Conference on Artificial Intelligence*, pp. 202-207.
- [Sob82] Sobel, M. J., 1982. "The Optimality of Full-Service Policies," *Operations Research*, Vol. 30, pp. 636-649.
- [Sti74] Stidham, S., and Prabhu, N. U., 1974. "Optimal Control of Queueing Systems," in *Mathematical Methods in Queueing Theory* (Lecture Notes in Economics and Math. Syst., Vol. 98), A. B. Clarke (Ed.), Springer-Verlag, N. Y., pp. 263-294.
- [Sti85] Stidham, S. S., 1985. "Optimal Control of Admission to a Queueing System," *IEEE Trans. Automatic Control*, Vol. AC-30, pp. 705-713.
- [Str66] Strauch, R., 1966. "Negative Dynamic Programming," *Ann. Math. Statist.*, Vol. 37, pp. 871-890.
- [SuC91] Suk, J.-B., and Cassandras, C. G., 1991. "Optimal Scheduling of Two Competing Queues with Blocking," *IEEE Trans. on Automatic Control*, Vol. 36, pp. 1086-1091.
- [Sut88] Sutton, R. S., 1988. "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, Vol. 3, pp. 9-44.
- [TSC92] Towsley, D., Sparaggis, P. D., and Cassandras, C. G., 1992. "Optimal Routing and Buffer Allocation for a Class of Finite Capacity Queueing Systems," *IEEE Trans. on Automatic Control*, Vol. 37, pp. 1446-1451.
- [Tes92] Tesrauro, G., 1992. "Practical Issues in Temporal Difference Learning," *Machine Learning*, Vol. 8, pp. 257-277.
- [TsV94] Tsitsiklis, J. N., and Van Roy, B., 1994. "Feature-Based Methods for Large-Scale Dynamic Programming," Lab. for Info. and Decision Systems Report LIDS-P-2277, Massachusetts Institute of Technology, Machine Learning, to appear.
- [Tse90] Tseng, P., 1990. "Solving  $H$ -Horizon, Stationary Markov Decision Problems in Time Proportional to  $\log(H)$ ," *Operations Research Letters*, Vol. 9, pp. 287-297.
- [Tsi86] Tsitsiklis, J. N., 1986. "A Lemma on the Multiarmed Bandit Problem," *IEEE Trans. Automatic Control*, Vol. AC-31, pp. 576-577.

- [Tsi89] Tsitsiklis, J. N., 1989. "A Comparison of Jacobi and Gauss-Seidel Parallel Iterations," *Applied Math. Lett.*, Vol. 2, pp. 167-170.
- [Tsi93a] Tsitsiklis, J. N., 1993. "Efficient Algorithms for Globally Optimal Trajectories," Lab. for Info. and Decision Systems Report LIDS-P-2210, Massachusetts Institute of Technology, *IEEE Trans. on Automatic Control*, to appear.
- [Tsi93b] Tsitsiklis, J. N., 1993. "A Short Proof of the Gittins Index Theorem," Lab. for Info. and Decision Systems Report LIDS-P-2171, Massachusetts Institute of Technology; also *Annals of Applied Probability*, Vol. 4, 1994, pp. 194-199.
- [Tsi94] Tsitsiklis, J. N., 1994. "Asynchronous Stochastic Approximation and Q-Learning," *Machine Learning*, Vol. 16, pp. 185-202.
- [Tso91] Tsoukas, P., 1991. "The Region of Achievable Performance in a Model of Klimov," *Research Report*, I.B.M.
- [VWB85] Varaiya, P. P., Walrand, J. C., and Buyukkoc, C., 1985. "Extensions of the Multiarmed Bandit Problem: The Discounted Case," *IEEE Trans. Automatic Control*, Vol. AC-30, pp. 426-439.
- [Var78] Varaiya, P. P., 1978. "Optimal and Suboptimal Stationary Controls of Markov Chains," *IEEE Trans. Automatic Control*, Vol. AC-23, pp. 388-394.
- [VeP84] Verd'eu, S., and Poor, H. V., 1984. "Backward, Forward, and Backward-Forward Dynamic Programming Models under Commutativity Conditions," *Proc. 1984 IEEE Decision and Control Conference*, Las Vegas, NE, pp. 1081-1086.
- [VeP87] Verd'eu, S., and Poor, H. V., 1987. "Abstract Dynamic Programming Models under Commutativity Conditions," *SIAM J. on Control and Optimization*, Vol. 25, pp. 990-1006.
- [Wei66] Veinott, A. F., Jr., 1966. "On Finding Optimal Policies in Discrete Dynamic Programming with no Discounting," *Ann. Math. Statist.*, Vol. 37, pp. 1284-1294.
- [Wei69] Veinott, A. F., Jr., 1969. "Discrete Dynamic Programming with Sensitive Discount Optimality Criteria," *Ann. Math. Statist.*, Vol. 40, pp. 1635-1660.
- [ViE88] Viniotis, I., and Ephremides, A., 1988. "Extension of the Optimality of the Threshold Policy in Heterogeneous Multiserver Queueing Systems," *IEEE Trans. on Automatic Control*, Vol. 33, pp. 101-109.
- [Wat89] Watkins, C. J. C. H., "Learning from Delayed Rewards," Ph.D. Thesis, Cambridge Univ., England.

