

**Probability and Statistics with
Reliability, Queuing, and
Computer Science Applications**

Probability and Statistics with Reliability, Queuing, and Computer Science Applications

Kishor S. Trivedi

*Duke University
Durham, North Carolina*

WILEY

Copyright © 2016 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey Published simultaneously in Canada
Hardcover edition originally published as per part of Wiley-Interscience

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Names: Trivedi, Kishor S., 1946- author.

Title: Probability and statistics with reliability, queuing, and computer science applications / Kishor S. Trivedi, Duke University, Durham, North Carolina.

Description: Hoboken, NJ : John Wiley & Sons, Inc., [2016] | Includes bibliographical references and index.

Identifiers: LCCN 2016010100 (print) | LCCN 2016013554 (ebook) | ISBN 9781119285427 (pbk. : acid-free paper) | ISBN 9780471460817 (pdf)

Subjects: LCSH: Probabilities--Data processing. | Mathematical statistics--Data processing. | Computer algorithms. | Engineering mathematics. | Reliability Engineering. | Computer Performance. | Queuing Theory. | Fault Tolerant Computing. | Software Reliability.

Classification: LCC QA273.19.E4 T74 2016 (print) | LCC QA273.19.E4 (ebook) | DDC 519.5--dc23

LC record available at <http://lccn.loc.gov/2016010100>

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents

Preface to the Paperback Edition	ix
Preface to the Second Edition	xi
Preface to the First Edition	xiii
Acronyms	xv
About the Companion Website	xix
1 Introduction	1
1.1 Motivation	1
1.2 Probability Models	2
1.3 Sample Space	3
1.4 Events	6
1.5 Algebra of Events	7
1.6 Graphical Methods of Representing Events	11
1.7 Probability Axioms	13
1.8 Combinatorial Problems	19
1.9 Conditional Probability	24
1.10 Independence of Events	26
1.11 Bayes' Rule	38
1.12 Bernoulli Trials	47
2 Discrete Random Variables	65
2.1 Introduction	65
2.2 Random Variables and Their Event Spaces	66
2.3 The Probability Mass Function	68
2.4 Distribution Functions	70
2.5 Special Discrete Distributions	72
2.6 Analysis of Program MAX	97
2.7 The Probability Generating Function	101
2.8 Discrete Random Vectors	104
2.9 Independent Random Variables	110

3 Continuous Random Variables	121
3.1 Introduction	121
3.2 The Exponential Distribution	125
3.3 The Reliability and Failure Rate	130
3.4 Some Important Distributions	135
3.5 Functions of a Random Variable	154
3.6 Jointly Distributed Random Variables	159
3.7 Order Statistics	163
3.8 Distribution of Sums	174
3.9 Functions of Normal Random Variables	190
4 Expectation	201
4.1 Introduction	201
4.2 Moments	205
4.3 Expectation Based on Multiple Random Variables	209
4.4 Transform Methods	216
4.5 Moments and Transforms of Some Distributions	226
4.6 Computation of Mean Time to Failure	238
4.7 Inequalities and Limit Theorems	247
5 Conditional Distribution and Expectation	257
5.1 Introduction	257
5.2 Mixture Distributions	266
5.3 Conditional Expectation	273
5.4 Imperfect Fault Coverage and Reliability	280
5.5 Random Sums	290
6 Stochastic Processes	301
6.1 Introduction	301
6.2 Classification of Stochastic Processes	307
6.3 The Bernoulli Process	313
6.4 The Poisson Process	317
6.5 Renewal Processes	327
6.6 Availability Analysis	332
6.7 Random Incidence	342
6.8 Renewal Model of Program Behavior	346
7 Discrete-Time Markov Chains	351
7.1 Introduction	351
7.2 Computation of n -step Transition Probabilities	356
7.3 State Classification and Limiting Probabilities	362
7.4 Distribution of Times Between State Changes	371
7.5 Markov Modulated Bernoulli Process	373
7.6 Irreducible Finite Chains with Aperiodic States	376

7.7 * The $M/G/1$ Queuing System	391
7.8 Discrete-Time Birth–Death Processes	400
7.9 Finite Markov Chains with Absorbing States	407
8 Continuous-Time Markov Chains	421
8.1 Introduction	421
8.2 The Birth–Death Process	428
8.3 Other Special Cases of the Birth–Death Model	465
8.4 Non-Birth–Death Processes	474
8.5 Markov Chains with Absorbing States	519
8.6 Solution Techniques	541
8.7 Automated Generation	552
9 Networks of Queues	577
9.1 Introduction	577
9.2 Open Queueing Networks	582
9.3 Closed Queueing Networks	590
9.4 General Service Distribution and Multiple Job Types	620
9.5 Non-product-form Networks	628
9.6 Computing Response Time Distribution	641
9.7 Summary	654
10 Statistical Inference	661
10.1 Introduction	661
10.2 Parameter Estimation	663
10.3 Hypothesis Testing	718
11 Regression and Analysis of Variance	753
11.1 Introduction	753
11.2 Least-squares Curve Fitting	758
11.3 The Coefficients of Determination	762
11.4 Confidence Intervals in Linear Regression	765
11.5 Trend Detection and Slope Estimation	768
11.6 Correlation Analysis	771
11.7 Simple Nonlinear Regression	774
11.8 Higher-dimensional Least-squares Fit	775
11.9 Analysis of Variance	778
A Bibliography	791
A.1 Theory	791
A.2 Applications	796
B Properties of Distributions	804

C Statistical Tables	807
D Laplace Transforms	828
E Program Performance Analysis	835
Author Index	837
Subject Index	845

Preface to the Paperback Edition

Nearly 15 years have passed since the publication of the second edition of this book by John Wiley. Indian edition (of the first edition this book from 1982), published by Prentice-Hall India, is still in print and doing quite well. Asian edition (of the second edition from 2001) is published by John Wiley Asia. In 2015, a Chinese translation of the second edition has been published. I did not see the need for adding or subtracting significant amount of material to produce a new, third edition of the “bluebook” as it is popularly known. However, I took the opportunity to correct nearly 100 minor errors from the book before supplying the camera ready copy to the publisher. I have added some more terms in the subject index leading to an extra page at the end. Thus, though this is not a new edition, it is a significant new reprint. I like to remind the readers that a complete solution manual is available to instructors as well the power point slides of all chapters. Instructors may also wish to use the SHARPE software package as a pedagogic aid. (See <http://sharpe.pratt.duke.edu/> for further details.)

I wish to thank many friends for pointing out and/or help fix the errors—Javier Alonso López, Yonghuan Cao, Xiaolin Chang, Olivia Das, Jie Feng, Marcos A. M. Ferreira, Lance Fiondella, Ravi Iyer, Michael Kwok, Bharat Madan, José M. Martínez, Rivalino Matias, Jr., Kesari Mishra, Yan Qu, Dharmaraja Selvamuthu, Vibhu Sharma, Harish Sukhwani, Alex Thomasian and (late) Ranjith Vasireddy.

Kishor S. Trivedi

Preface to the Second Edition

Nearly 20 years have passed since the publication of the first edition of this book. Its Indian edition is still in print and doing quite well. In this second edition, I have thoroughly revised all the chapters. Many examples and problems are updated, and many new examples and problems have been added. There is a considerable addition of examples on system availability modeling, wireless system performance and availability modeling, software reliability modeling, and system performability modeling. New material on fault trees and stochastic Petri nets, and numerical solution techniques for Markov chains has been added. A section on the computation of response time distribution for Markovian queuing networks has also been added. Chapter 8, on continuous-time Markov chains, has undergone the most change. My research experience and the application of these methods in practice for the past 25 years (at the time of writing) have been distilled in these chapters as much as possible. I hope that the book will be of use as a classroom textbook as well as of use for practicing engineers. Researchers will also find valuable material here. I have tried to avoid adding excessive and very advanced material. Thus, for instance, I have omitted discussion of Markov regenerative processes, fluid stochastic Petri nets, and binary decision diagrams. Other topics that are omitted include material on self-similarity, large deviation theory, and diffusion approximation. The topic of hierarchical and fixed-point iterative models is covered very briefly. Modeling software fault tolerance with various kinds of correlation is also excluded.

I wish to thank many of my current students—Yonghuan Cao, Dong Chen, Dongyan Chen, Christophe Hirel, Lei Li, Yun Liu, Rajiv Poonamalli, Srinivasan Ramani, Kalyan Vaidyanathan, Wei Xie, and Liang Yin; current postdoctoral associates—Dr. Katerina Goseva-Popstojanova, Dr. Yiguang Hong, Dr. Xiaomin Ma, and Dr. Dharmaraja Selvamuthu; former students—Dr. Hoon Choi, Dr. Gianfranco Ciardo, Dr. Ricardo Fricks, Dr. Sachin Garg, Dr. Swapna Gokhale, Wei Li, Xuemei (Judith) Lou, Dr. Tong Luo, Dr. Yue Ma, Dr. Varsha Mainkar, Dr. Manish Malhotra, Anu Mohan, Dr. Jogesh Muppala, Hu Pan, Dr. Anapathur Ramesh, Dr. Robin Sahner, Kartik Sudeep, Dr. Steve Woolet, and Dr. Xinyu (Henry) Zang;

former postdoctoral associates—Dr. Hairong Sun and Dr. Bruno Tuffin; and other friends—Prof. Tadashi Dohi, Dr. Zahava Koren, Prof. Igor Kovalenko, Prof. Kang Lee, Dr. Bharat Madan and Prof. Alex Thomasian.

Kishor S. Trivedi

Preface to the First Edition

The aim of this book is to provide an introduction to probability, stochastic processes, and statistics for students of computer science, electrical/computer engineering, reliability engineering, and applied mathematics. The prerequisites are two semesters of calculus, a course on introduction to computer programming, and preferably, a course on computer organization.

I have found that the material in the book can be covered in a two-semester or three-quarter course. However, through a choice of topics, shorter courses can also be organized. I have taught the material in this book to seniors and first-year graduate students but with the text in printed form, it could be given to juniors as well.

With a specific audience in mind, I have attempted to provide examples and problems, with which the student can identify, as motivation for the probability concepts. The majority of applications are drawn from reliability analysis and performance analysis of computer systems and from probabilistic analysis of algorithms. Although there are many good texts on each of these application areas, I felt the need for a text that treats them in a balanced fashion.

Chapters 1–5 provide an introduction to probability theory. These five chapters provide the core for one semester course on introduction to applied probability. Chapters 6–9 deal with stochastic processes and their applications. These four chapters form the core of the second course with a title such as systems modeling. I have included an entire chapter on networks of queues. The last two chapters are on statistical inference and regression, respectively. I have placed the material on sampling distributions in Chapter 3 dealing with continuous random variables. Portions of the chapters on statistics can be taught with the first course and other portions in the second course.

Besides more than 200 worked examples, most sections conclude with a number of exercises. Difficult exercises are indicated by a star. A solution manual for instructors is available from the publisher.

I am indebted to the Department of Computer Science, Duke University and to Merrell Patrick for their encouragement and support during this project. The efficient typing skills of Patricia Land helped make the job of writing the book much easier than it could have been.

Many of my friends, colleagues, and students carefully read several drafts and suggested many changes that improved the readability and the accuracy of

this text. Many thanks to Robert Geist, Narayan Bhat, Satish Tripathi, John Meyer, Frank Harrell, Veena Adlakha, and Jack Stiffler for their suggestions. Joey de la Cruz and Nelson Strothers helped in the editing and the typing process very early in the project. The help by the staff of Prentice-Hall in preparation of the book is also appreciated.

I would like to thank my wife Kalpana, and my daughters Kavita and Smita for enduring my preoccupation with this work for so long. The book is dedicated to my parents.

Kishor S. Trivedi

Acronyms

APS	automatic protection switching
ARQ	automatic repeat request
ATM	asynchronous transfer mode
BCC	blocked calls cleared
BDD	binary decision diagram
BER	bit error rate
BR	base radio
BTS	base-transceiver system
CDF	cumulative distribution function
CFR	constant failure rate
CME	conditional mean exceedance
CPU	central processing unit
CRTD	conditional response time distribution
CSM	central server model
CTMC	continuous-time Markov chain
DFR	decreasing failure rate
DTMC	discrete-time Markov chain
ESS	electronic switching system
FCFS	first come, first served
FIFO	first in, first out
FIT	failure in time
FTREE	fault tree
GBN	Go Back N
GMR	general(ized) multiplication rule
GSPN	generalized stochastic Petri net
IBP	interrupted Bernoulli process
IFR	increasing failure rate
I/O	input output
IRM	independent reference model
IS	infinite server
ISP	Internet service provider
LAN	local area network
LCFS	last come, first served
LCFS-PR	LCFS-preemptive resume
LCL	lower confidence limit
LLC	logical link control

LRU	least recently used
LST	Laplace–Stieltjes transform
LT	Laplace transform
MAC	medi(um)(a) access control
MAP	Markovian arrival process
METF	mean effort to (security) failure
MGF	moment generating function
MLE	maximum-likelihood estimator
MMBP	Markov modulated Bernoulli process
MMPP	Markov modulated Poisson process
MRM	Markov reward model
MTBF	mean time between failures
MTTF	mean time to failure
MTTR	mean time to repair
MVA	mean-value analysis
NHCTMC	Nonhomogeneous CTMC
NHPP	Nonhomogeneous Poisson process
NPFQN	Non-product-form queuing network(s)
ODE	ordinary differential equation
PASTA	Poisson arrivals see time averages
PN	Petri net
pdf	probability density function
PFQN	Product-form queuing network(s)
pmf	probability mass function
prs	preemptive resume
prt	preemptive repeat
PS	processor sharing
QoS	quality of service
RBD	reliability block diagram
RG	reliability graph
RR	round robin
SDP	sum of disjoint products
SEN	shuffle exchange network
SHARPE	symbolic hierarchical automated reliability and performance evaluator
SLTF	shortest latency time first
SMP	semi-Markov process
SOR	successive overrelaxation
SPN	stochastic Petri net
SPNP	stochastic Petri net package
SREPT	software reliability estimation and prediction tool
SRGM	software reliability growth model
SRM	Software reliability model
SRN	stochastic reward net
SRPTF	shortest remaining processing time first

TDMA	time division multiple access
TLP	total loss probability
TMR	triple modular redundancy
UBT	upside-down bathtub
UCL	upper confidence limit
URTD	unconditional response time distribution
VLSI	very large scale integrated (circuits)
WFS	workstation–file server (system)

About the Companion Website

This book is accompanied by a password protected companion website for instructors only.

This website contains:

- Course Slides
- Solutions Manual

To access the material on the instructor's website simply visit www.wiley.com/go/trivedi/probabilityandstatistics2e and follow the instructions for how to register.

Chapter 1

Introduction

1.1 MOTIVATION

Computer scientists and engineers need powerful techniques to analyze algorithms and computer systems. Similarly, networking engineers need methods to analyze the behavior of protocols, routing algorithms, and congestion in networks. Computer systems and networks are subject to failure, and hence methods for their reliability and availability are needed. Many of the tools necessary for these analyses have their foundations in probability theory. For example, in the analysis of algorithm execution times, it is common to draw a distinction between the *worst-case* and the *average-case* behavior of an algorithm. The distinction is based on the fact that for certain problems, while an algorithm may require an inordinately long time to solve the least favorable instance of the problem, the average solution time is considerably shorter. When many instances of a problem have to be solved, the probabilistic (or average-case) analysis of the algorithm is likely to be more useful. Such an analysis accounts for the fact that the performance of an algorithm is dependent on the distributions of input data items. Of course, we have to specify the relevant probability distributions before the analysis can be carried out. Thus, for instance, while analyzing a sorting algorithm, a common assumption is that every permutation of the input sequence is equally likely to occur.

Similarly, if the storage is dynamically allocated, a probabilistic analysis of the storage requirement is more appropriate than a worst-case analysis. In a like fashion, a worst-case analysis of the accumulation of roundoff errors in a numerical algorithm tends to be rather pessimistic; a probabilistic analysis, although harder, is more useful.

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

When we consider the analysis of a Web server serving a large number of users, several types of random phenomena need to be accounted for. First, the arrival pattern of requests is subject to randomness due to a large population of diverse users. Second, the resource requirements of requests will likely fluctuate from request to request as well as during the execution of a single request. Finally, the resources of the Web server are subject to random failures due to environmental conditions and aging phenomena. The theory of stochastic (random) processes is very useful in evaluating various measures of system effectiveness such as throughput, response time, reliability, and availability.

Before an algorithm (or protocol) or a system can be analyzed, various probability distributions have to be specified. Where do the distributions come from? We may collect data during the actual operation of the system (or the algorithm). These measurements can be performed by hardware monitors, software monitors, or both. Such data must be analyzed and compressed to obtain the necessary distributions that drive the analytical models discussed above. Mathematical statistics provides us with techniques for this purpose, such as the **design of experiments, hypothesis testing, estimation, analysis of variance, and linear and nonlinear regression**.

1.2 PROBABILITY MODELS

Probability theory is concerned with the study of random (or chance) phenomena. Such phenomena are characterized by the fact that their future behavior is not predictable in a deterministic fashion. Nevertheless, such phenomena are usually capable of mathematical descriptions due to certain statistical regularities. This can be accomplished by constructing an idealized probabilistic model of the real-world situation. Such a model consists of a list of all possible outcomes and an assignment of their respective probabilities. The theory of probability then allows us to predict or deduce patterns of future outcomes.

Since a model is an abstraction of the real-world problem, predictions based on the model must be validated against actual measurements collected from the real phenomena. A poor validation may suggest modifications to the original model. The theory of statistics facilitates the process of validation. Statistics is concerned with the inductive process of drawing inferences about the model and its parameters based on the limited information contained in real data.

The role of probability theory is to analyze the behavior of a system or an algorithm assuming the given probability assignments and distributions. The results of this analysis are as good as the underlying assumptions. Statistics helps us in choosing these probability assignments and in the process of validating model assumptions. The behavior of the system (or the algorithm) is observed, and an attempt is made to draw inferences about the underlying unknown distributions of random variables that describe system activity. Methods of statistics, in turn, make heavy use of probability theory.

Consider the problem of predicting the number of request arrivals to a Web server in a fixed time interval $(0, t]$. A common model of this situation is to assume that the number of arrivals in this period has a particular distribution, such as the Poisson distribution (see Chapter 2). Thus we have replaced a complex physical situation by a simple model with a single unknown parameter, namely, the average arrival rate λ . With the help of probability theory we can then deduce the pattern of future arrivals. On the other hand, statistical techniques help us estimate the unknown parameter λ based on actual observations of past arrival patterns. Statistical techniques also allow us to test the validity of the Poisson model.

As another example, consider a fault-tolerant computer system with automatic error recovery capability. Model this situation as follows. The probability of successful recovery is c and probability of an abortive error is $1 - c$. The uncertainty of the physical situation is once again reduced to a simple probability model with a single unknown parameter c . In order to estimate parameter c in this model, we observe N errors out of which n are successfully recovered. A reasonable estimate of c is the relative frequency n/N , since we expect this ratio to converge to c in the limit $N \rightarrow \infty$. Note that this limit is a limit in a probabilistic sense:

$$\lim_{N \rightarrow \infty} P\left(\left|\frac{n}{N} - c\right| > \epsilon\right) = 0.$$

Axiomatic approaches to probability allow us to define such limits in a mathematically consistent fashion (e.g., see the law of large numbers in Chapter 4) and hence allow us to use relative frequencies as estimates of probabilities.

1.3 SAMPLE SPACE

Probability theory is rooted in the real-life situation where a person performs an experiment the outcome of which may not be certain. Such an experiment is called a **random experiment**. Thus, an experiment may consist of the simple process of noting whether a component is functioning properly or has failed; it may consist of determining the execution time of a program; or it may consist of determining the response time of a server request. The result of any such observations, whether they are simple “yes” or “no” answers, meter readings, or whatever, are called **outcomes** of the experiment.

Definition (Sample Space). The totality of the possible outcomes of a random experiment is called the **sample space** of the experiment and it will be denoted by the letter S .

We point out that the sample space is not determined completely by the experiment. It is partially determined by the purpose for which the experiment is carried out. If the status of two components is observed, for some purposes it is sufficient to consider only three possible outcomes: two functioning, two malfunctioning, one functioning and one malfunctioning. These



Figure 1.1. A one-dimensional sample space

three outcomes constitute the sample space S . On the other hand, we might be interested in exactly which of the components has failed, if any has failed. In this case the sample space S must be considered as four possible outcomes where the earlier single outcome of one failed, one functioning is split into two outcomes: first failed, second functioning and first functioning, second failed. Many other sample spaces can be defined if we take into account such things as type of failure and so on.

Frequently, we use a larger sample space than is strictly necessary because it is easier to use; specifically, it is always easier to discard excess information than to recover lost information. For instance, in the preceding illustration, the first sample space might be denoted $S_1 = \{0, 1, 2\}$ (where each number indicates how many components are functioning) and the second sample space might be denoted $S_2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ (where 0 = failed, 1 = functioning). Given a selection from S_2 , we can always add the two components to determine the corresponding choice from S_1 ; but, given a choice from S_1 (in particular 1), we cannot necessarily recover the corresponding choice from S_2 .

It is useful to think of the outcomes of an experiment, the **elements** of the sample space, as points in a space of one or more dimensions. For example, if an experiment consists of examining the state of a single component, it may be functioning properly (denoted by the number 1), or it may have failed (denoted by the number 0). The sample space is one-dimensional, as shown in Figure 1.1. If a system consists of two components there are four possible outcomes, as shown in the two-dimensional sample space of Figure 1.2. Here each coordinate is 0 or 1 depending on whether the corresponding component is functioning properly or has failed. In general, if a system has n components, there are 2^n possible outcomes each of which can be regarded as a point in an

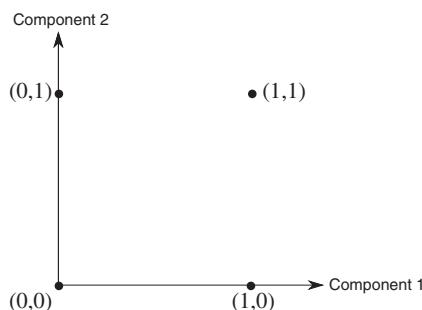


Figure 1.2. A two-dimensional sample space

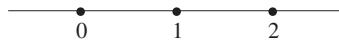


Figure 1.3. A one-dimensional sample space

n -dimensional sample space. It should be noted that the sample space used here in connection with the observation of the status of components could also serve to describe the results of other experiments; for example, the experiment of observing n successive executions of an **if** statement, with 1 denoting the execution of the **then** clause and 0 denoting the execution of the **else** clause.

The geometric configuration that is used to represent the outcomes of an experiment (e.g., Figure 1.2) is not necessarily unique. For example, we could have regarded the outcomes of the experiment of observing the two-component system to be the total number functioning, and the outcomes would be 0,1,2, as depicted in the one-dimensional sample space of Figure 1.3. Note that point 1 in Figure 1.3 corresponds to points (0,1) and (1,0) in Figure 1.2. It is often easier to use sample spaces whose elements cannot be further “subdivided”; that is, the individual elements of a sample space should not represent two or more outcomes that are distinguishable in some fashion. Thus, sample spaces like those of Figures 1.1 and 1.2 should be used in preference to sample spaces like the one in Figure 1.3.

It is convenient to classify sample spaces according to the number of elements they contain. If the set of all possible outcomes of the experiment is finite, then the associated sample space is a **finite sample space**. Thus, the sample spaces of Figures 1.1–1.3 are finite sample spaces.

To consider an example where a finite sample space does not suffice, suppose we inspect components coming out of an assembly line and that we are interested in the number inspected before we observe the first defective component. It could be the first, the second, ..., the hundredth, ..., and, for all we know, we might have to inspect a billion or more before we find a defective component. Since the number of components to be inspected before the first defective one is found is not known in advance, it is appropriate to take the sample space to be the set of natural numbers. The same sample space results for the experiment of tossing a coin until a head is observed. A sample space such as this, where the set of all outcomes can be put into a one-to-one correspondence with the natural numbers, is said to be **countably infinite**. Usually it is not necessary to distinguish between finite and countably infinite sample spaces. Therefore, if a sample space is either finite or countably infinite, we say that it is a **countable** or a **discrete sample space**.

Measurement of the time until failure of a component would have an entire interval of real numbers as possible values. Since the interval of real numbers cannot be enumerated—that is, they cannot be put into one-to-one correspondence with natural numbers—such a sample space is said to be **uncountable** or **nondenumerable**. If the elements (points) of a sample space constitute a continuum, such as all the points on a line, all the points on a line segment or

all the points in a plane, the sample space is said to be **continuous**. Certainly, no real experiment conducted using real measuring devices can ever yield such a continuum of outcomes, since there is a limit to the fineness to which any instrument can measure. However, such a sample space can often be taken as an idealization of, an approximation to, or a model of a real world situation, which may be easier to analyze than a more exact model.

Problems

1. Problems Describe a possible sample space for each of the following experiments:
 - (a) A large lot of RAM (random access memory) chips is known to contain a small number of ROM (read-only memory) chips. Three chips are chosen at random from this lot and each is checked to see whether it is a ROM or a RAM.
 - (b) A box of 10 chips is known to contain one defective and nine good chips. Three chips are chosen at random from the box and tested.
 - (c) An `if...then...else...` statement is executed 4 times.

1.4 EVENTS

An **event** is simply a collection of certain sample points, that is, a subset of the sample space. Equivalently, any statement of conditions that defines this subset is called an event. Intuitively, an event is defined as a statement whose truth or falsity is determined after the experiment. The set of all experimental outcomes (sample points) for which the statement is true defines the subset of the sample space corresponding to the event. A single performance of the experiment is known as a **trial**. Let E be an event defined on a sample space S ; that is, E is a subset of S . Let the outcome of a specific trial be denoted by s , an element of S . If s is an element of E , then we say that the event E has occurred. Only one outcome s in S can occur on any trial. However, every event that includes s will occur.

Consider the experiment of observing a two-component system and the corresponding sample space of Figure 1.2. Let event A_1 be described by the statement “Exactly one component has failed.” Then it corresponds to the subset $\{(0,1), (1,0)\}$ of the sample space. We will use the term **event** interchangeably to describe the subset or the statement. There are sixteen different subsets of this sample space with four elements, and each of these subsets defines an event. In particular, the entire sample space $S = \{(0,0), (0,1), (1,0), (1,1)\}$ is an event (called the **universal event**), and so is the null set \emptyset (called the **null** or **impossible event**). The event $\{s\}$ consisting of a single sample point will be called an **elementary event**.

Consider the experiment of observing the time to failure of a component. The sample space, in this case, may be thought of as the set of all nonnegative real numbers, or the interval $[0, \infty) = \{t \mid 0 \leq t < \infty\}$. Note that this is an

TABLE 1.1. Sample Points

$s_0 = (0, 0, 0, 0, 0)$	$s_{16} = (1, 0, 0, 0, 0)$
$s_1 = (0, 0, 0, 0, 1)$	$s_{17} = (1, 0, 0, 0, 1)$
$s_2 = (0, 0, 0, 1, 0)$	$s_{18} = (1, 0, 0, 1, 0)$
$s_3 = (0, 0, 0, 1, 1)$	$s_{19} = (1, 0, 0, 1, 1)$
$s_4 = (0, 0, 1, 0, 0)$	$s_{20} = (1, 0, 1, 0, 0)$
$s_5 = (0, 0, 1, 0, 1)$	$s_{21} = (1, 0, 1, 0, 1)$
$s_6 = (0, 0, 1, 1, 0)$	$s_{22} = (1, 0, 1, 1, 0)$
$s_7 = (0, 0, 1, 1, 1)$	$s_{23} = (1, 0, 1, 1, 1)$
$s_8 = (0, 1, 0, 0, 0)$	$s_{24} = (1, 1, 0, 0, 0)$
$s_9 = (0, 1, 0, 0, 1)$	$s_{25} = (1, 1, 0, 0, 1)$
$s_{10} = (0, 1, 0, 1, 0)$	$s_{26} = (1, 1, 0, 1, 0)$
$s_{11} = (0, 1, 0, 1, 1)$	$s_{27} = (1, 1, 0, 1, 1)$
$s_{12} = (0, 1, 1, 0, 0)$	$s_{28} = (1, 1, 1, 0, 0)$
$s_{13} = (0, 1, 1, 0, 1)$	$s_{29} = (1, 1, 1, 0, 1)$
$s_{14} = (0, 1, 1, 1, 0)$	$s_{30} = (1, 1, 1, 1, 0)$
$s_{15} = (0, 1, 1, 1, 1)$	$s_{31} = (1, 1, 1, 1, 1)$

example of a continuous sample space. Now if this component is part of a system that is required to carry out a mission of certain duration t , then an event of interest is “The component does not fail before time t .” This event may also be denoted by the set $\{x \mid x \geq t\}$, or by the interval $[t, \infty)$.

1.5 ALGEBRA OF EVENTS

Consider an example of a wireless cell with five identical channels. One possible random experiment consists of checking the system to see how many channels are currently available. Each channel is in one of two states: busy (labeled 0) and available (labeled 1). An outcome of the experiment (a point in the sample space) can be denoted by a 5-tuple of 0s and 1s. A 0 in position i of the 5-tuple indicates that channel i is busy and a 1 indicates that it is available. The sample space S has $2^5 = 32$ sample points, as shown in Table 1.1. The event E_1 described by the statement “At least four channels are available” is given by

$$\begin{aligned} E_1 &= \{(0, 1, 1, 1, 1), (1, 0, 1, 1, 1), (1, 1, 0, 1, 1), (1, 1, 1, 0, 1), \\ &\quad (1, 1, 1, 1, 0), (1, 1, 1, 1, 1)\} \\ &= \{s_{15}, s_{23}, s_{27}, s_{29}, s_{30}, s_{31}\}. \end{aligned}$$

The **complement** of this event, denoted by \overline{E}_1 , is defined to be $S - E_1$, and contains all of the sample points not contained in E_1 ; that is, $\overline{E}_1 = \{s \in S \mid s \notin E_1\}$. In our example, $\overline{E}_1 = \{s_0 \text{ through } s_{14}\}$,

s_{16} through s_{22} , s_{24} through s_{26} , s_{28} }.

 \overline{E}_1 may also be described by the statement “at most three channels are available.” Let E_2 be the event “at most four channels are available.” Then $E_2 = \{s_0 \text{ through } s_{30}\}$. The **intersection** E_3 of the two events E_1 and E_2 is denoted by $E_1 \cap E_2$ and is given by:

$$\begin{aligned} E_3 &= E_1 \cap E_2 \\ &= \{s \in S \mid s \text{ is an element of both } E_1 \text{ and } E_2\} \\ &= \{s \in S \mid s \in E_1 \text{ and } s \in E_2\} \\ &= \{s_{15}, s_{23}, s_{27}, s_{29}, s_{30}\}. \end{aligned}$$

Let E_4 be the event “channel 1 is available.” Then $E_4 = \{s_{16} \text{ through } s_{31}\}$. The **union** E_5 of the two events E_1 and E_4 is denoted by $E_1 \cup E_4$ and is given by:

$$\begin{aligned} E_5 &= E_1 \cup E_4 \\ &= \{s \in S \mid \text{either } s \in E_1 \text{ or } s \in E_4 \text{ or both}\} \\ &= \{s_{15} \text{ through } s_{31}\}. \end{aligned}$$

Note that E_1 has 6 points, E_4 has 16 points, and E_5 has 17 points. In general:

$$\begin{aligned} |E_5| &= |E_1 \cup E_4| \\ &\leq |E_1| + |E_4|. \end{aligned}$$

Here, the notation $|A|$ is used to denote the number of elements in the set A (also known as the **cardinality** of A).

Two events A and B are said to be **mutually exclusive events** or **disjoint events** provided $A \cap B$ is the null set. If A and B are mutually exclusive, then it is not possible for both events to occur on the same trial. For example, let E_6 be the event “channel 1 is busy.” Then E_4 and E_6 are mutually exclusive events since $E_4 \cap E_6 = \emptyset$.

Although the definitions of union and intersection are given for two events, we observe that they extend to any finite number of sets. However, it is customary to use a more compact notation. Thus we define

$$\begin{aligned} \bigcup_{i=1}^n E_i &= E_1 \cup E_2 \cup E_3 \cdots \cup E_n \\ &= \{s \text{ element of } S \mid s \text{ element of } E_1 \text{ or } s \text{ element of } E_2 \\ &\quad \text{or } \cdots s \text{ element of } E_n\} \\ \bigcap_{i=1}^n E_i &= E_1 \cap E_2 \cap E_3 \cap \cdots \cap E_n \\ &= \{s \text{ element of } S \mid s \text{ element of } E_1 \text{ and } s \text{ element of } E_2 \\ &\quad \text{and } \cdots s \text{ element of } E_n\} \end{aligned}$$

These definitions can also be extended to the union and intersection of a countably infinite number of sets.

The algebra of events may be fully defined by the following five laws or axioms, where A , B , and C are arbitrary sets (or events), and S is the universal set (or event):

(E1) *Commutative laws:*

$$A \cup B = B \cup A, \quad A \cap B = B \cap A.$$

(E2) *Associative laws:*

$$A \cup (B \cup C) = (A \cup B) \cup C,$$

$$A \cap (B \cap C) = (A \cap B) \cap C.$$

(E3) *Distributive laws:*

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

(E4) *Identity laws:*

$$A \cup \emptyset = A, \quad A \cap S = A.$$

(E5) *Complementation laws:*

$$A \cup \overline{A} = S, \quad A \cap \overline{A} = \emptyset.$$

Any relation that is valid in the algebra of events can be proved by using these axioms [(E1–E5)]. Some other useful relations are as follows:

(R1) *Idempotent laws:*

$$A \cup A = A, \quad A \cap A = A.$$

(R2) *Domination laws:*

$$A \cup S = S, \quad A \cap \emptyset = \emptyset.$$

(R3) *Absorption laws:*

$$A \cap (A \cup B) = A, \quad A \cup (A \cap B) = A.$$

(R4) *de Morgan's laws:*

$$\overline{(A \cup B)} = \overline{A} \cap \overline{B}, \quad \overline{(A \cap B)} = \overline{A} \cup \overline{B}.$$

$$(R5) \quad \overline{(\overline{A})} = A.$$

$$(R6) \quad A \cup (\overline{A} \cap B) = A \cup B.$$

From the complementation laws, we note that A and \overline{A} are mutually exclusive since $A \cap \overline{A} = \emptyset$. In addition, A and \overline{A} are collectively exhaustive since any point s (an element of S) is either in \overline{A} or in A . These two notions can be generalized to a list of events.

A list of events A_1, A_2, \dots, A_n is said to be composed of **mutually exclusive** events if and only if

$$A_i \cap A_j = \begin{cases} A_i & \text{if } i = j, \\ \emptyset & \text{otherwise.} \end{cases}$$

Intuitively, a list of events is composed of mutually exclusive events if there is no point in the sample space that is included in more than one event in the list.

A list of events A_1, A_2, \dots, A_n is said to be **collectively exhaustive** if and only if

$$A_1 \cup A_2 \cup \dots \cup A_n = S.$$

Given a list of events that is collectively exhaustive, each point in the sample space is included in at least one event in the list. An arbitrary list of events may be mutually exclusive, collectively exhaustive, both, or neither. For each point s in the sample space S , we may define an event $A_s = \{s\}$. The resulting list of events is mutually exclusive and collectively exhaustive (such a list of events is also called a **partition** of the sample space S). Thus, a sample space may be defined as the mutually exclusive and collectively exhaustive listing of all possible outcomes of an experiment.

Problems

1. Four components are inspected and three events are defined as follows:

A = “all four components are found defective.”

B = “exactly two components are found to be in proper working order.”

C = “at most three components are found to be defective.”

Interpret the following events:

- (a) $B \cup C$.
- (b) $B \cap C$.
- (c) $A \cup C$.
- (d) $A \cap C$.

2. Use axioms of the algebra of events to prove the relations:

- (a) $A \cup A = A$.

- (b) $A \cup S = S$.
- (c) $A \cap \emptyset = \emptyset$.
- (d) $A \cap (A \cup B) = A$.
- (e) $A \cup (\overline{A} \cap B) = A \cup B$.

1.6 GRAPHICAL METHODS OF REPRESENTING EVENTS

Venn diagrams often provide a convenient means of ascertaining relations between events of interest. Thus, for a given sample space S and the two events A and B , we have the Venn diagram shown in Figure 1.4. In this figure, the set of all points in the sample space is symbolically denoted by the ones within the rectangle. The events A and B are represented by certain regions in S .

The union of two events A and B is represented by the set of points lying in either A or B . The union of two mutually exclusive events A and B is represented by the shaded region in Figure 1.5. On the other hand, if A and B are not mutually exclusive, they might be represented by a Venn diagram like Figure 1.6. $A \cup B$ is represented by the shaded region; a portion of this shaded region is $A \cap B$ and is so labeled.

For an event A , the complement \overline{A} consists of all points in S that do not belong to A , thus \overline{A} is represented by the unshaded region in Figure 1.7. The usefulness of Venn diagrams becomes apparent when we see that the following laws of event algebra, discussed in the last section, are easily seen to hold true by reference to Figures 1.6 and 1.7:

$$\begin{aligned} A \cap S &= A, \\ A \cup S &= S, \\ \overline{(\overline{A})} &= A, \\ \overline{(A \cup B)} &= \overline{A} \cap \overline{B}, \\ \overline{(A \cap B)} &= \overline{A} \cup \overline{B}. \end{aligned}$$

Another useful graphical device is the **tree diagram**. As an example, consider the experiment of observing two successive executions of an **if** statement in a certain program. The outcome of the first execution of the **if** statement

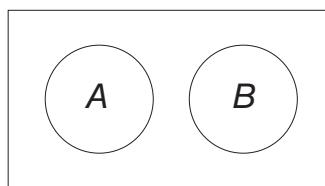


Figure 1.4. Venn diagram for sample space S and events A and B

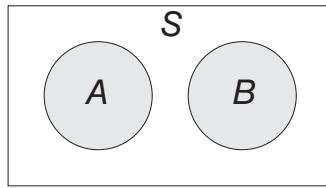


Figure 1.5. Venn diagram of disjoint events A and B

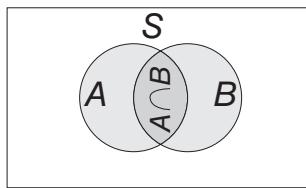


Figure 1.6. Venn diagram for two intersecting events A and B

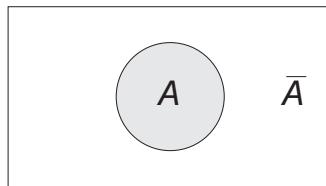
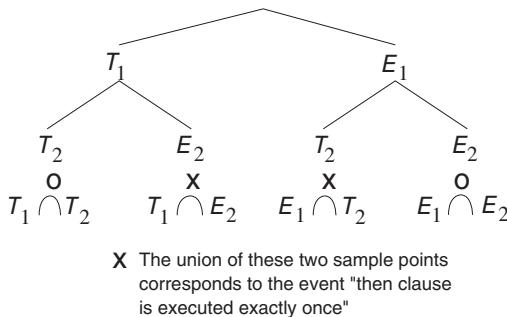
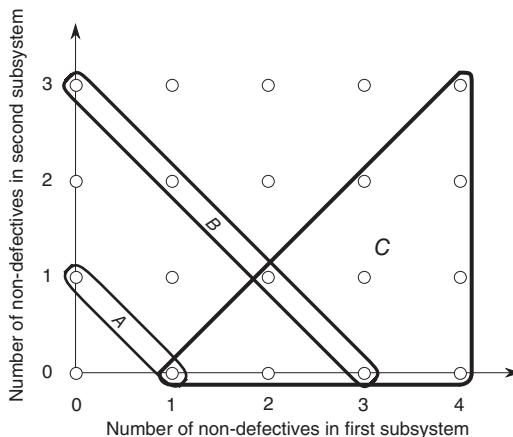


Figure 1.7. Venn diagram of A and its complement

may be the execution of the **then** clause (denoted by T_1) or the execution of the **else** clause (denoted by E_1). Similarly the outcome of the second execution is T_2 or E_2 . This is an example of a **sequential sample space** and leads to the tree diagram of Figure 1.8. We picture the experiment proceeding sequentially downward from the root. The set of all leaves of the tree is the sample space of interest. Each sample point represents the event corresponding to the intersection of all events encountered in tracing a path from the root to the leaf corresponding to the sample point. Note that the four sample points (the leaves of the tree) and their labels constitute the sample space of the experiment. However, when we deal with a sequential sample space, we normally picture the entire generating tree as well as the resulting sample space.

When the outcomes of the experiment may be expressed numerically, yet another graphical device is a coordinate system. As an example, consider a system consisting of two subsystems. The first subsystem consists of four components and the second subsystem contains three components. Assuming that we are concerned only with the total number of defective components in each

**Figure 1.8.** Tree diagram of a sequential sample space**Figure 1.9.** A two-dimensional sample space

subsystem (not with what particular components have failed), the cardinality of the sample space is $5 \cdot 4 = 20$, and the corresponding two-dimensional sample space is illustrated in Figure 1.9. The three events identified in Figure 1.9 are easily seen to be

A = “the system has exactly one non-defective component.”

B = “the system has exactly three non-defective components.”

C = “the first subsystem has more non-defective components than the second subsystem.”

1.7 PROBABILITY AXIOMS

We have seen that the physical behavior of random experiments can be modeled naturally using the concepts of events in a suitably defined sample space.

To complete our specification of the model, we shall assign **probabilities** to the events in the sample space. The probability of an event is meant to represent the “relative likelihood” that a performance of the experiment will result in the occurrence of that event. $P(A)$ will denote the probability of the event A in the sample space S .

In many engineering applications and in games of chance, the so-called relative frequency interpretation of the probability is utilized. However, such an approach is inadequate for many applications. We would like the mathematical construction of the probability measure to be independent of the intended application. This leads to an *axiomatic* treatment of the theory of probability. The theory of probability starts with the assumption that probabilities can be assigned so as to satisfy the following three basic **axioms of probability**. The assignment of probabilities is perhaps the most difficult aspect of constructing probabilistic models. Assignments are commonly based on intuition, experience, or experimentation. The theory of probability is neutral; it will make predictions regardless of these assignments. However, the results will be strongly affected by the choice of a particular assignment. Therefore if the assignments are inaccurate, the predictions of the model will be misleading and will not reflect the behavior of the “real world” problem being modeled.

Let S be a sample space of a random experiment. We use the notation $P(A)$ for the probability measure associated with event A . If the event A consists of a single sample point s then $P(A) = P(\{s\})$ will be written as $P(s)$. The probability function $P(\cdot)$ must satisfy the following Kolmogorov’s axioms:

- (A1) For any event A , $P(A) \geq 0$.
- (A2) $P(S) = 1$.
- (A3) $P(A \cup B) = P(A) + P(B)$ provided A and B are mutually exclusive events (i.e., when $A \cap B = \emptyset$).

The first axiom states that all probabilities are nonnegative real numbers. The second axiom attributes a probability of unity to the universal event S , thus providing a normalization of the probability measure (the probability of a certain event, an event that must happen, is equal to 1). The third axiom states that the probability function must be additive. These three axioms are easily seen to be consistent with our intuitive ideas of how probabilities behave.

The **principle of mathematical induction** can be used to show [using axiom (A3) as the basis of induction] that for any positive integer n the probability of the union of n mutually exclusive events A_1, A_2, \dots, A_n is equal to the sum of their probabilities:

$$P(A_1 \cup A_2 \cup \dots \cup A_n) = \sum_{i=1}^n P(A_i).$$

The three axioms, (A1)–(A3), are adequate if the sample space is finite but to deal with problems with infinite sample spaces, we need to modify axiom A3:

(A3') For any countable sequence of events $A_1, A_2, \dots, A_n, \dots$, that are mutually exclusive (that is, $A_j \cap A_k = \emptyset$ whenever $j \neq k$):

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n).$$

All of conventional probability theory follows from the three axioms (A1) through (A3') of probability measure and the 5 axioms (E1)–(E5) of the algebra of events discussed earlier. These eight axioms can be used to show several useful relations:

(Ra) For any event A , $P(\overline{A}) = 1 - P(A)$.