- [Web92] Weber, R., 1991. "On the Gittins Index for Multiarmed Bandits," preprint; Annals of Applied Probability, Vol. 3, 1993.
- [Whi80] White, C. C., and Kim, K., 1980. "Solution Procedures for Partially Observed Markov Decision Processes," J. Large Scale Systems, Vol. 1, pp. 129-140.
- [Whi63] White, D. J., 1963. "Dynamic Programming, Markov Chains, and the Method of Successive Approximations," J. Math. Anal. and Appl., Vol. 6, pp. 373-376.
- [Whi78] Whitt, W., 1978. "Approximations of Dynamic Programs I," Math. Operations Research, Vol. 3, pp. 231-243.
- [Whi79] Whitt, W., 1979. "Approximations of Dynamic Programs II," Math. Operations Research, Vol. 4, pp. 179-185.
- [Whi80a] White, D. J., 1980. "Finite State Approximations for Denumerable State Infinite Horizon Discounted Markov Decision Processes: The Method of Successive Approximations," in Recent Developments in Markov Decision Processes, Hartley, R., Thomas, L. C., and White, D. J. (eds.), Academic Press, N. Y., pp. 57-72.
- [Whi80b] Whittle, P., 1980. "Multi-Armed Bandits and the Gittins Index," J. Roy. Statist. Soc. Ser. B, Vol. 42, pp. 143-149.
- [Whi81] Whittle, P., 1981. "Arm-Acquiring Bandits," The Annals of Probability, Vol. 9, pp. 284-292.
- [Whi82] Whittle, P., 1982. Optimization Over Time, Wiley, N. Y., Vol. 1, 1982, Vol. 2, 1983.

# INDEX

## A

- Admissible policy, 3
- Advantage updating, 122, 132
- Aggregation, 44, 104, 219
- Approximation in policy space, 117
- Asset selling, 157, 275
- Asynchronous algorithms, 30, 74, 120
- Average cost problem, 184, 249, 266

## B

- Basis functions, 51, 65, 103
- Bellman's equation, 8, 11, 83, 108, 137, 186, 191, 196, 225, 247, 268
- Blackwell optimal policy, 193, 233
- Bold strategy, 162

## C

- Chess, 102, 117
- Column reduction, 67
- Contraction mappings, 52, 65, 86, 128
- Consistently improving policies, 90, 122, 127
- Controllability, 151, 228
- Cost approximation, 51, 101, 225

## D

- Data transformations, 72, 263, 271
- Differential cost, 186, 192
- Dijkstra's algorithm, 90, 122
- Discounted cost, 9, 186, 243, 262
- Discretization, 65
- Distributed computation, 74, 120
- Duality, 65, 222

## E

- $\epsilon$ -optimal policy, 172
- Error bounds, 19, 69, 209, 213, 234, 239

## F

- Feature-based aggregation, 101
- Feature extraction, 103
- Feature vectors, 103

## G

- Gambling, 160, 173, 180
- Gauss-Seidel method, 28, 88, 208

## I

- Improper policy, 80
- Index function, 56
- Index of a project, 55
- Index rule, 55, 65
- Inventory control, 153, 179
- Irreducible Markov chain, 211

## J

- Jacobi method, 68

## L

- LLL strategy, 90
- Label correcting method, 90
- Linear programming, 49, 150, 221
- Linear quadratic problems, 150, 176-178, 228, 235

## M

- Measurability issues, 61, 172
- Minimax problems, 72
- Monotone convergence theorem, 136
- Monte-Carlo simulation, 96, 112, 120, 131, 223
- Multiarmed bandit problem, 54, 256
- Multiple-rank corrections, 48, 64

## N

- Negative DP model, 134
- Neuro-dynamic programming, 122
- Newton's method, 71

Nonstationary problems, 167

## O

Observability, 151, 228  
One-step-lookahead rule, 157, 159, 160  
Optimistic policy iteration, 116, 122

## P

Parallel computation, 64, 74, 120  
Periodic problems, 167, 171, 177, 179  
Policy, 3  
Policy evaluation, 36, 214  
Policy existence, 160, 172, 182, 226  
Policy improvement, 36, 214  
Policy iteration, 35, 71, 73, 91, 149, 186, 213, 223  
Policy iteration, approximate, 41, 91, 112, 115  
Policy iteration, modified, 39, 91  
Polynomial approximations, 102  
Positive DP model, 131  
Priority assignment, 254  
Proper policy, 80

## Q

Q-factor, 99, 132  
Q-learning, 16, 99, 122, 224, 230, 239  
Quadratic cost, 150, 176-178, 228, 235  
Queueing control, 250, 265

## R

Randomized policy, 222  
Rank-one correction, 30, 68  
Reachability, 181, 182  
Reinforcement learning, 122  
Relative cost, 186, 192  
Replacement problems, 14, 200, 276  
Riccati equation, 151, 228  
Robbins-Monro method, 98  
Routing, 257

## S

SLP strategy, 90  
Scheduling problems, 54  
Semi-Markov problems, 261  
Sequential hypothesis testing, 158  
Sequential probability ratio, 158  
Sequential space decomposition, 125  
Shortest-path problem, 78, 90, 126  
Simulation-based methods, 16, 78, 91, 222  
Stochastic approximation method, 98  
Stationary policy, 3  
Stationary policy, existence, 13, 83, 143, 160, 172, 182, 227  
Stochastic shortest paths, 78, 185, 236-239  
Stopping problems, 87, 155  
Successive approximation, 19

## T

Temporal differences, 16, 97, 115, 122, 223  
Tetris, 105, 111  
Threshold policies, 73

## U

Unbounded costs per stage, 134  
Uncontrollable state components, 105, 125  
Undiscounted problems, 134, 249  
Uniformization, 242, 274  
Unichain policy, 196

## V

Value iteration, 19, 88, 144, 186, 202, 211, 224, 238  
Value iteration, approximate, 33  
Value iteration, relative, 204, 211, 229, 232  
Value iteration, termination, 23, 89

## W

Weighted sup norm, 86, 128