Proof: A and \overline{A} are mutually exclusive, and $S = A \cup \overline{A}$. Then by axioms (A2) and (A3), $1 = P(S) = P(A) + P(\overline{A})$, from which the assertion follows.

(Rb) If \emptyset is the impossible event, then $P(\emptyset) = 0$.

Proof: Observe that $\emptyset = \overline{S}$ so that the result follows from relation (Ra) and axiom (A2).

(Rc) If A and B are any events, not necessarily mutually exclusive, then $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

Proof: From the Venn diagram of Figure 1.6, we note that $A \cup B = A \cup (\overline{A} \cap B)$ and $B = (A \cap B) \cup (\overline{A} \cap B)$, where the events on the right-hand side are mutually exclusive in each equation. By axiom (A3), we obtain

$$\begin{aligned} P(A \cup B) &= P(A) + P(\overline{A} \cap B) \\ P(B) &= P(A \cap B) + P(\overline{A} \cap B). \end{aligned}$$

The second equation implies $P(\overline{A} \cap B) = P(B) - P(A \cap B)$, which, after substitution in the first equation, yields the desired assertion.

The relation (Rc) can be generalized to a formula similar to the principle of inclusion and exclusion of combinatorial mathematics [LIU 1968]:

(Rd) If A_1, A_2, \dots, A_n are any events, then

$$\begin{aligned} P\left(\bigcup_{i=1}^n A_i\right) &= P(A_1 \cup A_2 \cup \dots \cup A_n) \\ &= \sum_i P(A_i) - \sum_{1 \leq i < j \leq n} P(A_i \cap A_j) \end{aligned}$$

$$+ \sum_{1 \leq i < j < k \leq n} P(A_i \cap A_j \cap A_k) + \cdots \\ + (-1)^{n-1} P(A_1 \cap A_2 \cap \cdots \cap A_n),$$

where the successive sums are over all possible events, pairs of events, triples of events, and so on.

Proof: We prove this result by induction on the number of events n . The result (Rc) above can serve as the basis of induction. Assume inductively that (Rd) holds for a union of $n - 1$ events. Define the event $B = A_1 \cup A_2 \cup \cdots \cup A_{n-1}$. Then

$$\bigcup_{i=1}^n A_i = B \cup A_n.$$

Using the result (Rc) above, we get

$$P\left(\bigcup_{i=1}^n A_i\right) = P(B \cup A_n) \\ = P(B) + P(A_n) - P(B \cap A_n). \quad (1.1)$$

Now

$$B \cap A_n = (A_1 \cap A_n) \cup (A_2 \cap A_n) \cup \cdots \cup (A_{n-1} \cap A_n)$$

is a union of $n - 1$ events and hence, using the inductive hypothesis, we get

$$P(B \cap A_n) = P(A_1 \cap A_n) + P(A_2 \cap A_n) + \cdots + P(A_{n-1} \cap A_n) \\ - P[(A_1 \cap A_n) \cap (A_2 \cap A_n)] \\ - P[(A_1 \cap A_n) \cap (A_3 \cap A_n)] \\ - \cdots \\ + P[(A_1 \cap A_n) \cap (A_2 \cap A_n) \cap (A_3 \cap A_n)] \\ + \cdots - \cdots \\ + (-1)^{n-2} P[(A_1 \cap A_n) \cap (A_2 \cap A_n) \cap \cdots \cap (A_{n-1} \cap A_n)] \\ = P(A_1 \cap A_n) + P(A_2 \cap A_n) + \cdots + P(A_{n-1} \cap A_n) \\ - P(A_1 \cap A_2 \cap A_n) - P(A_1 \cap A_3 \cap A_n) - \cdots \\ + P(A_1 \cap A_2 \cap A_3 \cap A_n) + \cdots \\ - \cdots \\ + (-1)^{n-2} P(A_1 \cap A_2 \cap A_3 \cap \cdots \cap A_{n-1} \cap A_n). \quad (1.2)$$

Also, since $B = A_1 \cup A_2 \cup \cdots \cup A_{n-1}$ is a union of $n - 1$ events, the inductive hypothesis gives

$$P(B) = P(A_1) + P(A_2) + \cdots + P(A_{n-1})$$

$$\begin{aligned}
 & -P(A_1 \cap A_2) - P(A_1 \cap A_3) - \dots \\
 & + \dots \\
 & +(-1)^{n-2} P(A_1 \cap A_2 \cap \dots \cap A_{n-1}). \tag{1.3}
 \end{aligned}$$

Substituting (1.2) and (1.3) into (1.1), we obtain the required result.

The relation (Rd) is computationally expensive to use. A computationally simpler formula is the sum of disjoint products (SDP) formula below.

(Re)

$$\begin{aligned}
 P\left(\bigcup_{i=1}^n A_i\right) = & P(A_1) + P(\overline{A}_1 \cap A_2) + P(\overline{A}_1 \cap \overline{A}_2 \cap A_3) + \dots \\
 & + P(\overline{A}_1 \cap \overline{A}_2 \cap \dots \cap \overline{A}_{n-1} \cap A_n). \tag{1.4}
 \end{aligned}$$

The SDP formula is frequently used in reliability computations [LUO 1998]. We leave the proof as an exercise.

To avoid certain mathematical difficulties, we must place restrictions on which subsets of the sample space may be termed *events* to which probabilities can be assigned. In a given problem there will be a particular class of subsets of S that is “measurable” and will be called the “class of events” \mathcal{F} . Since we would like to perform the standard set operations on events, it is reasonable to demand that \mathcal{F} be closed under countable unions of events in \mathcal{F} as well as under complementation. A collection of subsets of a given set S that is closed under countable unions and complementation is called a σ field of subsets of S . Now a **probability space or probability system** may be defined as a triple (S, \mathcal{F}, P) , where S is a set, \mathcal{F} is a σ -field of subsets of S , and P is a probability measure on \mathcal{F} assumed to satisfy axioms (A1)–(A3').

If the sample space is discrete (finite or countable), then every subset of S can be an event belonging to \mathcal{F} . However, in the case that S is uncountable, this is no longer true. For example, let S be the interval $[0,1]$ and assume the probability assignment $P(a \leq x \leq b) = b - a$ for $0 \leq a \leq b \leq 1$. Then it can be shown that not all possible subsets of S can be assigned a probability in a manner consistent with the three axioms of P . In such cases, the smallest σ field of subsets of S containing all open and closed intervals is usually adopted as the class of events \mathcal{F} .

In summary, P is a function with domain \mathcal{F} and range $[0, 1]$, which satisfies the three axioms (A1), (A2), and (A3'). P assigns a number between 0 and 1 to any event in \mathcal{F} . In general, \mathcal{F} does not include all possible subsets of S , and the subsets (events) included in \mathcal{F} are called *measurable*. However, for our purposes, every subset of a sample space constructed here can be considered an event having a probability.

We now outline the steps of a basic procedure to be followed in solving problems [GOOD 1977]:

1. *Identify the sample space S.* The sample space S must be chosen so that all of its elements are mutually exclusive and collectively exhaustive, that is, no two elements can occur simultaneously and one element must occur on any trial. Many of the “trick” probability problems are based on some ambiguity in the problem statement or an inexact formulation of the model of a physical situation. The choice of an appropriate sample space resulting from a detailed description of the model, will do much to resolve common difficulties. Since many choices for the sample space are possible, it is advisable to use a sample space whose elements cannot be further “subdivided”—that is, all possible distinguishable outcomes of the experiment should be listed separately.
2. *Assign probabilities.* Assign probabilities to the elements in S . This assignment must be consistent with the axioms (A1), (A2), and (A3). In practice, probabilities are assigned either on the basis of estimates obtained from past experience, or on the basis of a careful analysis of conditions underlying the random experiment, or on the basis of assumptions, such as the common assumption that various outcomes in a finite sample space are equiprobable (equally likely).
3. *Identify the events of interest.* The events of interest, in a practical situation, will be described by statements. These need to be recast as subsets of the sample space. The laws of event algebra (E1)–(E5) and (R1)–(R6) may be used for any simplification. Pictorial devices such as Venn diagrams, tree diagrams, or coordinate system plots may also be used to advantage.
4. *Compute desired probabilities.* Calculate the probabilities of the events of interest using the axioms (A1), (A2), and (A3') and any derived laws such as (Ra), (Rb), (Rc), (Rd), and (Re). It is usually helpful to express the event of interest as a union of mutually exclusive points in the sample space and summing the probabilities of all points included in the union.

Example 1.1

As a simple illustration of this procedure, consider the example of the wireless cell with 5 channels.

Step 1: An appropriate sample space consists of 32 points (see Table 1.1), each represented by a 5-tuple of 0s and 1s. A 0 in position i indicates that channel i is busy and a 1 indicates that it is available.

Step 2: In the absence of detailed knowledge about the system, we assume that each sample point is equally likely. Since there are 32 sample points, we assign a probability of $\frac{1}{32}$ to each, that is, $P(s_0) = P(s_1) = \dots = P(s_{31}) = \frac{1}{32}$. It is easily seen that this assignment is consistent with the three probability axioms.

Step 3: Assume that we are required to determine the probability that a call is not blocked, given that the conference call needs at least three channels for its execution. The event E of interest, then, is “three or more channels are available.”

From the definition of the sample points, we see that

$$\begin{aligned} E &= \{s_7, s_{11}, s_{13}, s_{14}, s_{15}, s_{19}, s_{21}, s_{22}, s_{23}, s_{25} - s_{31}\} \\ &= \{s_7\} \cup \{s_{11}\} \cup \{s_{13}\} \cup \{s_{14}\} \cup \{s_{15}\} \cup \{s_{19}\} \cup \{s_{21}\} \cup \{s_{22}\} \\ &\quad \cup \{s_{23}\} \cup \{s_{25}\} \cup \{s_{26}\} \cup \{s_{27}\} \cup \{s_{28}\} \cup \{s_{29}\} \cup \{s_{30}\} \\ &\quad \cup \{s_{31}\}. \end{aligned}$$

Step 4: We have already simplified E so that it is expressed as a union of mutually exclusive events. The probability of each of these elementary events is $\frac{1}{32}$. Thus, a repeated application of axiom (A3') gives us

$$\begin{aligned} P(E) &= \sum_{s_i \in E} P(s_i) \\ &= \frac{1}{32} + \frac{1}{32} \\ &\quad + \frac{1}{32} \\ &= \frac{1}{2}. \end{aligned}$$

Alternatively, we could have noted that E consists of 16 sample points and since each 32 sample point is equally likely, $P(E) = \frac{16}{32}$.

#

Problems

1. Give the proof of the relation (Re) in this section.
2. Consider a pool of six I/O (input/output) buffers. Assume that any buffer is just as likely to be available (or occupied) as any other. Compute the probabilities associated with the following events:

A = “at least 2 but no more than 5 buffers occupied.”

B = “at least 3 but no more than 5 occupied.”

C = “all buffers available or an even number of buffers occupied.”

Also determine the probability that at least one of the events A , B , and C occurs.

3. Show that if event B is contained in event A , then $P(B) \leq P(A)$.

1.8 COMBINATORIAL PROBLEMS

If the sample space of an experiment consists of only a finite number n of sample points, or **elementary events**, then the computation of probabilities is often simple. Assume that assignment of probabilities is made such that for

s_i (an element of S), $P(s_i) = p_i$ and

$$\sum_{i=1}^n p_i = 1.$$

Since any event E consists of a certain collection of these sample points, $P(E)$ can be found, using axiom (A3'), by adding up the probabilities of the separate sample points that make up E (recall the wireless cell example of the last section).

Example 1.2

Consider the following **if** statement in a program:

```
if B then s1 else s2.
```

The random experiment consists of “observing” two successive executions of the **if** statement. The sample space consists of the four possible outcomes:

$$\begin{aligned} S &= \{(s_1, s_1), (s_1, s_2), (s_2, s_1), (s_2, s_2)\} \\ &= \{t_1, t_2, t_3, t_4\}. \end{aligned}$$

Assume that on the basis of strong experimental evidence, the following probability assignment is justified:

$$P(t_1) = 0.34, P(t_2) = 0.26, P(t_3) = 0.26, P(t_4) = 0.14.$$

The events of interest are given E_1 = “at least one execution of the statement s_1 ” and E_2 = “statement s_2 is executed the first time.” It is easy to see that

$$E_1 = \{(s_1, s_1), (s_1, s_2), (s_2, s_1)\}$$

$$= \{t_1, t_2, t_3\},$$

$$E_2 = \{(s_2, s_1), (s_2, s_2)\}$$

$$= \{t_3, t_4\},$$

$$P(E_1) = P(t_1) + P(t_2) + P(t_3) = 0.86,$$

$$P(E_2) = P(t_3) + P(t_4) = 0.4.$$

#

In the special case when $S = \{s_1, \dots, s_n\}$ and $P(s_i) = p_i = (1/n)$ (equally likely sample points), the situation is even simpler. Calculation of probabilities is then reduced to simply counting the number of sample points in the event

of interest. If the event E consists of k sample points, then

$$\begin{aligned} P(E) &= \frac{\text{number of points in } E}{\text{number of points in } S} \\ &= \frac{\text{favorable outcomes}}{\text{total outcomes}} \\ &= \frac{k}{n}. \end{aligned} \tag{1.5}$$

Example 1.3

A group of four VLSI chips consists of two good chips, labeled g_1 and g_2 , and two defective chips, labeled d_1 and d_2 . If three chips are selected at random from this group, what is the probability of the event E = “two of the three selected chips are defective”?

A natural sample space for this problem consists of all possible three chip selections from the group of four chips: $S = \{g_1g_2d_1, g_1g_2d_2, g_1d_1d_2, g_2d_1d_2\}$. It is customary to interpret the phrase “selected at random” as implying equiprobable sample points. Since the two sample points $g_1d_1d_2$ and $g_2d_1d_2$ are favorable to the event E , and since the sample space has four points, we conclude that $P(E) = \frac{2}{4} = \frac{1}{2}$.

‡

We have seen that under the equiprobability assumption, finding $P(E)$ simply involves counting the number of outcomes favorable to E . However, counting by hand may not be feasible when the sample space is large. Standard counting methods of combinatorial analysis can often be used to avoid writing down the list of favorable outcomes explicitly.

1.8.1 Ordered Samples of Size k , with Replacement

Here we are interested in counting the number of ways we can select k objects from among n objects where order is important and when the same object is allowed to be repeated any number of times (**permutations with replacement**). Alternatively, we are interested in the number of ordered sequences $(s_{i_1}, s_{i_2}, \dots, s_{i_k})$, where each s_{i_r} belongs to $\{s_1, \dots, s_n\}$. It is not difficult to see that the required number is $(n \cdot n \cdots n(k\text{times}))$, or n^k .

Example 1.4

Assume that we are interested in finding the probability that some randomly chosen k -digit decimal number is a valid k -digit octal number. The sample space, in this case, is

$$S = \{(x_1, x_2, \dots, x_k) \mid x_i \in \{0, 1, 2, \dots, 9\}\}$$

and the event of interest is

$$E = \{(x_1, x_2, \dots, x_k) \mid x_i \in \{0, 1, 2, \dots, 7\}\}.$$

By the above counting principle, $|S| = 10^k$ and $|E| = 8^k$. Now, if we assume that all the sample points are equally likely, then the required answer is

$$P(E) = \frac{|E|}{|S|} = \frac{8^k}{10^k} = \frac{4^k}{5^k}.$$

‡

1.8.2 Ordered Samples of Size k , without Replacement

The number of ordered sequences $(s_{i_1}, s_{i_2}, \dots, s_{i_k})$, where each s_{i_r} belongs to $\{s_1, \dots, s_n\}$, but repetition is not allowed (i.e., no s_i can appear more than once in the sequence), is given by

$$n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!} \quad \text{for } k = 1, 2, \dots, n.$$

This number is also known as the number of permutations of n distinct objects taken k at a time, and denoted by $P(n, k)$.

Example 1.5

Suppose we wish to find the probability that a randomly chosen three-letter sequence will not have any repeated letters.

Let $I = \{a, b, \dots, z\}$ be the alphabet of 26 letters. Then the sample space is given by

$$S = \{(\alpha, \beta, \gamma) \mid \alpha \in I, \beta \in I, \gamma \in I\}$$

and the event of interest is

$$E = \{(\alpha, \beta, \gamma) \mid \alpha \in I, \beta \in I, \gamma \in I, \alpha \neq \beta, \beta \neq \gamma, \alpha \neq \gamma\}.$$

By the abovementioned counting principle, $|E|$ is simply $P(26, 3) = 15,600$. Furthermore, $|S| = 26^3 = 17,576$. Therefore, the required answer is

$$P(E) = \frac{15,600}{17,576} = 0.8875739.$$

‡

1.8.3 Unordered Samples of Size k , without Replacement

The number of unordered sets $\{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$, where s_{i_r} ($r = 1, 2, \dots, k$) are distinct elements of $\{s_1, \dots, s_n\}$ is

$$\frac{n!}{k!(n-k)!} \quad \text{for } k = 0, 1, \dots, n.$$

This is also known as the number of combinations of n distinct objects taken k at a time, and is denoted by $\binom{n}{k}$.

Example 1.6

If a box contains 75 good VLSI chips and 25 defective chips, and 12 chips are selected at random, find the probability that at least one chip is defective.

By the counting principle described above, the number of unordered samples without replacement is $\binom{100}{12}$ and hence the size of the sample space is $|S| = \binom{100}{12}$. The event of interest is E = “at least one chip is defective.” Here we find it easier to work with the complementary event \overline{E} = “no chip is defective.” Since there are 75 good chips, the preceding counting principle yields $|\overline{E}| = \binom{75}{12}$. Then

$$\begin{aligned} P(\overline{E}) &= \frac{|\overline{E}|}{|S|} \\ &= \frac{\binom{75}{12}}{\binom{100}{12}} \\ &= \frac{75! \cdot 12! \cdot 88!}{12! \cdot 63! \cdot 100!} \\ &= \frac{75! \cdot 88!}{63! \cdot 100!}. \end{aligned}$$

Now since $P(E) = 1 - P(\overline{E})$, the required probability is easily obtained. ‡

Example 1.7

Consider a TDMA (time division multiple access) wireless system [SUN 1999], where the base transceiver system of each cell has n base repeaters [also called base radio (BR)]. Each base repeater provides m time-division-multiplexed channels. Thus, there are mn channels in the system. We note that normally a cell reserves one or more channels for signaling transfer, which resides in one of n base repeaters. However, for simplicity, we do not consider signaling channels (also called *control channels*) in this example.

A base repeater is subject to failure. In order to evaluate the impact of such a failure on the performability of the system, we should know the number of ongoing talking channels on the failed base repeater. Suppose the channels are allocated randomly to the users. Denote the total number of talking channels in the whole system as k , and the number of idle channels in the whole system as j ($j + k = mn$ always holds), when the failure occurs. Then the probability, p_i , that i talking channels reside in the failed base repeater is given by

$$p_i = \frac{\binom{m(n-1)}{k-i} \binom{m}{i}}{\binom{mn}{k}}, \text{ for } 0 \leq i \leq \min(m, k). \quad (1.6)$$

Clearly, the total number of possible combinations to have k talking channels out of mn channels is $\binom{mn}{k}$, namely, the size of the sample space, $|S|$. The

event of interest is $E = \text{"}i \text{ talking channels on the failed base repeater."}$ Now if i ($0 \leq i \leq \min(m, k)$) out of the k talking channels are on the failed base repeater, corresponding to a total of $\binom{m}{i}$ combinations, then $(k - i)$ talking channels are on the rest of the $(n - 1)$ base repeaters, which has $\binom{m(n-1)}{k-i}$ combinations. Thus, $|E| = \binom{m(n-1)}{k-i} \binom{m}{i}$. Probability p_i can now be easily obtained as $|E|/|S|$.

#

Problems

- How many even two-digit numbers can be constructed out of the digits 3, 4, 5, 6, and 7? Assume first that you may use the same digit again. Next, answer this question assuming that you cannot use a digit more than once.
- Three couples (husbands and their wives) must sit at a round table in such a way that no husband is placed next to his wife. How many configurations exist?. If seats are occupied at random, what is the probability of such a configuration?
- If a three-digit decimal number is chosen at random, find the probability that exactly k digits are ≥ 5 , for $0 \leq k \leq 3$.
- A box with 15 VLSI chips contains five defective ones. If a random sample of three chips is drawn, what is the probability that all three are defective?
- In a party of five persons, compute the probability that at least two of the persons have the same birthday (month/day), assuming a 365-day year.
- * A series of n jobs arrive at a multiprocessor computer with n processors. Assume that each of the n^n possible assignment vectors (processor for job 1, ..., processor for job n) is equally likely. Find the probability that exactly one processor will not be assigned a job.

1.9 CONDITIONAL PROBABILITY

So far, we have assumed that the only information about the outcome of a trial of a given experiment, available before the trial, is that the outcome will correspond to some point in the sample space S . With this assumption, we can compute the probability of some event A . Suppose that we are given the added information that the outcome s of a trial is contained in a subset B of the sample space, with $P(B) \neq 0$. Knowledge of the occurrence of the event B may change the probability of the occurrence of the event A . We wish to define the **conditional probability** of the event A given that the event B occurs, or the **conditional probability of A given B** , symbolically as $P(A|B)$. Given that event B has occurred, the sample point corresponding to the outcome of the trial must be in B and cannot be in \overline{B} . To reflect this partial information, we define the conditional probability of a sample point s (an element of S) by

$$P(s|B) = \begin{cases} \frac{P(s)}{P(B)} & \text{if } s \in B, \\ 0 & \text{if } s \in \overline{B}. \end{cases}$$

Thus the original probability assigned to a sample point in B is scaled up by $1/P(B)$, so that the probabilities of the sample points in B will add up to 1. Now the conditional probability of any other event, such as A , can be obtained by summing over the conditional probabilities of the sample points included in A (noting that $A = (A \cap B) \cup (A \cap \bar{B})$):

$$\begin{aligned} P(A|B) &= \sum_{s \in A} P(s|B) \\ &= \sum_{s \in A \cap \bar{B}} P(s|B) + \sum_{s \in A \cap B} P(s|B) \\ &= \sum_{s \in A \cap B} P(s|B) \\ &= \sum_{s \in A \cap B} \frac{P(s)}{P(B)} \\ &= \frac{P(A \cap B)}{P(B)}, \quad P(B) \neq 0. \end{aligned}$$

This leads us to the following definition.

Definition (Conditional Probability). The conditional probability of A given B is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

if $P(B) \neq 0$ and it is undefined otherwise.

A rearrangement of this definition yields the following **multiplication rule (MR)**:

$$P(A \cap B) = \begin{cases} P(B)P(A|B) & \text{if } P(B) \neq 0, \\ P(A)P(B|A) & \text{if } P(A) \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Example 1.8

We are given a box containing 5000 VLSI chips, 1000 of which are manufactured by company X and the rest by company Y. Ten percent of the chips made by company X are defective and 5% of the chips made by company Y are defective. If a randomly chosen chip is found to be defective, find the probability that it came from company X.

Define the events A = “chip made by company X” and B = “chip is defective.” Since out of 5000 chips, 1000 are made by company X, we conclude that $P(A) = 1000/5000 = 0.2$. Also, out of a total of 5000 chips, 300 are defective. Therefore, $P(B) = 300/5000 = 0.06$. Now the event $A \cap B$ = “the chip is made by company X and is defective.” Out of 5000 chips, 100 chips qualify for this statement. Thus

$P(A \cap B) = 100/5000 = 0.02$. Now the quantity of interest is

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{0.02}{0.06} = \frac{1}{3}.$$

Thus the knowledge of the occurrence of B has increased the probability of the occurrence of event A . Similarly we find that the knowledge of the occurrence of A has increased the chances for the occurrence of the event B , since $P(B|A) = 0.1$. In fact, note that

$$\frac{P(A|B)}{P(B|A)} = \frac{1/3}{0.1} = \frac{0.2}{0.06} = \frac{P(A)}{P(B)}.$$

This interesting property of conditional properties is easily shown to hold in general

$$\frac{P(A|B)}{P(B|A)} = \frac{P(A \cap B)/P(B)}{P(A \cap B)/P(A)} = \frac{P(A)}{P(B)}.$$

‡

Problems

- Consider four computer firms, A, B, C, D , bidding for a certain contract. A survey of past bidding success of these firms on similar contracts shows the following probabilities of winning:

$$P(A) = 0.35, P(B) = 0.15, P(C) = 0.3, P(D) = 0.2.$$

Before the decision is made to award the contract, firm B withdraws its bid. Find the new probabilities of A, C, D winning the bid.

1.10 INDEPENDENCE OF EVENTS

We have seen that it is possible for the probability of an event A to decrease or increase given that event B has occurred. If the probability of the occurrence of an event A does not change regardless of whether event B has occurred, we are likely to conclude that the two events are independent. Thus we define two events A and B to be independent if and only if

$$P(A|B) = P(A).$$

From the definition of conditional probability, we have [provided $P(A) \neq 0$ and $P(B) \neq 0$]:

$$P(A \cap B) = P(A)P(B|A) = P(B)P(A|B).$$

From this we conclude that the condition for the independence of A and B can also be given either as $P(A|B) = P(A)$ or as $P(A \cap B) = P(A)P(B)$. Note that $P(A \cap B) = P(A)P(B|A)$ (if $P(A) \neq 0$) holds regardless of whether

A and B are independent, but $P(A \cap B) = P(A)P(B)$ holds only when A and B are independent. In fact this latter condition is the usual definition of independence.

Definition (Independent Events). Events A and B are said to be independent if

$$P(A \cap B) = P(A)P(B).$$

This equation is symmetric in A and B and shows that whenever A is independent of B , so is B of A . Some authors use the phrases “stochastically independent events” or “statistically independent events” in place of just “independent events.” Note that if A and B are not independent, then $P(A \cap B)$ is computed using the multiplication rule of the last section. The abovementioned condition for independence can be derived in another way by first noting that the event A is a disjoint union of events $A \cap B$ and $A \cap \overline{B}$. Now the conditional probability of all the sample points in the latter event is zero while the conditional probability of all the sample points in the former event is increased by the factor $1/P(B)$. Therefore, for $P(A|B) = P(A)$ to hold, the decrease in probability due to points in $A \cap \overline{B}$ must be balanced by the increase in probability due to points in $A \cap B$. In other words

$$\frac{P(A \cap B)}{P(B)} - P(A \cap B) = P(A \cap \overline{B}) - 0$$

or

$$\begin{aligned} \frac{P(A \cap B)}{P(B)} &= P(A \cap \overline{B}) + P(A \cap B) \\ &= P(A). \end{aligned}$$

Example 1.9

A microcomputer system consists of a microprocessor CPU chip and a random access main memory chip. The CPU is selected from a lot of 100, 10 of which are defective, and the memory chip is selected from a lot of 300, 15 of which are defective. Define A to be the event “the selected CPU is defective,” and let B be the event “the selected memory chip is defective.” Then $P(A) = 10/100 = 0.1$, and $P(B) = 15/300 = 0.05$. Since the two chips are selected from different lots, we may expect the events A and B to be independent. This can be checked since there are $10 \cdot 15$ ways of choosing both defective chips and there are $100 \cdot 300$ ways of choosing two chips. Thus

$$\begin{aligned} P(A \cap B) &= \frac{10 \cdot 15}{100 \cdot 300} \\ &= 0.005 \\ &= 0.10 \cdot 0.05 \\ &= P(A)P(B). \end{aligned}$$

Several important points are worth noting about the concept of independence:

1. If A and B are two mutually exclusive events, then $A \cap B = \emptyset$, which implies $P(A \cap B) = 0$. Now if they are independent as well, then either $P(A) = 0$ or $P(B) = 0$.
2. If an event A is independent of itself, that is, if A and A are independent, then $P(A) = 0$ or $P(A) = 1$, since the assumption of independence yields $P(A \cap A) = P(A)P(A)$ or $P(A) = [P(A)]^2$.
3. If the events A and B are independent and the events B and C are independent, then events A and C need not be independent. In other words, the relation of independence is not a transitive relation.
4. If the events A and B are independent, then so are events \bar{A} and B , events A and \bar{B} , and events \bar{A} and \bar{B} . To show the independence of events \bar{A} and B , note that $A \cap B$ and $\bar{A} \cap B$ are mutually exclusive events whose union is B . Therefore

$$\begin{aligned} P(B) &= P(A \cap B) + P(\bar{A} \cap B) \\ &= P(A)P(B) + P(\bar{A})P(B) \end{aligned}$$

since A and B are independent. This implies that $P(\bar{A} \cap B) = P(B) - P(A)P(B) = P(B)[1 - P(A)] = P(B)P(\bar{A})$, which establishes the independence of \bar{A} and B . Independence of A and \bar{B} , and \bar{A} and \bar{B} can be shown similarly.

The concept of independence of two events can be naturally extended to a list of n events.

Definition (Independence of a Set of Events). A list of n events A_1, A_2, \dots, A_n is defined to be mutually independent if and only if for each set of k ($2 \leq k \leq n$) distinct indices i_1, i_2, \dots, i_k , which are elements of $\{1, 2, \dots, n\}$, we have

$$P(A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}) = P(A_{i_1})P(A_{i_2}) \cdots P(A_{i_k}).$$

Given that a list of events A_1, A_2, \dots, A_n is mutually independent, it is straightforward to show that for each set of distinct indices i_1, i_2, \dots, i_k , which are elements of $\{1, 2, \dots, n\}$:

$$P(B_{i_1} \cap B_{i_2} \cap \cdots \cap B_{i_k}) = P(B_{i_1})P(B_{i_2}) \cdots P(B_{i_k}) \quad (1.7)$$

where each B_{i_k} may be either A_{i_k} or \bar{A}_{i_k} . In other words, if the A_i s are independent and we replace any event by its complement, we still have independence.

By the probability axiom (A3), if a list of events is mutually exclusive, the probability of their union is the sum of their probabilities. On the other hand, if a list of events is mutually independent, the probability of their intersection is the product of their probabilities. The additive and multiplicative nature, respectively, of two event lists should be noted.

Note that it is possible to have $P(A \cap B \cap C) = P(A)P(B)P(C)$ with $P(A \cap B) \neq P(A)P(B)$, $P(A \cap C) \neq P(A)P(C)$, and $P(B \cap C) \neq P(B)P(C)$. Under these conditions, events A , B , and C are not mutually independent. Similarly, the condition $P(A_1 \cap A_2 \cdots \cap A_n) = P(A_1)P(A_2) \cdots P(A_n)$ does not imply a similar condition for any smaller family of events, and therefore this condition does not imply that events A_1, A_2, \dots, A_n are mutually independent.

Example 1.10 [ASH 1970]

Consider the experiment of rolling two dice. Let the sample space $S = \{(i, j) \mid 1 \leq i, j \leq 6\}$. Also assume that each sample point is assigned a probability of $\frac{1}{36}$. Define the events A , B , and C so that

A = “first die results in a 1, 2, or 3.”

B = “first die results in a 3, 4, or 5.”

C = “the sum of the two faces is 9.”

Then $A \cap B = \{(3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6)\}$, $A \cap C = \{(3, 6)\}$, $B \cap C = \{(3, 6), (4, 5), (5, 4)\}$, and $A \cap B \cap C = \{(3, 6)\}$. Therefore

$$P(A \cap B) = \frac{1}{6} \neq P(A)P(B) = \frac{1}{4},$$

$$P(A \cap C) = \frac{1}{36} \neq P(A)P(C) = \frac{1}{18},$$

$$P(B \cap C) = \frac{1}{12} \neq P(B)P(C) = \frac{1}{18},$$

but

$$P(A \cap B \cap C) = \frac{1}{36} = P(A)P(B)P(C).$$

#

If the events A_1, A_2, \dots, A_n are such that every pair is independent, then such events are called **pairwise independent**. It does not follow that the list of events is **mutually independent**.

Example 1.11 [ASH 1970]

Consider the above experiment of tossing two dice. Let

A = “first die results in a 1, 2, or 3.”

B = “second die results in a 4, 5, or 6.”

C = “the sum of the two faces is 7.”

Then

$$A \cap B = \{(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)\}$$

and

$$\begin{aligned} A \cap C &= B \cap C \\ &= A \cap B \cap C \\ &= \{(1, 6), (2, 5), (3, 4)\}. \end{aligned}$$

Therefore

$$\begin{aligned} P(A \cap B) &= \frac{1}{4} = P(A)P(B), \\ P(A \cap C) &= \frac{1}{12} = P(A)P(C), \\ P(B \cap C) &= \frac{1}{12} = P(B)P(C), \end{aligned}$$

but

$$P(A \cap B \cap C) = \frac{1}{12} \neq P(A)P(B)P(C) = \frac{1}{24}.$$

In this example, events A , B , and C are **pairwise independent** but not **mutually independent**.

#

We illustrate the idea of independence by considering the problem of computing reliability of so-called series-parallel systems. A **series system** is one in which all components are so interrelated that the entire system will fail if any one of its components fails. On the other hand, a **parallel system** is one that will fail only if all of its components fail. We will assume that failure events of components in a system are mutually independent.

First consider a series system of n components. For $i = 1, 2, \dots, n$, define events A_i = “component i is functioning properly.” Let the **reliability**, R_i , of component i be defined as the probability that the component is functioning properly; then $R_i = P(A_i)$. By the assumption of series connections, the system reliability:

$$\begin{aligned} R_s &= P(\text{“the system is functioning properly”}) \\ &= P(A_1 \cap A_2 \cap \dots \cap A_n) \\ &= P(A_1)P(A_2) \cdots P(A_n) \\ &= \prod_{i=1}^n R_i. \end{aligned} \tag{1.8}$$

This simple **product law of reliabilities**, applicable to series systems of independent components, demonstrates how quickly system reliability degrades with an increase in complexity. For example, if a system consists of five components each in series, each having a reliability of 0.970, then the system reliability is $0.970^5 = 0.859$. Now if the system complexity is increased so that it contained 10 similar components, its reliability would be reduced to $0.970^{10} = 0.738$. Consider what happens to system reliability when a large system like a computer system consists of tens to hundreds of thousands of components!

One way to increase the reliability of a system is to use **redundancy**. The first scheme that comes to mind is to replicate components with small reliabilities (**parallel redundancy**). First consider a system consisting of n independent components in parallel, so that it will fail to function only if all n components have failed. Define event A_i = “the component i is functioning properly” and A_p = “the parallel system of n components is functioning properly.” Also let $R_i = P(A_i)$ and $R_p = P(A_p)$. To establish a relation between A_p and the A_i values, it is easier to consider the complementary events. Thus

$$\begin{aligned}\overline{A}_p &= \text{“the parallel system has failed”} \\ &= \text{“all } n \text{ components have failed”} \\ &= \overline{A}_1 \cap \overline{A}_2 \cap \cdots \cap \overline{A}_n.\end{aligned}$$

Therefore

$$\begin{aligned}P(\overline{A}_p) &= P(\overline{A}_1 \cap \overline{A}_2 \cap \cdots \cap \overline{A}_n) \\ &= P(\overline{A}_1)P(\overline{A}_2) \cdots P(\overline{A}_n)\end{aligned}$$

by independence. Now let $F_p = 1 - R_p$ be the **unreliability** of the parallel system and similarly let $F_i = 1 - R_i$ be the unreliability of component i . Then, since A_i and \overline{A}_i are mutually exclusive and collectively exhaustive events, we have

$$\begin{aligned}1 &= P(S) \\ &= P(A_i) + P(\overline{A}_i)\end{aligned}$$

and

$$\begin{aligned}F_i &= P(\overline{A}_i) \\ &= 1 - P(A_i).\end{aligned}$$

Then

$$\begin{aligned}F_p &= P(\overline{A}_p) \\ &= \prod_{i=1}^n F_i\end{aligned}$$

and

$$\begin{aligned} R_p &= 1 - F_p \\ &= 1 - \prod_{i=1}^n (1 - R_i). \end{aligned} \quad (1.9)$$

Thus, for parallel systems of n independent components, we have a **product law of unreliabilities** analogous to the product law of reliabilities of series systems. If we have a parallel system of five components, each with a reliability of 0.970, then the system reliability is increased to

$$\begin{aligned} 1 - (1 - 0.970)^5 &= 1 - (0.03)^5 \\ &= 1 - 0.000000243 \\ &= 0.999999757. \end{aligned}$$

However, one should be aware of a **law of diminishing returns**, according to which the rate of increase in reliability with each additional component decreases rapidly as n increases. This is illustrated in Figure 1.10, where we have plotted R_p as a function of n . [This remark is easily formalized by noting that R_p is a concave function of n since $R'_p(n) = -(1 - R)^n \ln(1 - R) > 0$, and $R''_p(n) = -(1 - R)^n (\ln(1 - R))^2 < 0$.]

The basic formulas (1.8) and (1.9) for the reliability computation of series and parallel systems can be used in combination to compute the reliability of a system having both series and parallel parts (**series-parallel systems**). Consider a series-parallel system of n serial stages where stage i consists of n_i identical components in parallel. Let the reliability of each component at stage i be R_i . Assuming that all components are independent, system reliability R_{sp} can be computed from the formula

$$R_{sp} = \prod_{i=1}^n [1 - (1 - R_i)^{n_i}]. \quad (1.10)$$

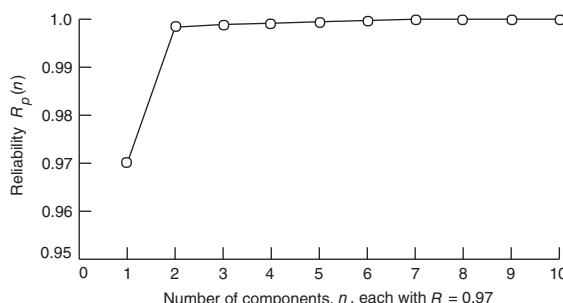


Figure 1.10. Reliability curve of a parallel redundant system

A series–parallel system can be graphically represented by a series–parallel reliability block diagram (RBD), in which components are combined into blocks in series, in parallel or in the k -out-of- n configuration (which will be introduced in the following sections). We use the following example to illustrate the use of RBD.

Example 1.12

Consider the system shown in Figure 1.11, consisting of five stages, with $n_1 = n_2 = n_5 = 1$, and $n_3 = 3$ and $n_4 = 2$. Also

$$R_1 = 0.95, R_2 = 0.99, R_3 = 0.70, R_4 = 0.75, \text{ and } R_5 = 0.9.$$

Then

$$\begin{aligned} R_{sp} &= 0.95 \cdot 0.99 \cdot (1 - (1 - 0.7)^3) \cdot (1 - (1 - 0.75)^2) \cdot 0.9 \\ &= 0.772. \end{aligned}$$

#

Fault trees provide another way to model system reliability [HENL 1981, MISR 1992, SAHN 1996]. A fault tree is a graphical representation of the combination of events that can cause the occurrence of system failure. An event is either a basic (primary) event or a logical combination of lower-level events. We assume that basic events are mutually independent and that probabilities for their occurrences are known. The occurrence of each event is denoted by a logic 1 at that node; otherwise the logic value of the node is 0. Logic value 1 for a basic event denotes failure of the corresponding component. Each gate has several inputs and one output. The inputs to a gate are either basic events or the outputs of other gates. The output of an **and** gate is a logic 1, if and only if, all of its inputs are logic 1. The output of an **or** gate is a logic 1 if one or more of its inputs are logic 1. There is a single output of the fault tree as a whole, called the *top event*, representing system failure.

Example 1.13

Consider a reliability model of alternate routing in a telephone network [BALA 1996]. The network is represented by a graph whose nodes denote the office locations

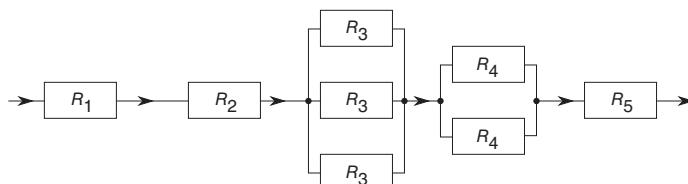


Figure 1.11. A series–parallel reliability block diagram

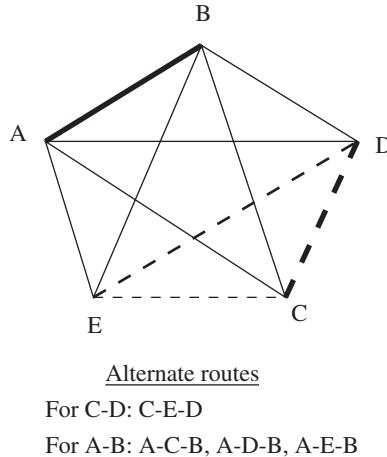


Figure 1.12. A communication network with five nodes

of a corporation and edges of the graph represent communication links between office locations as shown in Figure 1.12. The measure of interest is reliability, R , a measure of the network's ability to maintain a given set of connections. In Figure 1.12, the network is up whenever node-pairs $A-B$ and $C-D$ are both connected, either directly, or by the two-link alternate routes listed. We impose the condition that the alternate routes of the node pair $A-B$ should be disjoint from those of node pair $C-D$. We assume that link failures are mutually independent. The fault tree is shown in Figure 1.13.

In a fault tree such as that in Figure 1.13, reliabilities of inputs to an or gate multiply while unreliabilities of inputs to an and gate multiply. Hence the network reliability is given by

$$R_{\text{network}} = [1 - (1 - R_{ab})(1 - R_{ac}R_{cb})(1 - R_{ad}R_{db})(1 - R_{ae}R_{eb})] \cdot [1 - (1 - R_{cd})(1 - R_{ce}R_{ed})].$$

#

Reliability of systems with more general interconnections cannot be computed with the preceding formula. In such a case, we may obtain structure function [MISR 1992] of the system first, then compute the reliability of the system. The structure function of a system is defined as follows.

Definition (Structure Function). Let \mathbf{X} be a state vector of a system with n components so that $\mathbf{X} = (x_1, x_2, \dots, x_n)$ where

$$x_i = \begin{cases} 1 & \text{if component } i \text{ is functioning,} \\ 0 & \text{if component } i \text{ has failed.} \end{cases}$$

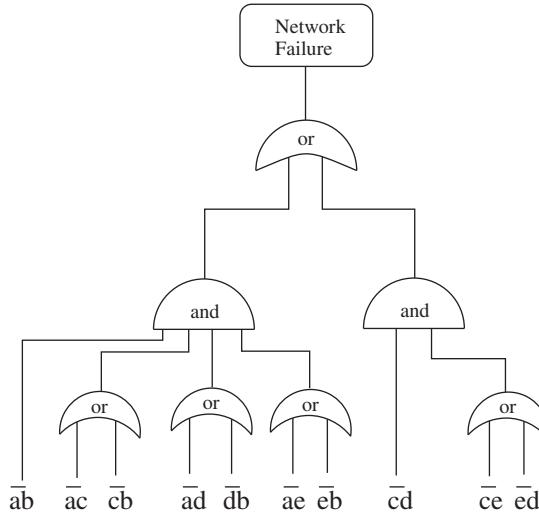


Figure 1.13. Fault tree for the communication network

The structure function $\Phi(\mathbf{X})$ is defined by

$$\Phi(\mathbf{X}) = \begin{cases} 1 & \text{if system is functioning,} \\ 0 & \text{if system has failed.} \end{cases}$$

Using the definition of system structure function, the reliability of a system can be written as

$$R = P(\Phi(\mathbf{X}) = 1).$$

Example 1.14

Consider the fault tree shown in Figure 1.14. Notice that event \overline{B}_3 is input to two gates; thus, the fault tree is said to have repeated (or shared) events. Such fault trees can no longer be solved by the simple method used for the fault tree without repeated events that we encountered in Example 1.13. For the current example, we have

$$\begin{aligned} \{\Phi = 0\} &= (\overline{A}_1 \cup (\overline{B}_1 \cap \overline{B}_3)) \cap (\overline{A}_2 \cup (\overline{B}_2 \cap \overline{B}_3)) \\ &= (\overline{A}_1 \cap \overline{A}_2) \cup (\overline{A}_1 \cap \overline{B}_2 \cap \overline{B}_3) \cup (\overline{A}_2 \cap \overline{B}_1 \cap \overline{B}_3) \cup (\overline{B}_1 \cap \overline{B}_2 \cap \overline{B}_3). \end{aligned}$$

Note that these four events are not mutually exclusive. Therefore, we cannot directly use axiom (A3), however, we could use SDP formula, i.e., relation (Re), to make them

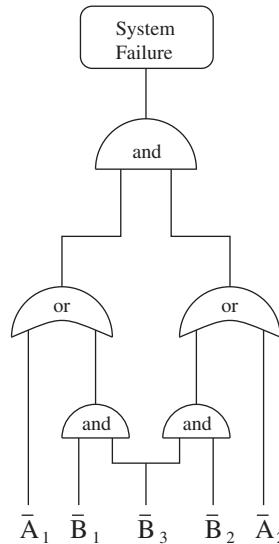


Figure 1.14. A fault tree

disjoint. Then, the reliability of the system is

$$\begin{aligned}
 R &= 1 - P(\Phi = 0) \\
 &= 1 - P((\overline{A}_1 \cap \overline{A}_2) \cup (\overline{A}_1 \cap \overline{B}_2 \cap \overline{B}_3) \cup (\overline{A}_2 \cap \overline{B}_1 \cap \overline{B}_3) \cup (\overline{B}_1 \cap \overline{B}_2 \cap \overline{B}_3)) \\
 &= 1 - P((\overline{A}_1 \cap \overline{A}_2) \cup (\overline{A}_1 \cap A_2 \cap \overline{B}_2 \cap \overline{B}_3) \cup (A_1 \cap \overline{A}_2 \cap \overline{B}_1 \cap \overline{B}_3) \\
 &\quad \cup (A_1 \cap A_2 \cap \overline{B}_1 \cap \overline{B}_2 \cap \overline{B}_3)) \\
 &= 1 - F_{A_1} F_{A_2} - F_{A_1} R_{A_2} F_{B_2} F_{B_3} - R_{A_1} F_{A_2} F_{B_1} F_{B_3} - R_{A_1} R_{A_2} F_{B_1} F_{B_2} F_{B_3}
 \end{aligned}$$

where $F_x = 1 - R_x$.

#

Starting with system structure function, there are two methods to obtain system reliability: (1) the use of inclusion–exclusion formula (Rd) and (2) the use of the SDP formula illustrated above. For an efficient implementation of the SDP method, see Luo and Trivedi [LUO 1998]. A third, even more efficient approach is based on the binary decision diagrams (BDDs) [ZANG 1999]. A fourth method is based on the use of conditioning (also called factoring) to be discussed in the next section. The BDD approach and factoring approach do not need the structure function to begin with. Further note that reliability of systems with standby redundancy cannot be computed using methods discussed in this chapter, but techniques to be discussed later in this book will enable us to do so.

Problems

1. Two towns are connected by a network of communication channels. The probability of a channel's failure-free operation is R , and channel failures are independent. Minimal level of communication between towns can be guaranteed provided at least one path containing properly functioning channels exists. Given the network of Figure 1.P.1, determine the probability that the two towns will be able to communicate. Here $\dashv \vdash$ denotes a communication channel.

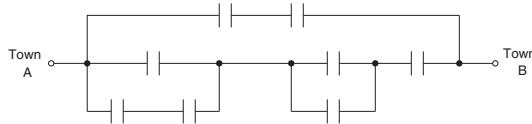


Figure 1.P.1. A network of communication channels

2. Given three components with respective reliabilities $R_1 = 0.8$, $R_2 = 0.75$, and $R_3 = 0.98$, compute the reliabilities of the three systems shown in Figure 1.P.2.

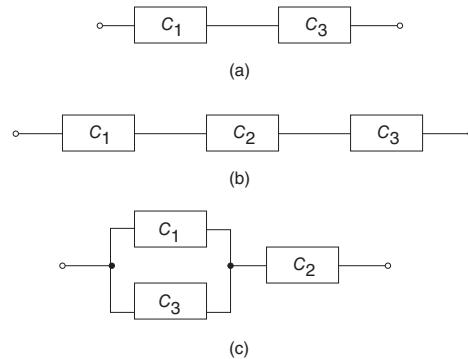


Figure 1.P.2. Reliability block diagrams

3. Determine the conditions under which an event A is independent of its subset B .
 4. *General multiplication rule* (GMR). Given a list of events A_1, A_2, \dots, A_n (not necessarily independent), show that

$$\begin{aligned} P(A_1 \cap A_2 \cap \dots \cap A_n) &= P[A_1 | (A_2 \cap A_3 \cap \dots \cap A_n)] \\ &\quad \cdot P[A_2 | (A_3 \cap \dots \cap A_n)] \\ &\quad \cdot P[A_3 | (A_4 \cap \dots \cap A_n)] \\ &\quad \dots \\ &\quad \cdot P(A_{n-1} | A_n) P(A_n), \end{aligned}$$

provided all the conditional probabilities on the right-hand side are defined.

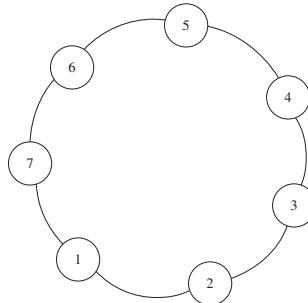


Figure 1.P.3. Lamp problem

5. Seven lamps are located as shown in Figure 1.P.3. Each lamp can fail with probability q , independently of all the others. The system is operational if no two adjacent lamps fail. Obtain an expression for system reliability.
6. Consider a base repeater in a cellular communication system with two control channels and three voice channels. Assume that the system is up so long as at least one control channel and at least one voice channel is functioning. Draw a reliability block diagram for this problem and write down an expression for system reliability. Next, draw a fault tree model for this system. Note that this fault tree has no repeated events and hence can be solved in a way similar to that for a series-parallel reliability block diagram.
7. Modify the base repeater problem above so that a control can also function as a voice channel. Draw a fault tree model for the modified problem. Notice that the fault tree has repeated events. Derive the reliability expression using the SDP method.
8. Return to Example 1.13 but now permitting a shared link $B-C$ as shown in Figure 1.P.4. Draw the fault tree for modeling the reliability for the communication network. Note that due to the shared link, the fault tree will have a shared or repeated event. Derive an expression for system reliability using SDP method as in Example 1.14.

1.11 BAYES' RULE

A given event B of probability $P(B)$ partitions the sample space S into two disjoint subsets B and \overline{B} . If we consider $S' = \{B, \overline{B}\}$ and associate the probabilities $P(B)$ and $P(\overline{B})$ to the respective points in S' , then S' is very similar to a sample space, except that there is a many-to-one correspondence between the outcomes of the experiment and the elements of S' . A space such as S' is often called an **event space**. In general, a list of n events B_1, B_2, \dots, B_n that are collectively exhaustive and mutually exclusive form an **event space**, $S' = \{B_1, B_2, \dots, B_n\}$.

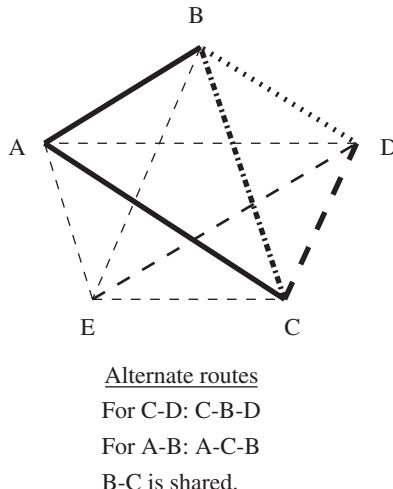


Figure 1.P.4. A modified communication network

Returning to the event space $S' = \{B, \overline{B}\}$, note that an event A is partitioned into two disjoint subsets:

$$A = (A \cap B) \cup (A \cap \overline{B}).$$

Then by axiom (A3):

$$\begin{aligned} P(A) &= P(A \cap B) + P(A \cap \overline{B}) \\ &= P(A|B)P(B) + P(A|\overline{B})P(\overline{B}) \end{aligned}$$

by definition of conditional probability.

This relation is analogous to Shannon's theorem in switching theory and can be generalized with respect to the event space $S' = \{B_1, B_2, \dots, B_n\}$:

$$P(A) = \sum_{i=1}^n P(A|B_i)P(B_i). \quad (1.11)$$

This relation is also known as the **theorem of total probability**, and is sometimes called the **rule of elimination**. This situation can be visualized by constructing a **tree diagram** (or a **probability tree**) as shown in Figure 1.15, where each branch is so labeled that the product of all branch probabilities from the root of the tree to any node equals the probability of the event represented by that node. Now $P(A)$ can be computed by summing probabilities associated with all the leaf nodes of the tree. In practice, after the experiment, a situation often arises in which the event A is known to have

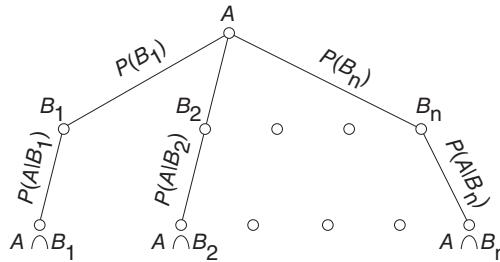


Figure 1.15. The theorem of total probability

occurred, but it is not known directly which of the mutually exclusive and collectively exhaustive events B_1, B_2, \dots, B_n has occurred. In this situation, we may be interested in evaluating $P(B_j|A)$, the conditional probability that one of these events B_j occurs, given that A occurs. By applying the definition of conditional probability followed by the use of theorem of total probability, we find that

$$\begin{aligned} P(B_j|A) &= \frac{P(B_j \cap A)}{P(A)} \\ &= \frac{P(A|B_j)P(B_j)}{\sum_i P(A|B_i)P(B_i)}. \end{aligned} \quad (1.12)$$

This relation is known as *Bayes' rule* and is useful in many applications. This rule also forms the basis of a statistical method called **Bayesian procedure**. $P(B_j|A)$ is sometimes called an **a posteriori probability**.

Example 1.15

Measurements at the North Carolina Super Computing Center (NCSU) on a certain day, indicated that the source of incoming jobs is 15% from Duke, 35% from University of North Carolina (UNC), and 50% from North Carolina State (NC State). Suppose that the probabilities that a job initiated from these universities is a multitasking job are 0.01, 0.05, and 0.02, respectively. Find the probability that a job chosen at random at NCSU is a multitasking job. Also find the probability that a randomly chosen job comes from the University of North Carolina, given that it is a multitasking job.

Define the events $B_i = \text{"job is from university } i\text{"}$ ($i = 1, 2, 3$ for Duke, UNC, and NC State, respectively), and $A = \text{"job uses multitasking."}$ Then, by the theorem of total probability, we obtain

$$\begin{aligned} P(A) &= P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + P(A|B_3)P(B_3) \\ &= (0.01) \cdot (0.15) + (0.05) \cdot (0.35) + (0.02) \cdot (0.5) \\ &= 0.029. \end{aligned}$$

Now the second event of interest is $[B_2|A]$, and from Bayes' rule:

$$\begin{aligned} P(B_2|A) &= \frac{P(A|B_2)P(B_2)}{P(A)} \\ &= \frac{0.05 \cdot 0.35}{0.029} \\ &= 0.603. \end{aligned}$$

Note that the knowledge that the job uses multitasking increases the chance that it came from UNC from 35% to about 60%. ‡

Example 1.16

A binary communication channel carries data as one of two types of signals denoted by 0 and 1. As a result of noise, a transmitted 0 is sometimes received as a 1 and a transmitted 1 is sometimes received as a 0. For a given channel, assume a probability of 0.94 that a transmitted zero is correctly received as a zero and a probability of 0.91 that a transmitted one is received as a one. Further assume a probability of 0.45 of transmitting a 0. If a signal is sent, determine the

1. Probability that a 1 is received.
2. Probability that a 0 is received.
3. Probability that a 1 was transmitted given that a 1 was received.
4. Probability that a 0 was transmitted given that a 0 was received.
5. Probability of an error.

Define events T_0 = “a 0 is transmitted” and event R_0 = “a 0 is received.” Then let $T_1 = \overline{T}_0$ = “a 1 is transmitted” and $R_1 = \overline{R}_0$ = “a 1 is received.” Then the events of interest under items 1, 2, 3, and 4 are respectively given by R_1 , R_0 , $[T_1|R_1]$, and $[T_0|R_0]$. An error in the transmitted signal is the union of the two disjoint events $[T_1 \cap R_0]$ and $[T_0 \cap R_1]$. The operation of a binary communication channel may be visualized by a **channel diagram** shown in Figure 1.16. In the given problem, we have $P(R_0|T_0) = 0.94$, $P(R_1|T_1) = 0.91$, and $P(T_0) = 0.45$. From these we get

$$\begin{aligned} P(R_1|T_0) &= P(\overline{R}_0|T_0) = 1 - P(R_0|T_0) = 0.06, \\ P(R_0|T_1) &= P(\overline{R}_1|T_1) = 1 - P(R_1|T_1) = 0.09, \\ P(T_1) &= P(\overline{T}_0) = 1 - P(T_0) = 0.55. \end{aligned}$$

Now from the theorem of total probability:

$$\begin{aligned} P(R_0) &= P(R_0|T_0)P(T_0) + P(R_0|T_1)P(T_1) \\ &= (0.94) \cdot (0.45) + (0.09) \cdot (0.55) \\ &= 0.423 + 0.0495 \\ &= 0.4725, \end{aligned}$$

$$\begin{aligned}
 P(R_1) &= P(\bar{R}_0) \\
 &= 1 - P(R_0) \\
 &= 1 - 0.4725 \\
 &= 0.5275.
 \end{aligned}$$

Using Bayes' rule, we have

$$\begin{aligned}
 P(T_1|R_1) &= \frac{P(R_1|T_1)P(T_1)}{P(R_1)} \\
 &= \frac{0.91 \cdot 0.55}{0.5275} \\
 &= 0.9488, \\
 P(T_0|R_0) &= \frac{P(R_0|T_0)P(T_0)}{P(R_0)} \\
 &= \frac{0.94 \cdot 0.45}{0.4725} \\
 &= 0.8952.
 \end{aligned}$$

Now:

$$\begin{aligned}
 P(T_1|R_0) &= P(\bar{T}_0|R_0) \\
 &= 1 - P(T_0|R_0) \\
 &= 0.1048, \\
 P(T_0|R_1) &= 1 - P(T_1|R_1) \\
 &= 0.0512
 \end{aligned}$$

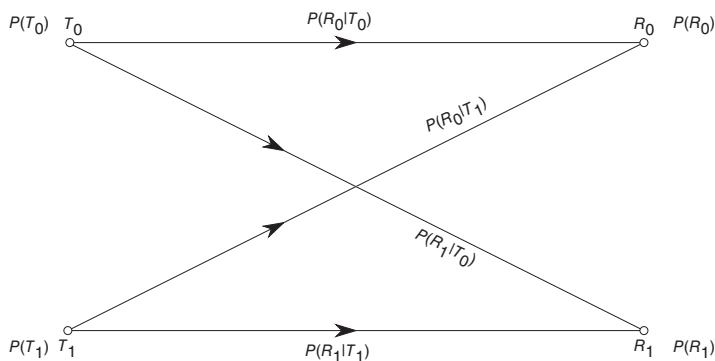


Figure 1.16. A channel diagram

and

$$\begin{aligned}
 P(\text{"error"}) &= P(T_1 \cap R_0) + P(T_0 \cap R_1) \\
 &= P(T_1|R_0)P(R_0) + P(T_0|R_1)P(R_1) \\
 &= 0.1048 \cdot 0.4725 + 0.0512 \cdot 0.5275 \\
 &= 0.0765.
 \end{aligned}$$

Alternately, the error probability can be evaluated by

$$\begin{aligned}
 P(\text{"error"}) &= P(T_1 \cap R_0) + P(T_0 \cap R_1) \\
 &= P(R_0|T_1)P(T_1) + P(R_1|T_0)P(T_0) \\
 &= 0.09 \cdot 0.55 + 0.06 \cdot 0.45 = 0.0765.
 \end{aligned}$$

[Quiz: Construct an appropriate sample space for this problem.]

#

Example 1.17

A given lot of VLSI chips contains 2% defective chips. Each chip is tested before delivery. The tester itself is not totally reliable so that

$$\begin{aligned}
 P(\text{"tester says chip is good"} | \text{"chip is actually good"}) &= 0.95, \\
 P(\text{"tester says chip is defective"} | \text{"chip is actually defective"}) &= 0.94.
 \end{aligned}$$

If a tested device is indicated to be defective, what is the probability that it is actually defective?

By Bayes' rule, we have

$$\begin{aligned}
 &P(\text{"chip is defective"} | \text{"tester says it is defective"}) \\
 &= \frac{P(\text{"tester says defective"} | \text{"chip defective"})P(\text{"chip defective"})}{P(\text{"tester says defective"} | \text{"chip defective"})P(\text{"chip defective"}) \\
 &\quad + P(\text{"tester says defective"} | \text{"chip is good"})P(\text{"chip is good"})} \\
 &= \frac{0.94 \cdot 0.02}{0.94 \cdot 0.02 + 0.05 \cdot 0.98} \\
 &= \frac{0.0188}{0.0188 + 0.049} \\
 &= \frac{0.0188}{0.0678} \\
 &= 0.2772861.
 \end{aligned}$$

#

Example 1.18

We have seen earlier how to compute the reliability of series-parallel systems. However, many systems in practice do not conform to a series-parallel structure. As an

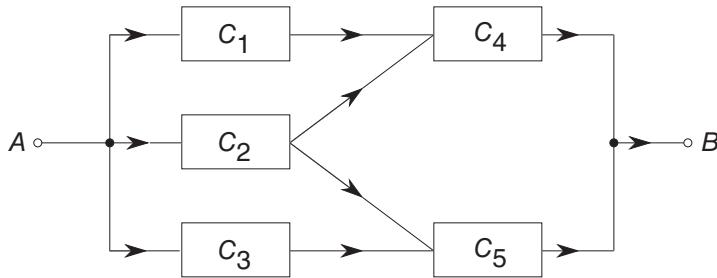


Figure 1.17. A non-series-parallel system

example, consider evaluating the reliability R of the five-component system shown in Figure 1.17. The system is said to be functioning properly only if all the components on at least one path from point A to point B are functioning properly.

Define for $i = 1, 2, \dots, 5$ event $X_i = \text{"component } i \text{ is functioning properly,"}$ and let $R_i = \text{reliability of component } i = P(X_i)$. Let $X = \text{"system functioning properly"}$ and let $R = \text{"system reliability"} = P(X)$. It is clear that X is union of four events:

$$X = (X_1 \cap X_4) \cup (X_2 \cap X_4) \cup (X_2 \cap X_5) \cup (X_3 \cap X_5). \quad (1.13)$$

These four events are not mutually exclusive. Therefore, we cannot directly use axiom (A3). Note, however, that we could use relation (Rd), which does apply to a union of intersecting events. But this method is computationally tedious for a relatively long list of events. We could use the sum of disjoint products (SDP) method (Relation Re) in this case. We illustrate the use of yet another method known as factoring or conditioning in this case. Observe that using the theorem of total probability, we have

$$\begin{aligned} P(X) &= P(X|X_2)P(X_2) + P(X|\overline{X}_2)P(\overline{X}_2) \\ &= P(X|X_2)R_2 + P(X|\overline{X}_2)(1 - R_2). \end{aligned} \quad (1.14)$$

Now to compute $P(X|X_2)$, we observe that since component C_2 is functioning, the status of components C_1 and C_3 are irrelevant. Thus, under this condition, the system is equivalent to two components C_4 and C_5 in parallel. Therefore using formula (1.9) we get

$$P(X|X_2) = 1 - (1 - R_4)(1 - R_5). \quad (1.15)$$

To compute $P(X|\overline{X}_2)$, we observe that since component C_2 is known to have malfunctioned, the resulting equivalent system is a series-parallel one whose reliability is easily computed:

$$P(X|\overline{X}_2) = 1 - (1 - R_1R_4)(1 - R_3R_5). \quad (1.16)$$

Combining equations (1.14)–(1.16), we have

$$\begin{aligned} R &= [1 - (1 - R_4)(1 - R_5)]R_2 + [1 - (1 - R_1R_4)(1 - R_3R_5)](1 - R_2) \\ &= 1 - R_2(1 - R_4)(1 - R_5) - (1 - R_2)(1 - R_1R_4)(1 - R_3R_5). \end{aligned}$$

‡

Problems

1. A technique for fault-tolerant software, suggested by Randell [RAND 1978], consists of a primary and an alternate module for each critical task, together with a test for determining whether a module performed its function correctly. Such a construct is called a **recovery block**. Define the following events:

A = “primary module functions correctly.”

B = “alternate module functions correctly.”

D = “detection test following the execution of the primary performs its task correctly.”

Assume that event pairs A and D as well as B and D are independent but events A and B are dependent. Derive an expression for the failure probability of a recovery block [HECH 1976]. (*Hint:* Use a tree diagram.)

2. Consider the non-series-parallel system of four independent components shown in Figure 1.P.5. The system is considered to be functioning properly if all components along at least one path from input to output are functioning properly. Determine an expression for system reliability as a function of component reliabilities. Also draw an equivalent fault tree model for the reliability block diagram described above.

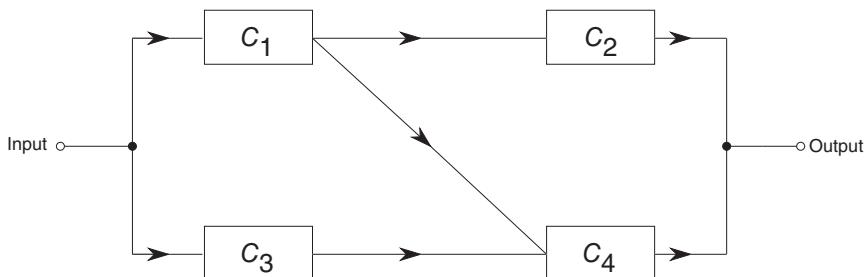


Figure 1.P.5. Another non-series-parallel system

3. A lot of components contains 0.6% defectives. Each component is subjected to a test that correctly identifies a defective, but about 2 in every 100 good components is also indicated defective. Given that a randomly chosen component

- is declared defective by the tester, compute the probability that it is actually defective.
4. A certain firm has plants A, B, and C producing respectively 35%, 15%, and 50%, of the total output. The probabilities of a nondefective product are, respectively, 0.75, 0.95, and 0.85. A customer receives a defective product. What is the probability that it came from plant C?
 5. Consider a trinary communication channel [STAR 1979] whose channel diagram is shown in Figure 1.P.6. For $i = 1, 2, 3$ let T_i denote the event “digit i is transmitted” and let R_i denote the event “digit i is received.” Assume that a 3 is transmitted 3 times more frequently than a 1, and a 2 is sent twice as often as 1. If a 1 has been received, what is the expression for the probability that a 1 was sent? Derive an expression for the probability of a transmission error.
-
- Figure 1.P.6.** A trinary communication channel: channel diagram
6. Of all the graduate students in a university, 70% are women and 30% are men. Suppose that 20% and 25% of the female and male population, respectively, smoke cigarettes. What is the probability that a randomly selected graduate student is
 - (a) A woman who smokes?
 - (b) A man who smokes?
 - (c) A smoker?
 7. Compute the reliability of the system discussed in Example 1.18 (Figure 1.17), starting from equation (1.13), first using the inclusion-exclusion formula (R_d) and then using the SDP formula (R_e). Also draw the fault tree model of this system.
 8. Yet another method of evaluating the reliability of the system such as that discussed in Example 1.16 is to use the methods of switching theory. Noting that

X_1, X_2, X_3, X_4, X_5 are Boolean variables and X is a switching function of these variables, we can draw a truth table with $2^5 = 32$ rows. Rows of the truth table represent a collection of mutually independent and collectively exhaustive events. Each row represents an elementary event that is an intersection of independent events and hence its probability can be computed. For example, the elementary event $\overline{X}_1 \cap X_2 \cap \overline{X}_3 \cap X_4 \cap X_5$ is assigned the probability $(1 - R_1)R_2(1 - R_3)R_4R_5$. Computing $P(X)$ now reduces to adding up probabilities of rows of the truth table with 1s in the function column. Use this method to compute the reliability of the system in Figure 1.17. This method is called the **state enumeration method** or the **Boolean truth table method**.

1.12 BERNOULLI TRIALS

Consider a random experiment that has two possible outcomes, “success” and “failure” (or “hit” and “miss,” or “good” and “defective,” or “digit received correctly” and “digit received incorrectly”) or the like. Let the probabilities of the two outcomes be p and q , respectively, with $p + q = 1$. Now consider the compound experiment consisting of a sequence of n independent repetitions of this experiment. Such a sequence is known as a **sequence of Bernoulli trials**. This abstract sequence models many physical situations of interest to us:

1. Observe n consecutive executions of an **if** statement, with success = “**then** clause is executed” and failure = “**else** clause is executed.”
2. Examine components produced on an assembly line, with success = “acceptable” and failure = “defective.”
3. Transmit binary digits through a communication channel, with success = “digit received correctly” and failure = “digit received incorrectly.”
4. Consider a computer system that allocates a finite quantum (or time slice) to a job scheduled for processor service, in an attempt to give fast service to requests for trivial processing. Observe n time slice terminations, with success = “job has completed processing” and failure = “job still requires processing and joins the tail end of the ready queue of processes.” This situation may be depicted as in Figure 1.18.

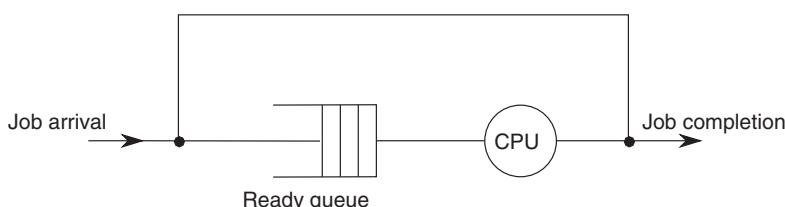


Figure 1.18. A CPU queue with time slicing

Let 0 denote failure and 1 denote success. Let S_n be the sample space of an experiment involving n Bernoulli trials, defined by

$$\begin{aligned}S_1 &= \{0, 1\}, \\S_2 &= \{(0, 0), (0, 1), (1, 0), (1, 1)\}, \\S_n &= \{2^n n\text{-tuples of } 0\text{s and } 1\text{s}\}.\end{aligned}$$

The probability assignment over the sample space S_1 is already specified: $P(0) = q \geq 0$, $P(1) = p \geq 0$, and $p + q = 1$. We wish to assign probabilities to the points in S_n .

Let A_i = “success on trial i ” and \bar{A}_i = “failure on trial i ,” then $P(A_i) = p$ and $P(\bar{A}_i) = q$. Now consider s an element of S_n such that $s = (1, 1, \dots, 1, 0, 0, \dots, 0)$ (k 1s and $(n - k)$ 0s). Then the elementary event $\{s\}$ can be written as

$$\{s\} = A_1 \cap A_2 \cap \dots \cap A_k \cap \bar{A}_{k+1} \cap \dots \cap \bar{A}_n$$

and

$$\begin{aligned}P(s) &= P(A_1 \cap A_2 \cap \dots \cap A_k \cap \bar{A}_{k+1} \cap \dots \cap \bar{A}_n) \\&= P(A_1)P(A_2) \dots P(A_k)P(\bar{A}_{k+1}) \dots P(\bar{A}_n)\end{aligned}$$

by independence. Therefore

$$P(s) = p^k q^{n-k}. \quad (1.17)$$

Similarly, any sample point with k 1s and $(n - k)$ 0s is assigned probability $p^k q^{n-k}$. Noting that there are $\binom{n}{k}$ such sample points, the probability of obtaining exactly k successes in n trials is

$$P(k) = \binom{n}{k} p^k q^{n-k}, k = 0, 1, \dots, n. \quad (1.18)$$

We may verify that (1.18) is a legitimate probability assignment over the sample space S_n since

$$\begin{aligned}\sum_{k=0}^n \binom{n}{k} p^k q^{n-k} &= (p + q)^n \\&= 1\end{aligned}$$

by the binomial theorem.

Consider the set of events $\{B_0, B_1, \dots, B_n\}$ where $B_k = \{s \in S_n \text{ such that } s \text{ has exactly } k \text{ 1s and } (n - k) \text{ 0s}\}$. It is clear that this is a mutually exclusive

and collectively exhaustive family of events. Furthermore

$$P(B_k) = \binom{n}{k} p^k q^{n-k} \geq 0 \text{ and } \sum_{k=0}^n P(B_k) = 1.$$

Therefore, this collection of events is an event space with $(n + 1)$ events. Compare this with 2^n sample points in S_n . Thus, when in a physical situation, if we are concerned not with the actual sequence of successes and failures but merely with the number of successes and the number of failures, it is profitable to use the event space rather than the original sample space.

Example 1.19

Consider a binary communication channel transmitting coded words of n bits each. Assume that the probability of successful transmission of a single bit is p (and the probability of an error is $q = 1 - p$), and that the code is capable of correcting up to e (where $e \geq 0$) errors. For example, if no coding or parity checking is used, then $e = 0$. If a single error correcting Hamming code is used then $e = 1$. For more details on this topic, see Hamming [HAMM 1980]. If we assume that the transmission of successive bits is independent, then the probability of successful word transmission is

$$\begin{aligned} P_w &= P(\text{"e or fewer errors in } n \text{ trials"}) \\ &= \sum_{i=0}^e \binom{n}{i} (1-p)^i p^{n-i}. \end{aligned}$$

‡

Example 1.20

In connection with reliability computation, we have considered series and parallel systems. Now we consider a system with n components that requires k ($\leq n$) or more components to function for the correct operation of the system. Such systems are often called k -out-of- n systems. If we let $k = n$, then we have a series system; if we let $k = 1$, then we have a system with parallel redundancy. Assume that all n components are statistically identical and function independently of each other. If we let R denote the reliability of a component (and $q = 1 - R$ gives its unreliability), then the experiment of observing the statuses of n components can be thought of as a sequence of n Bernoulli trials with the probability of success equal to R . Now the reliability of the system is

$$\begin{aligned} R_{k|n} &= P(\text{"k or more components functioning properly"}) \\ &= P\left(\bigcup_{i=k}^n \{\text{"exactly } i \text{ components functioning properly"}\}\right) \\ &= \sum_{i=k}^n P(\text{"exactly } i \text{ components functioning properly"}) \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=k}^n P(i), \\
R_{k|n} &= \sum_{i=k}^n \binom{n}{i} R^i (1-R)^{n-i}.
\end{aligned} \tag{1.19}$$

Verify that $R_{1|n} = R_p$:

$$\begin{aligned}
R_{1|n} &= \sum_{i=1}^n \binom{n}{i} R^i (1-R)^{n-i} \\
&= \sum_{i=0}^n \binom{n}{i} R^i (1-R)^{n-i} - \binom{n}{0} R^0 (1-R)^n \\
&= [R + (1-R)]^n - (1-R)^n \\
&= 1 - (1-R)^n.
\end{aligned}$$

Verify that $R_{n|n} = R_s$:

$$\begin{aligned}
R_{n|n} &= \sum_{i=n}^n \binom{n}{i} R^i (1-R)^{n-i} \\
&= \binom{n}{n} R^n (1-R)^0 \\
&= R^n.
\end{aligned}$$

‡

As another special case of formula (1.19), consider a system with triple modular redundancy, often known as TMR or a triplex system (see Figure 1.19). In such a system there are three components, two of which are required to be in working order for the system to function properly (i.e., $n = 3$ and $k = 2$). This is achieved by feeding the outputs of the three components into a majority voter. Then

$$\begin{aligned}
R_{\text{TMR}} &= \sum_{i=2}^3 \binom{3}{i} R^i (1-R)^{(3-i)} \\
&= \binom{3}{2} R^2 (1-R) + \binom{3}{3} R^3 (1-R)^0 \\
&= 3R^2 (1-R) + R^3
\end{aligned}$$

and thus

$$R_{\text{TMR}} = 3R^2 - 2R^3. \tag{1.20}$$

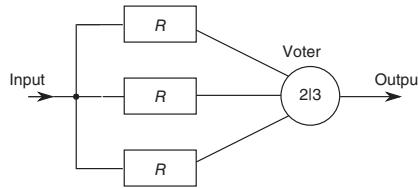


Figure 1.19. A triple modular redundant system

Note that

$$R_{\text{TMR}} = \begin{cases} > R & \text{if } R > \frac{1}{2}, \\ = R & \text{if } R = \frac{1}{2}, \\ < R & \text{if } R < \frac{1}{2}. \end{cases}$$

Thus TMR increases reliability over the simplex system only if the simplex reliability is greater than 0.5; otherwise this type of redundancy actually *decreases* reliability.

It should be noted that the voter output simply corresponds to the majority, and therefore it is possible for two or more malfunctioning units to agree, producing an erroneous voter output. Additional detection logic is required to avoid this situation. Also, the unreliability of the voter will further degrade the TMR reliability.

In the above example, we assumed that the n successive trials have the same probability of success. Now consider **nonhomogeneous Bernoulli trials**, where probability of success changes with each trial. In the reliability context, let R_i denote the reliability of the i th component for $i = 1, \dots, n$. Then the calculation is a bit more complicated [SAHN 1996]:

$$R_{k|n} = 1 - \sum_{|I| \geq k} \left(\prod_{i \in I} (1 - R_i) \right) \left(\prod_{i \notin I} R_i \right), \quad (1.21)$$

where I ranges over all choices $i_1 < i_2 < \dots < i_m$ such that $k \leq m \leq n$.

Let us still consider the TMR system with $n = 3$ and $k = 2$. However, the individual reliabilities are not identical any longer. Then, by formula (1.21), we have

$$\begin{aligned} R_{2|3} &= 1 - (1 - R_1)(1 - R_2)R_3 - R_1(1 - R_2)(1 - R_3) \\ &\quad - (1 - R_1)R_2(1 - R_3) - (1 - R_1)(1 - R_2)(1 - R_3) \\ &= R_1 R_2 + R_1 R_3 + R_2 R_3 - 2R_1 R_2 R_3 \end{aligned} \quad (1.22)$$

Example 1.21 [DOSS 2000]

Consider a BTS (base transceiver system) sector/transmitter system shown in Figure 1.20. It consists of three RF (radio frequency) carriers (transceiver and

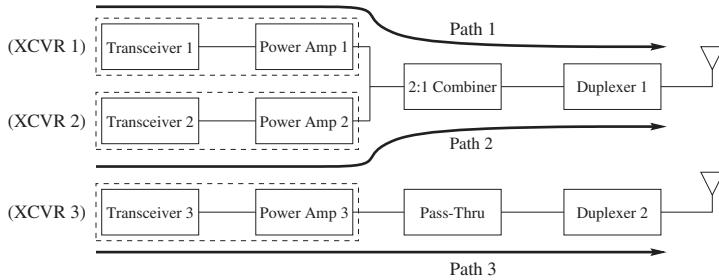


Figure 1.20. BTS sector/transmitter

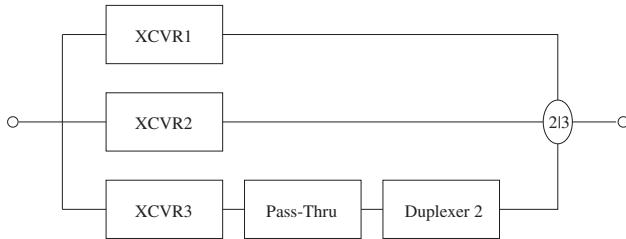


Figure 1.21. Reliability block diagram when *2:1 combiner* and *duplexer 1* are up

power amplifier) on two antennas. In order for the system to be operational, at least two functional transmitter paths are needed.

We use the factoring method to arrive at the reliability block diagram for the system. Observe that the failure of the *2:1 combiner* or *duplexer 1* would disable both path 1 and path 2, which would lead to system failure. So, we condition on these components. When both these components are functional, the system reliability is given by the RBD shown in Figure 1.21. As noted before, failure of any one of these two components results in system failure. Hence, the overall system reliability is captured by the RBD shown in Figure 1.22. If we let R_x , R_p , R_d , and R_c be the reliabilities of an XCVR, a pass-thru, a duplexer, and a combiner, then the reliabilities of XCVR1, XCVR2, XCVR3 with the “pass-thru” and duplexer 2, and the 2:1 combiner with duplexer 1 are $R_1 = R_x$, $R_2 = R_x$, $R_3 = R_x R_p R_d$, and $R_4 = R_c R_d$, respectively. Therefore, by formula (1.22), the overall system reliability is given by

$$\begin{aligned} R &= (R_1 R_2 + R_1 R_3 + R_2 R_3 - 2R_1 R_2 R_3) R_4 \\ &= (1 + 2R_p R_d - 2R_x R_p R_d) R_x^2 R_c R_d \end{aligned}$$

For a detailed discussion of various SDP methods and the factoring method of reliability computation see Rai et al. [RAI 1995].

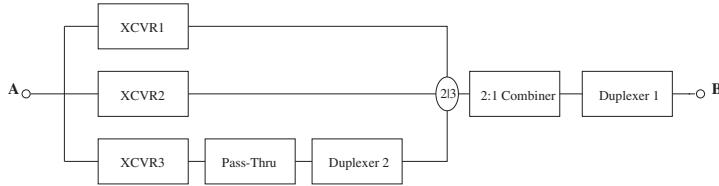


Figure 1.22. System reliability block diagram

Next, we consider **generalized Bernoulli trials**. Here we have a sequence of n independent trials, and on each trial the result is exactly one of the k possibilities b_1, b_2, \dots, b_k . On a given trial, let b_i occur with probability $p_i, i = 1, 2, \dots, k$ such that

$$p_i \geq 0 \text{ and } \sum_{i=1}^k p_i = 1.$$

The sample space S consists of all k^n n -tuples with components b_1, b_2, \dots, b_k . To a point $s \in S$

$$s = (\underbrace{b_1, b_1, \dots, b_1}_{n_1}, \underbrace{b_2, b_2, \dots, b_2}_{n_2}, \dots, \underbrace{b_k, \dots, b_k}_{n_k})$$

we assign the probability of $p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$, where $\sum_{i=1}^k n_i = n$. This is the probability assigned to any n -tuple having n_i occurrences of b_i , where $i = 1, 2, \dots, k$. The number of such n -tuples are given by the multinomial coefficient [LIU 1968]:

$$\binom{n}{n_1 \ n_2 \ \cdots \ n_k} = \frac{n!}{n_1! n_2! \cdots n_k!}.$$

As before, the probability that b_1 will occur n_1 times, b_2 will occur n_2 times, ..., and b_k will occur n_k times is given by

$$P(n_1, n_2, \dots, n_k) = \frac{n!}{n_1! n_2! \cdots n_k!} p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k} \quad (1.23)$$

and

$$\begin{aligned} \sum_{n_i \geq 0} P(n_1, n_2, \dots, n_k) &= (p_1 + p_2 + \cdots + p_k)^n \\ &= 1 \end{aligned}$$

(where $\sum n_i = n$) by the multinomial theorem.

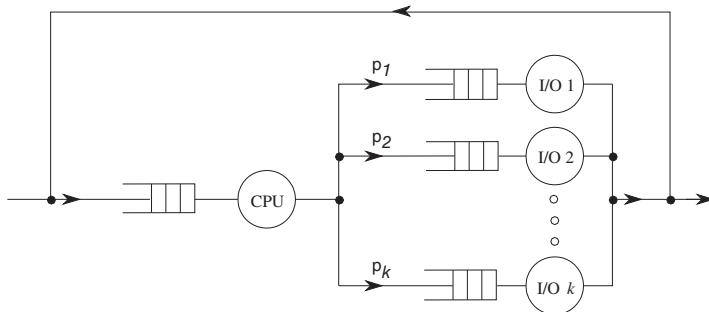


Figure 1.23. A CPU to I/O device queuing scheme

If we let $k = 2$, then generalized Bernoulli trials reduce to ordinary Bernoulli trials where b_1 = “success,” b_2 = “failure,” $p_1 = p$, $p_2 = q = 1 - p$, $n_1 = k$, and $n_2 = n - k$.

Two situations of importance are examples of generalized Bernoulli trials:

1. We are given that at the end of a CPU (central processing unit) burst, a program will request service from an I/O device i with probability p_i , where $i = 1, 2, \dots, k$ and $\sum_i p_i = 1$. If we assume that successive CPU bursts are independent of each other, then the observation of n CPU burst terminations corresponds to a sequence of generalized Bernoulli trials. This situation may be pictorially visualized by the queuing network shown in Figure 1.23.
2. If we observe n consecutive independent executions of a switch statement (see below), then we have a sequence of generalized Bernoulli trials where p_i is the probability of executing the statement group S_i on an individual trial.

```
switch( I ) {
    case 1: S1;
    case 2: S2;
    :
    case k: Sk;
}
```

Example 1.22

Out of every 100 jobs received at a server, 50 are of class 1, 30 of class 2, and 20 of class 3. A sample of 30 jobs is taken with replacement.

1. Find the probability that the sample will contain 10 jobs of each class.
2. Find the probability that there will be exactly 12 jobs of class 2.

This is an example of generalized Bernoulli trials with $k = 3$, $n = 30$, $p_1 = 0.5$, $p_2 = 0.3$, and $p_3 = 0.2$. The answer to part (1) is

$$\begin{aligned} P(10, 10, 10) &= \frac{30!}{10! \cdot 10! \cdot 10!} \cdot 0.5^{10} \cdot 0.3^{10} \cdot 0.2^{10} \\ &= 0.003278. \end{aligned}$$

The answer to part (2) is obtained more easily if we collapse class 1 and class 3 together and consider this as an example of an ordinary Bernoulli trial with $p = 0.3$ (success corresponds to a class 2 job), $q = 1 - p = 0.7$ (failure corresponds to a class 1 or class 3 job). Then the required answer is as follows:

$$\begin{aligned} P(12) &= \binom{30}{12} \cdot 0.3^{12} \cdot 0.7^{18} \\ &= \frac{30!}{12! \cdot 18!} \cdot 0.3^{12} \cdot 0.7^{18} \\ &= 0.07485. \end{aligned}$$

‡

Example 1.23

So far, we have assumed that a component is either functioning properly or it has malfunctioned. Sometimes it is useful to consider more than two states. For example, a diode functions properly with probability p_1 , develops a short circuit with probability p_2 , and develops an open circuit with probability p_3 such that $p_1 + p_2 + p_3 = 1$. Thus there are two types of malfunctions, an open circuit and a closed circuit. In order to protect against such malfunctions, we investigate three types of redundancy schemes (refer to Figure 1.24): (a) a series connection, (b) a parallel connection, and (c) a series-parallel configuration.

First we analyze the series configuration. Let s_1 , s_2 , and s_3 respectively denote the probabilities of correct functioning, a short circuit, and an open circuit for the series configuration as a whole. The experiment of observing n diodes corresponds to a sequence of n generalized Bernoulli trials. Let n_1 diodes be functioning properly, n_2 diodes be short-circuited, and n_3 diodes be open-circuited. Then the event “the series configuration is functioning properly” is described by “none of the diodes is open-circuited and at least one of the diodes is functioning properly.” This event consists of the sample points $\{(n_1, n_2, n_3) | n_1 \geq 1, n_2 \geq 0, n_3 = 0\}$,

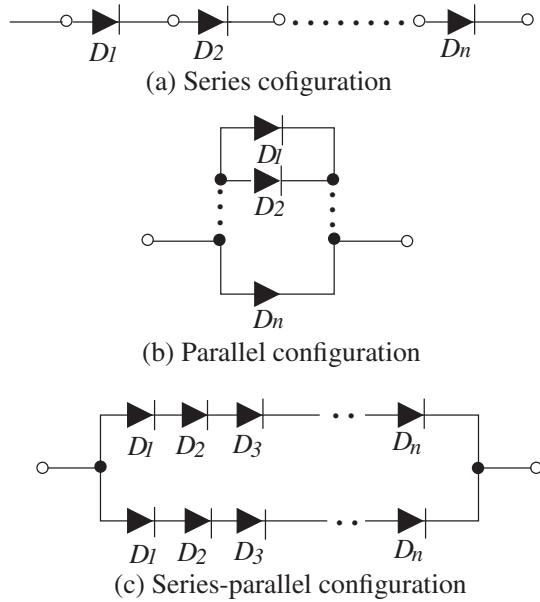


Figure 1.24. (a) Series configuration; (b) parallel configuration; (c) series-parallel configuration

$n_1 + n_2 = n\}$. Therefore

$$\begin{aligned}
 s_1 &= \sum_{\substack{n_1 \geq 1 \\ n_2 \geq 0 \\ n_1 + n_2 = n}} p(n_1, n_2, 0) \\
 &= \sum_{n_1, n_2, 0} \binom{n}{n_1, n_2, 0} p_1^{n_1} p_2^{n_2} p_3^0 \\
 &= \sum_{n_1 \geq 1} \frac{n!}{n_1!(n-n_1)!} p_1^{n_1} p_2^{n-n_1} \\
 &= \sum_{n_1=0}^n \binom{n}{n_1} p_1^{n_1} p_2^{n-n_1} - \frac{n!}{0!n!} p_1^0 p_2^n \\
 &= (p_1 + p_2)^n - p_2^n \\
 &= (1 - p_3)^n - p_2^n.
 \end{aligned}$$

Note that $(1 - p_3)^n$ is the probability that none of the diodes is open and p_2^n is the probability that all diodes are short-circuited. Similarly

$$\begin{aligned}
 s_2 &= P(\text{"Series combination is short-circuited"}) \\
 &= P(\text{"All diodes are short-circuited"})
 \end{aligned}$$

$$\begin{aligned}
&= P(\{(n_1, n_2, n_3) | n_2 = n\}) \\
&= p_2^n.
\end{aligned}$$

Also

$$\begin{aligned}
s_3 &= P(\text{"Series combination is open-circuited"}) \\
&= P(\text{"At least one diode is open-circuited"}) \\
&= p(\{(n_1, n_2, n_3) | n_3 \geq 1, n_1 + n_2 + n_3 = n\}) \\
&= 1 - P(\{(n_1, n_2, n_3) | n_3 = 0, n_1 + n_2 = n\}) \\
&= 1 - \sum_{n_1+n_2=n} \binom{n}{n_1, n_2} p_1^{n_1} p_2^{n_2} \\
&= 1 - (p_1 + p_2)^n \\
&= 1 - (1 - p_3)^n \\
&= 1 - P(\text{"no diodes are open-circuited"}).
\end{aligned}$$

Check that $s_1 + s_2 + s_3 = 1$.

Next, consider the parallel configuration, with P_i ($i = 1, 2, 3$) respectively denoting the probabilities of properly functioning, short-circuit, and open-circuit situations. Then,

$$\begin{aligned}
P_1 &= P(\text{"parallel combination working properly"}) \\
&= P(\text{"at least one diode functioning and none of them short-circuited"}) \\
&= P(\{(n_1, n_2, n_3) | n_1 \geq 1, n_2 = 0, n_1 + n_3 = n\}) \\
&= (1 - p_2)^n - p_3^n \\
&= P(\text{"no diodes short-circuited"}) - P(\text{"all diodes are open-circuited"}), \\
P_2 &= P(\{(n_1, n_2, n_3) | n_2 \geq 1, n_1 + n_2 + n_3 = n\}) \\
&= 1 - (1 - p_2)^n, \\
P_3 &= P(\{(n_1, n_2, n_3) | n_3 = n\}) \\
&= p_3^n.
\end{aligned}$$

To analyze the series-parallel configuration, we first reduce each one of the series configurations to an “equivalent” diode with respective probabilities s_1 , s_2 , and s_3 . The total configuration is then a parallel combination of two “equivalent” diodes. Thus the probability that series-parallel diode configuration functions properly is given by

$$\begin{aligned}
R_1 &= (1 - s_2)^2 - s_3^2 \\
&= s_1^2 + 2s_1 s_3 \\
&= s_1(s_1 + 2s_3)
\end{aligned}$$

$$\begin{aligned}
 &= [(1 - p_3)^n - p_2^n][(1 - p_3)^n - p_2^n + 2 - 2(1 - p_3)^n] \\
 &= [(1 - p_3)^n - p_2^n][2 - (1 - p_3)^n - p_2^n].
 \end{aligned}$$

‡

For an example of use of this technique in the context of availability analysis of VAXcluster systems, see Ibe et al. [IBE 1989]. For further study of multi-state components (as opposed to two-state or binary components) and their reliability analysis, see Zang et al. [ZANG 1999].

Problems

1. Consider the following program segment:

```

if  $B$  then
    repeat  $S_1$  until  $B_1$ 
    else
        repeat  $S_2$  until  $B_2$ 

```

Assume that $P(B = \text{true}) = p$, $P(B_1 = \text{true}) = \frac{3}{5}$, and $P(B_2 = \text{true}) = \frac{2}{5}$. Exactly one statement is common to statement groups S_1 and S_2 : write (“good day”). After many repeated executions of the preceding program segment, it has been estimated that the probability of printing exactly three “good day” messages is $\frac{3}{25}$. Derive the value of p .

2. Given that the probability of error in transmitting a bit over a communication channel is 8×10^{-4} , compute the probability of error in transmitting a block of 1024 bits. Note that this model assumes that bit errors occur at random, but in practice errors tend to occur in bursts. Actual block error rate will be considerably lower than that estimated here.
3. In order to increase the probability of correct transmission of a message over a noisy channel, a *repetition* code is often used. Assume that the “message” consists of a single bit, and that the probability of a correct transmission on a single trial is p . With a repetition code of rate $1/n$, the message is transmitted a fixed number (n) of times and a majority voter at the receiving end is used for decoding. Assuming $n = 2k + 1$, $k = 0, 1, 2, \dots$, determine the error probability P_e of a repetition code as a function of k .
4. An application requires that at least two processors in a multiprocessor system be available with more than 95% probability. The cost of a processor with 60% reliability is \$1000, and each 10% increase in reliability will cost \$800. Determine the number of processors (n) and the reliability (p) of each processor (assume that all processors have the same reliability) that minimizes the total system cost.
5. * Show that the number of terms in the multinomial expansion:

$$\left[\sum_{i=1}^k (p_i) \right]^n \text{ is } \binom{n+k-1}{n}.$$

Note that the required answer is the number of unordered sets of size n chosen from a set of k distinct objects with repetition allowed [LIU 1968].

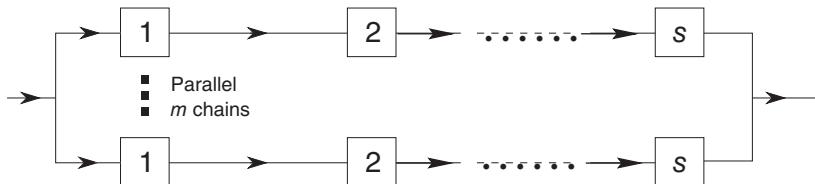
6. A communication channel receives independent pulses at the rate of 12 pulses per microsecond ($12 \mu\text{s}^{-1}$). The probability of a transmission error is 0.001 for each pulse. Compute the probabilities of
 - (a) No errors per microsecond
 - (b) One error per microsecond
 - (c) At least one error per microsecond
 - (d) Exactly two errors per microsecond
7. Plot the reliabilities of a k out of n system as a function of the simplex reliability R ($0 \leq R \leq 1$) using $n = 3$ and $k = 1, 2, 3$ [parallel redundancy, TMR (triple modular redundancy), and a series system, respectively].
8. Determine the conditions under which diode configurations in Figures 1.24(a)–(c) will improve reliability over that of a single diode. Use $n = 2$ to simplify the problem.
9. Consider a system with n capacitors in parallel. For the system to function properly, at least k -out-of- n capacitors should be functioning properly. A capacitor can fail in two modes: open and short (circuit). If a capacitor develops an open circuit, and the number of remaining working capacitors is greater than or equal to k , then the system still functions properly. If any one capacitor develops a short circuit then the system fails immediately. Given the probability of a capacitor functioning properly $p_1=0.3$, the probability of a capacitor developing a short circuit $p_2=0.4$, the probability of a capacitor developing an open circuit $p_3=0.3$, $n=10$ and $k=7$, calculate the probability of the system functioning properly.
10. Consider an example of n nonhomogeneous Bernoulli trials where a failure can occur on each trial independently, with a probability $1 - e^{-\alpha i}$ for the i th trial [KOVA 2000]. Prove that over n trials,
 - (a) $P(\text{"no failure occurs"}) = e^{-[n(n+1)/2]\alpha}$.
 - (b) $P(\text{"no more than one failure occurs"}) = e^{-[n(n+1)/2]\alpha} \left[\frac{e^\alpha - e^{-(n+1)\alpha}}{1-e^\alpha} - n + 1 \right]$.

Review Problems

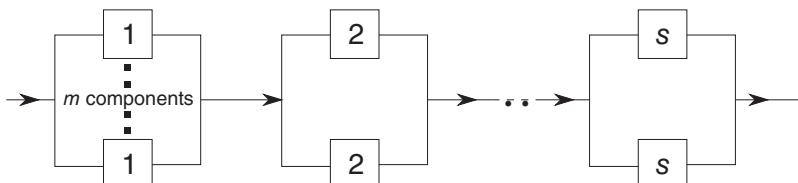
1. In the computation of TMR reliability, we assumed that when two units have failed they will both produce incorrect results and, hence after voting, the wrong answer will be produced by the TMR configuration. In the case that the two faulty units produce the opposite answers (one correct and the other incorrect) the overall result will be correct. Assuming that the probability of such a compensating error is c , derive the reliability expression for the TMR configuration.
2. See Ramamoorthy and Han [RAMA 1975]. In order to use parallel redundancy in digital logic, we have to associate an online detector with each unit giving us detector-redundant systems. However, a detector may itself fail. Compare the reliability of a three-unit detector-redundant system with a TMR system (without online detectors). Assume the reliability of a simplex unit is r , the reliability

of a detector is d and the reliability of a voter is v . A detector redundant system is said to have failed when all unit-detector pairs have failed and a unit-detector pair is a series combination of the unit and its associated detector.

3. In manufacturing a certain component, two types of defects are likely to occur with respective probabilities 0.05 and 0.1. What is the probability that a randomly chosen component
 - (a) does not have both kinds of defects?
 - (b) is defective?
 - (c) has only one kind of defect given that it is found to be defective?
4. Assume that the probability of successful transmission of a single bit over a binary communication channel is p . We desire to transmit a 4-bit word over the channel. To increase the probability of successful word transmission, we may use 7-bit Hamming code (4 data bits + 3 check bits). Such a code is known to be able to correct single-bit errors [HAMM 1980]. Derive the probabilities of successful word transmission under the two schemes and derive the condition under which the use of Hamming code will improve reliability.
5. We want to compare two different schemes of increasing reliability of a system using redundancy. Suppose that the system needs s identical components in series for proper operation. Further suppose that we are given $m \cdot s$ components. Out of the two schemes shown in Figure 1.P.7, which one will provide a higher reliability? Given that the reliability of an individual component is r , derive the expressions for the reliabilities of two configurations. For $m = 3$ and $s = 2$, compare the two expressions.



Scheme I: Redundancy at the system level



Scheme II: Redundancy at the subsystem level

Figure 1.P.7. Comparison of two redundancy schemes

6. In three boxes there are capacitors as shown in the following table:

Capacitance (in μF)	Number in box		
	1	2	3
1.0	10	90	25
0.1	50	30	80
0.01	70	90	120

An experiment consists of first randomly selecting a box (assume that each box has the same probability of selection) and then randomly selecting a capacitor from the chosen box.

- (a) What is the probability of selecting a $0.1 \mu F$ capacitor, given that box 3 is chosen?
 - (b) If a $0.1 \mu F$ capacitor is chosen, what is the probability that it came from box 1?
 - (c) List all nine conditional probabilities of capacitor selections, given certain box selections.
7. For the fault tree shown in Figure 1.P.8

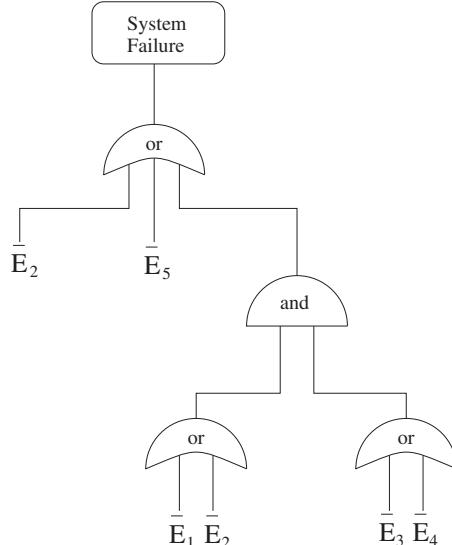


Figure 1.P.8. A fault tree

- (1) Write down the structure function.
- (2) Derive reliability expressions by
 - (a) State enumeration method
 - (b) Method of inclusion–exclusion
 - (c) Sum of disjoint products method
 - (d) Conditioning on the shared event \overline{E}_2
8. For the BTS sector/transmitter of Example 1.21, draw the equivalent fault tree, and derive reliability expressions by means of state enumeration, inclusion–exclusion, and SDP methods.

REFERENCES

- [ASH 1970] R. B. Ash, *Basic Probability Theory*, J. Wiley, New York, 1970.
- [BALA 1996] M. Balakrishnan and K. S. Trivedi, “Stochastic Petri nets for the reliability analysis of communication network applications with alternate-routing,” *Reliability Eng. Syst. Safety*, **52**, 243–259 (1996).
- [DOSS 2000] K. Doss, personal communication, 2000.
- [GOOD 1977] S. E. Goodman and S. Hedetniemi, *Introduction to the Design and Analysis of Algorithms*, McGraw-Hill, New York, 1977.
- [HAMM 1980] R. W. Hamming, *Coding and Information Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [HECH 1976] H. Hecht, “Fault-tolerant software for real-time applications,” *ACM Comput. Surv.*, 391–408 (Dec. 1976).
- [HENL 1981] E. Henley and H. Kumamoto, *Reliability Engineering and Risk Assessment*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [IBE 1989] O. Ibe, R. Howe, and K. S. Trivedi, “Approximate availability analysis of VAXCluster systems,” *IEEE Trans. Reliability*, **38**(1), 146–152 (Apr. 1989).
- [KOVA 2000] I. Kovalenko, personal communication, 2000.
- [LIU 1968] C. L. Liu, *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.
- [LUO 1998] T. Luo and K. S. Trivedi, “An improved algorithm for coherent system reliability,” *IEEE Trans. Reliability*, **47**(1), 73–78 (March 1998).
- [MISR 1992] K. B. Misra, *Reliability Analysis and Prediction: A Methodology Oriented Treatment*, Elsevier, Amsterdam, 1992.
- [RAI 1995] S. Rai, M. Veeraraghavan, and K. S. Trivedi, “A survey on efficient computation of reliability using disjoint products approach,” *Networks*, **25**(3), 147–163 (1995).
- [RAMA 1975] C. V. Ramamoorthy and Y.-W. Han, “Reliability analysis of systems with concurrent error detection,” *IEEE Trans. Comput.*, 868–878 (Sept. 1975).

- [RAND 1978] B. Randell, P. A. Lee, and P. C. Treleaven, “Reliability issues in computing system design,” *ACM Comput. Surv.*, **10**(2), 123–166 (June 1978).
- [SAHN 1996] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer System: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.
- [STAR 1979] H. Stark and F. B. Tuteur, *Modern Electrical Communications*, Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [SUN 1999] H.-R. Sun, Y. Cao, K. S. Trivedi, and J. J. Han, “Availability and performance evaluation for automatic protection switching in TDMA wireless system,” *Pacific Rim Int. Symp. Dependable Computing, Hong Kong (PRDC99)*, Dec. 1999, pp. 15–22.
- [ZANG 1999] X. Zang, H.-R. Sun, and K. S. Trivedi, “A BDD approach to dependability analysis of distributed computer systems with imperfect coverage,” in *Dependable Network Computing*, D. R. Avresky (ed.), Kluwer Academic Publishers, Amsterdam, Dec. 1999, pp. 167–190.

Chapter 2

Discrete Random Variables

2.1 INTRODUCTION

Thus far we have treated the sample space as the set of all possible outcomes of a random experiment. Some examples of sample spaces we have considered are

$$\begin{aligned}S_1 &= \{0, 1\}, \\S_2 &= \{(0, 0), (0, 1), (1, 0), (1, 1)\}, \\S_3 &= \{\text{success, failure}\}.\end{aligned}$$

Some experiments yield sample spaces whose elements are numbers, but some other experiments do not yield numerically valued elements. For mathematical convenience, it is often desirable to associate one or more numbers (in addition to probabilities) with each possible outcome of an experiment. Such numbers might naturally correspond, for instance, to the cost of each experimental outcome, the total number of defective items in a batch, or the time to failure of a component.

Through the notion of *random variables*, this and the following chapter extend our earlier work to develop methods for the study of experiments whose outcomes may be described numerically. Besides this convenience, random variables also provide a more compact description of an experiment than the finest grain description of the sample space. For example, in the inspection of manufactured products, we may be interested only in the total number of defective items and not in the nature of the defects; in a sequence of Bernoulli trials, we may be interested only in the number of successes and not in the

actual sequence of successes and failures. The notion of random variables provides us the power of abstraction and thus allows us to discard unimportant details in the outcome of an experiment. Virtually all serious probabilistic computations are performed in terms of random variables.

2.2 RANDOM VARIABLES AND THEIR EVENT SPACES

A **random variable** is a rule that assigns a numerical value to each possible outcome of an experiment. The term “random variable” is actually a misnomer, since a random variable X is really a function whose domain is the sample space S , and whose range is the set of all real numbers, \mathbb{R} . The set of all values taken by X , called the *image of X* , will then be a subset of the set of all real numbers.

Definition (Random Variable). A random variable X on a sample space S is a function $X : S \rightarrow \mathbb{R}$ that assigns a real number $X(s)$ to each sample point $s \in S$.

Example 2.1

As an example, consider a random experiment defined by a sequence of three Bernoulli trials. The sample space S consists of eight triples of 0s and 1s. We may define any number of random variables on this sample space. For our example, define a random variable X to be the total number of successes from the three trials.

The tree diagram of this sequential sample space is shown in Figure 2.1, where S_n and F_n respectively denote a success and a failure on the n th trial, and the probability of success, p , is equal to 0.5. The value of random variable X assigned to each sample point is also included.

If the outcome of one performance of the experiment were $s = (0, 1, 0)$, then the resulting experimental value of the random variable X is 1, that is $X(0, 1, 0) = 1$. Note that two or more sample points might give the same value for X (i.e., X may

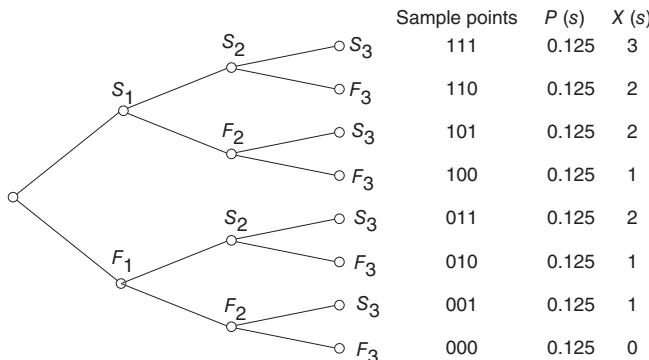


Figure 2.1. Tree diagram of a sequential sample space

not be a one-to-one function), but that two different numbers in the range cannot be assigned to the same sample point (i.e., X is a well-defined function). For example

$$X(1, 0, 0) = X(0, 1, 0) = X(0, 0, 1) = 1.$$

‡

A random variable partitions its sample space into a mutually exclusive and collectively exhaustive set of events. Thus for a random variable X and a real number x , we define the event A_x [commonly called the **inverse image** of the set $\{x\}$] to be the subset of S consisting of all sample points s to which the random variable X assigns the value x :

$$A_x = \{s \in S | X(s) = x\}.$$

It is clear that $A_x \cap A_y = \emptyset$ if $x \neq y$, and that

$$\bigcup_{x \in \Re} A_x = S$$

(see problem 1 at the end of this section). Thus the collection of events A_x for all x defines an **event space**. We may find it more convenient to work in this event space (rather than the original sample space), provided our only interest in performing the experiment has to do with the resulting experimental value of random variable X . The notation $[X = x]$ will be used as an abbreviation for the event A_x . Thus

$$[X = x] = \{s \in S | X(s) = x\}.$$

In Example 2.1 the random variable X defines four events:

$$\begin{aligned} A_0 &= \{s \in S | X(s) = 0\} = \{(0, 0, 0)\}, \\ A_1 &= \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}, \\ A_2 &= \{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}, \\ A_3 &= \{(1, 1, 1)\}. \end{aligned}$$

For all values of x outside the image of X (i.e., values of x other than 0, 1, 2, 3), A_x is the null set. The resulting event space contains four event points (see Figure 2.2). For a sequence of n Bernoulli trials, the event space defined by X will have $(n + 1)$ points, compared with 2^n sample points in the original sample space!

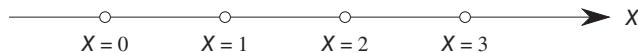


Figure 2.2. Event space for three Bernoulli trials

The random variable discussed in our example could take on values from a set of discrete numbers and hence, the image of the random variable is either finite or countable. Such random variables, known as **discrete random variables**, are the subject of this chapter, while continuous random variables are discussed in the next chapter. A random variable defined on a discrete sample space will be discrete, while it is possible to define a discrete random variable on a continuous sample space. For instance, for a continuous sample space S , the random variable defined by, say, $X(s) = 4$ for all $s \in S$ is discrete.

Problems

- Given a discrete random variable X , define the event A_x by

$$A_x = \{s \in S \mid X(s) = x\}.$$

Show that the family of events $\{A_x\}$ defines an event space.

2.3 THE PROBABILITY MASS FUNCTION

We have defined the event A_x as the set of all sample points $\{s \mid X(s) = x\}$. Consequently

$$\begin{aligned} P(A_x) &= P([X = x]) \\ &= P(\{s \mid X(s) = x\}) \\ &= \sum_{X(s)=x} P(s). \end{aligned}$$

This formula provides us with a method of computing $P(X = x)$ for all $x \in \mathfrak{R}$. Thus we have defined a function with its domain consisting of the event space of the random variable X , and with its range in the closed interval $[0,1]$. This function is known as the **probability mass function** (pmf) or the **discrete density function** of the random variable X , and will be denoted by $p_x(x)$. Thus

$$\begin{aligned} p_x(x) &= P(X = x) \\ &= \sum_{X(s)=x} P(s) \\ &= \text{probability that the value of the random variable } \\ &\quad X \text{ obtained on a performance of the experiment is} \\ &\quad \text{equal to } x. \end{aligned}$$

It should be noted that the argument x of the pmf $p_x(x)$ is a dummy variable, hence it can be changed to any other dummy variable y with no effect on the definition.

The following properties hold for the pmf:

(p1) $0 \leq p_x(x) \leq 1$ for all $x \in \mathfrak{R}$. This must be true, since $p_x(x)$ is a probability.

(p2) Since the random variable assigns some value $x \in \mathfrak{R}$ to each sample point $s \in S$, we must have

$$\sum_{x \in \mathfrak{R}} p_x(x) = 1.$$

(p3) For a discrete random variable X , the set $\{x \mid p_x(x) \neq 0\}$ is a finite or countably infinite subset of real numbers (this set is defined to be the image of X). Let this set be denoted by $\{x_1, x_2, \dots\}$. Then property (p2) may be restated as

$$\sum_i p_x(x_i) = 1.$$

A real-valued function $p_x(x)$ defined on \mathfrak{R} is the pmf of some random variable X provided that it satisfies properties (p1) to (p3). Continuing with Example 2.1, we can easily obtain $p_x(x)$ for $x = 0, 1, 2, 3$ from the preceding definitions:

$$\begin{aligned} p_x(0) &= \frac{1}{8}, \\ p_x(1) &= \frac{3}{8}, \\ p_x(2) &= \frac{3}{8}, \\ p_x(3) &= \frac{1}{8}. \end{aligned}$$

Check that all the properties listed above hold. This pmf may be visualized as a bar histogram drawn over the event space for the random variable (see Figure 2.3).

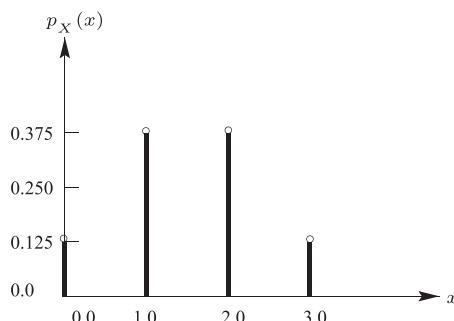


Figure 2.3. Histogram of pmf for Example 2.1

Example 2.2

Returning to the example of a wireless cell with five channels from Chapter 1, and defining the random variable X = the number of available channels, we have

$$\begin{aligned} p_X(0) &= \frac{1}{32}, \quad p_X(1) = \frac{5}{32}, \quad p_X(2) = \frac{10}{32}, \\ p_X(3) &= \frac{10}{32}, \quad p_X(4) = \frac{5}{32}, \quad p_X(5) = \frac{1}{32}. \end{aligned}$$

‡

2.4 DISTRIBUTION FUNCTIONS

So far we have restricted our attention to computing $P(X = x)$, but often we may be interested in computing the probability of the set $\{s \mid X(s) \in A\}$ for some subset A of \mathbb{R} other than a one-point set. It is clear that

$$\{s \mid X(s) \in A\} = \bigcup_{x_i \in A} \{s \mid X(s) = x_i\}. \quad (2.1)$$

Usually this event is denoted as $[X \in A]$ and its probability by $P(X \in A)$. If $-\infty < a < b < \infty$ and A is an interval with endpoints a and b , say, $A = (a, b)$, then we usually write $P(a < X < b)$ instead of $P[X \in (a, b)]$. Similarly, if $A = [a, b]$, then $P(X \in A)$ will be written as $P(a < X \leq b)$. The semiinfinite interval $A = (-\infty, x]$ will be of special interest and in this case we denote the event $[X \in A]$ by $[X \leq x]$.

If $p_X(x)$ denotes the pmf of random variable X , then, from equation (2.1), we have

$$P(X \in A) = \sum_{x_i \in A} p_X(x_i).$$

Thus in Example 2.2, the probability that two or fewer channels will be available may now be evaluated quite simply as

$$\begin{aligned} P(X \leq 2) &= P(X = 0) + P(X = 1) + P(X = 2) \\ &= p_X(0) + p_X(1) + p_X(2) \\ &= \frac{1}{32} + \frac{5}{32} + \frac{10}{32} \\ &= \frac{16}{32} \\ &= \frac{1}{2}. \end{aligned}$$

The function $F_X(t)$, $-\infty < t < \infty$, defined by

$$\begin{aligned} F_X(t) &= P(-\infty < X \leq t) \\ &= P(X \leq t) \\ &= \sum_{x \leq t} p_X(x) \end{aligned} \quad (2.2)$$

is called the **cumulative distribution function** (CDF) or the **probability distribution function** or simply the **distribution function** of the random variable X . We will omit the subscript X whenever no confusion arises. It follows from this definition that

$$\begin{aligned} P(a < X \leq b) &= P(X \leq b) - P(X \leq a) \\ &= F(b) - F(a). \end{aligned}$$

If X is an integer-valued random variable, then

$$F(t) = \sum_{-\infty < x \leq \lfloor t \rfloor} p_X(x)$$

where $\lfloor t \rfloor$ denotes the greatest integer less than or equal to t (also known as the **floor** of t).

Several properties of $F_X(x)$ follow directly from its definition.

(F1) $0 \leq F(x) \leq 1$ for $-\infty < x < \infty$. This follows because $F(x)$ is a probability.

(F2) $F(x)$ is a monotone increasing function of x ; that is, if $x_1 \leq x_2$, then $F(x_1) \leq F(x_2)$. This follows by first observing that the interval $(-\infty, x_1]$ is contained in the interval $(-\infty, x_2]$ whenever $x_1 \leq x_2$ and hence

$$P(-\infty < X \leq x_1) \leq P(-\infty < X \leq x_2).$$

That is, $F(x_1) \leq F(x_2)$.

(F3) $\lim_{x \rightarrow -\infty} F(x) = 0$, and $\lim_{x \rightarrow \infty} F(x) = 1$. If the random variable X has a finite image, then $F(x) = 0$ for all x sufficiently small and $F(x) = 1$ for all x sufficiently large.

(F4) $F(x)$ has a positive jump equal to $p_X(x_i)$ at $i = 1, 2, \dots$, and in the interval $[x_i, x_{i+1})$ $F(x)$ has a constant value. Thus

$$F(x) = F(x_i) \quad \text{for } x_i \leq x < x_{i+1}$$

and

$$F(x_{i+1}) = F(x_i) + p_X(x_{i+1}).$$

It can be shown that any function $F(x)$ satisfying properties (F1)–(F4) is the distribution function of some discrete random variable.

We note that distribution functions of discrete random variables grow only by jumps, whereas the distribution functions of continuous random variables are continuous functions and hence have no jumps. A random variable X is said to be of **mixed type** if its distribution function has both jumps as well as continuous growth. In most practical situations, the random variable is

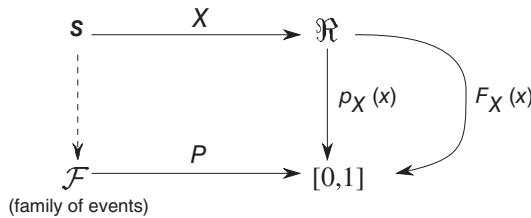


Figure 2.4. Domain and range of P , X , pmf, and CDF

either discrete or continuous. Therefore, we will study only these two cases in detail. The domains and ranges of the four functions (the probability measure, the random variable X , the pmf, and the CDF) we have studied so far are summarized in Figure 2.4.

The cumulative distribution function contains most of the interesting information about the underlying probability system and will be used extensively. Often the concepts of sample space, event space, and probability measure, which are fundamental in building the theory of probability, will fade into the background, and functions such as the distribution function or the probability mass function become the most important entities. It is important, nevertheless, to keep this background in mind. You will often see the statement “Let X be a discrete random variable with pmf p_x ,” with no reference made to the underlying probability space. We can always construct an appropriate space, as follows. Take $S = \mathfrak{R}$; $X(s) = s$, for $s \in S$; $\mathcal{F} = \text{union of the inverse images of } A_x \text{ of all the subsets } x \text{ pertaining to the set of real numbers } \mathfrak{R} \text{ and}$

$$P(A) = \sum_{x \in A} p_x(x)$$

for a subset, A , of \mathfrak{R} . In this case, the event space of the random variable X is identical to the sample space defined above. Similarly, the statement, “Let X be a discrete random variable with the CDF F ,” always makes sense.

Example 2.3

The CDF of the running example of the sequence of three Bernoulli trials is shown in Figure 2.5. The properties (F1)–(F4) above are easily seen to hold.

#

2.5 SPECIAL DISCRETE DISTRIBUTIONS

In many theoretical and practical problems, several probability mass functions appear frequently enough that they are worth exploring here.

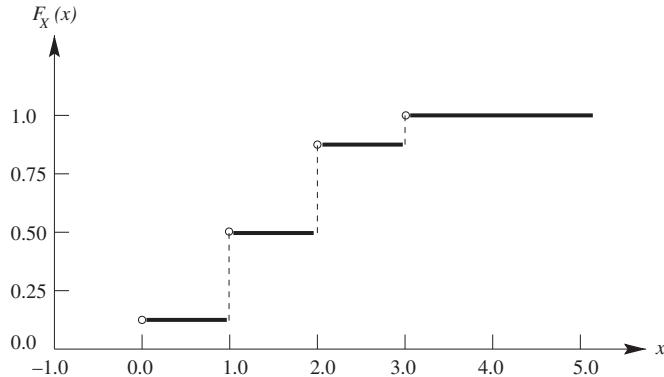


Figure 2.5. CDF of three Bernoulli trials

2.5.1 The Bernoulli pmf

The Bernoulli pmf is the density function of a discrete random variable X having 0 and 1 as its only possible values; it originates from the experiment consisting of a single Bernoulli trial. It is given by

$$\begin{aligned} p_x(0) &= p_0 = P(X = 0) = q, \\ p_x(1) &= p_1 = P(X = 1) = p, \end{aligned}$$

where $p + q = 1$. The corresponding CDF is given by (see Figure 2.6)

$$F(x) = \begin{cases} 0 & \text{for } x < 0, \\ q & \text{for } 0 \leq x < 1, \\ 1 & \text{for } x \geq 1. \end{cases}$$

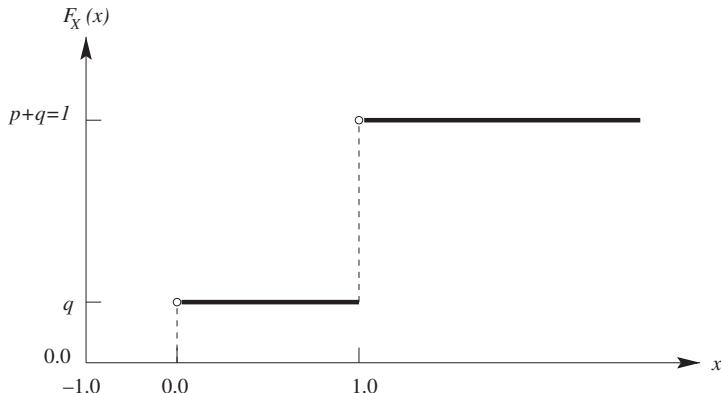


Figure 2.6. CDF of Bernoulli random variable

2.5.2 The Binomial pmf

To generate the Bernoulli pmf, we considered a single Bernoulli trial. Now we consider a sequence of n independent Bernoulli trials with the probability of success equal to p on each trial. Let Y_n denote the number of successes in n trials. The domain of the random variable Y_n is all the n -tuples of 0s and 1s, and the image is $\{0, 1, \dots, n\}$. The value assigned to a sample point (an n -tuple) by Y_n simply corresponds to the number of 1s in the n -tuple. As was shown in Section 1.12, the pmf of Y_n is

$$\begin{aligned} p_k &= P(Y_n = k) \\ &= p_{Y_n}(k) \\ &= \begin{cases} \binom{n}{k} p^k (1-p)^{n-k} & \text{for } 0 \leq k \leq n, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (2.3)$$

This equation gives the probability of k successes in n independent trials of an experiment that has probability p of success on each trial. One of the more important discrete densities in probability theory, this is called the **binomial density** with parameters n and p , often denoted by $b(k; n, p)$. An example of $b(k; 3, 0.5)$ was presented earlier in this chapter (see Figure 2.3).

It is easily verified using the binomial theorem that

$$\begin{aligned} \sum_{i=0}^n p_i &= \sum_{i=0}^n \binom{n}{i} p^i (1-p)^{n-i} \\ &= [p + (1-p)]^n \\ &= 1. \end{aligned}$$

This is the reason for the term *binomial* pmf. We often refer to a random variable Y_n having a binomial pmf by saying that Y_n has a **binomial distribution** (with parameters n and p if we want to be more precise). Similar phraseology will be used for other random variables having a named density. The distribution function of a binomial random variable will be denoted by $B(t; n, p)$ and is given by

$$\begin{aligned} B(t; n, p) &= F_{Y_n}(t) \\ &= \sum_{i=0}^{\lfloor t \rfloor} \binom{n}{i} p^i (1-p)^{n-i}. \end{aligned} \quad (2.4)$$

The binomial distribution is applicable whenever a series of trials is made satisfying the following conditions:

1. Each trial has exactly two mutually exclusive outcomes, usually labeled “success” and “failure.”
2. The probability of “success” on each trial is a constant, denoted by p . The probability of “failure” is $q = 1 - p$.
3. The outcomes of successive trials are mutually independent.

A typical situation in which these conditions will apply (at least approximately) occurs when several components are selected at random (with replacement) from a large batch of components and examined to see if there are any defective components (i.e., failures). The number of defectives in a sample of size n is a random variable, denoted by Y_n , which is binomially distributed.

These assumptions constitute what is called a *binomial model*, which is a typical example of a mathematical model in that it attempts to describe a physical situation in mathematical terms. Models such as these depend on one or more **parameters** that govern their behavior. The binomial model has two parameters, n and p . If the values of model parameters are known, then it is relatively easy to evaluate the probabilities of the events of interest.

We emphasize that the three properties listed above are **assumptions** and need not always hold. We may wish to analyze empirically observed data, and may hypothesize that the assumptions of the binomial model (or any other such model) hold. This hypothesis needs to be tested and can be either rejected or accepted on the basis of the test. Hypothesis testing is discussed in Chapter 10.

Example 2.4

As an example of binomial distribution, consider a plant manufacturing VLSI (very large-scale integrated circuit) chips, 10% of which are expected to be defective. The quality control procedure consists of counting the number of defective chips in a sample of size 35. Suppose after 800 applications of this procedure we find that our experience is reflected in the following table. Although we do not expect exactly 10% defectives every time, are the observations consistent with our hypothesis that 10% are defective?

<i>Number of defects</i>	<i>Number of samples showing this number of defects</i>	<i>Fraction (of 800 samples) showing this number of defects</i>
0	11	0.01375
1	95	0.11875
2	139	0.17375
3	213	0.26625
4	143	0.17875

(continued overleaf)

<i>Number of defects</i>	<i>Number of samples showing this number of defects</i>	<i>Fraction (of 800 samples) showing this number of defects</i>
5	113	0.14125
6	49	0.06125
7	27	0.03375
8	6	0.00750
9	4	0.00500
10	<u>0</u>	<u>0.00000</u>
	800	1.00000

This situation is typical of those fitting a binomial model. “Success” is finding a defective chip, and we are counting the number of successes in 35 trials. Since the probability of success is $p = 0.1$, the observed fraction defective should be close to the binomial pmf:

$$b(k; 35, 0.1) = \binom{35}{k} \cdot 0.1^k \cdot 0.9^{(35-k)}.$$

The observed data and the binomial pmf are compared in the following table as well as in Figure 2.7. In Chapter 10 we will study statistical tests that will allow us to quantify the goodness of fit of the data presented above to the binomial model.

<i>k = defects/sample</i>	<i>Data</i>	<i>b(k; 35, 0.1)</i>
0	0.01375	0.0250
1	0.11875	0.0974
2	0.17375	0.1839
3	0.26625	0.2248
4	0.17875	0.1998
5	0.14125	0.1376
6	0.06125	0.0765
7	0.03375	0.0352
8	0.00750	0.0137
9	0.00500	0.0046
10	0.00000	0.0013
11	0.00000	0.0003
12	0.00000	0.0000

‡

Example 2.5

The number of surviving components, Y_n , out of a given number of n identical and independent components has a binomial distribution $B(k; n, R)$, where R is

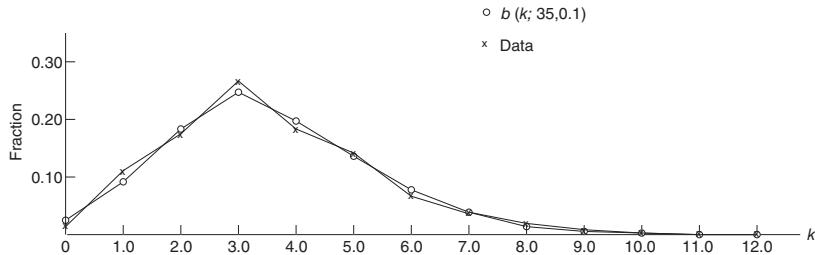


Figure 2.7. Comparing the model pmf with data of Example 2.4

the reliability of a single component. Thus the reliability of an k -out-of- n system is given by

$$\begin{aligned}
 R_{k|n} &= P(\text{"}k \text{ or more components have not failed"}) \\
 &= 1 - \sum_{i=0}^{k-1} p_{Y_n}(i) \\
 &= 1 - F_{Y_n}(k-1) \\
 &= \sum_{i=k}^n \binom{n}{i} R^i (1-R)^{(n-i)}. \tag{2.5}
 \end{aligned}$$

‡

Example 2.6

While transmitting binary digits through a communication channel, the number of digits received correctly, C_n , out of n transmitted digits has a binomial distribution $B(k; n, p)$, where p is the probability of successfully transmitting one digit. The probability of exactly i errors is given by

$$P_e(i) = p_{C_n}(n-i) = \binom{n}{i} p^{(n-i)} (1-p)^i,$$

and thus the probability of an error-free transmission is given by:

$$P_e(0) = p^n.$$

‡

Example 2.7

Now consider the logical link control (LLC) and medium access control (MAC) protocol of a wireless communication system [DEME 1999]. When an LLC frame is passed to the MAC layer, it is segmented into n MAC blocks of fixed size and these n blocks are transmitted through the radio channel separately. Assume that the automatic repeat request (ARQ) scheme is applied in case an error occurred during the transmission. Let $P_c(k)$ denote the probability that after the $(k-1)$ st

MAC retransmission there are MAC blocks in error that are corrected by the k th MAC retransmission. We are interested in the pmf of K , which is the number of LLC transmissions required for the error free transmission of n MAC blocks.

Assume that the probability of successful transmission of a single block is $p(>0)$. Then the probability $P_c(1)$ that all n MAC blocks of an LLC frame are received error-free at the first transmission is equal to

$$P_c(1) = p^n,$$

where we assume that the transmission of MAC blocks are statistically independent events. To calculate $P_c(2)$, we note that $1 - (1 - p)^2$ is the probability that a given MAC block is successfully received after two MAC retransmissions. Then, $(1 - (1 - p)^2)^n$ is the probability that the LLC frame is correctly received within one or two transmissions. This yields

$$P_c(2) = [1 - (1 - p)^2]^n - p^n.$$

Following the above approach, the general form of $P_c(k)$ is

$$P_c(k) = \left[1 - (1 - p)^k\right]^n - \left[1 - (1 - p)^{k-1}\right]^n.$$

From above equation we have

$$\lim_{k \rightarrow \infty} P_c(k) = \lim_{k \rightarrow \infty} \left\{ \left[1 - (1 - p)^k\right]^n - \left[1 - (1 - p)^{k-1}\right]^n \right\} = 0 \quad (2.6)$$

and

$$\sum_{k=1}^{\infty} \left\{ \left[1 - (1 - p)^k\right]^n - \left[1 - (1 - p)^{k-1}\right]^n \right\} = 1.$$

#

Example 2.8

Consider taking a random sample of 10 VLSI chips from a very large batch. If no chips in the sample are found to be defective, then we accept the entire batch; otherwise we reject the batch. The number of defective chips in a sample has the pmf $b(k; 10, p)$, where p denotes the probability that a randomly chosen chip is defective. Thus

$$\begin{aligned} P(\text{"No defectives"}) &= (1 - p)^{10} \\ &= \text{probability that a batch is accepted.} \end{aligned}$$

If $p = 0$, the batch is certain to be accepted and if $p = 1$, the batch will certainly be rejected. The expression for the probability of acceptance is plotted in Figure 2.8.

#

The student should not be misled by these examples into thinking that quality control problems can always be solved by simply plugging numbers into the binomial pmf.

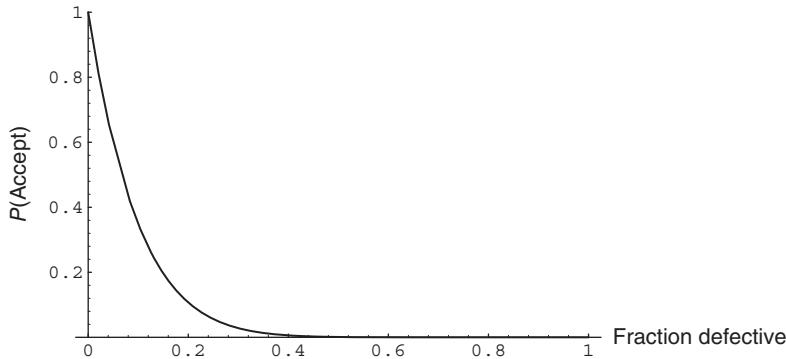


Figure 2.8. Probability of acceptance versus fraction defective

Example 2.9 (Simpson's Reversal Paradox)

Consider two shipments (labeled I and II) of VLSI chips from each of the two manufacturers A and B. Suppose that the proportion of defectives among the four shipments are as follows:

		<i>Manufacturer</i>	
		A	B
<i>Shipment</i>	I	600 good 500 defective	400 good 300 defective
	II	300 good 600 defective	500 good 900 defective

On inspecting shipment I, the quality control engineer will find

$$\begin{aligned} P(\text{selecting a defective chip from A}) &= \frac{5}{11} \\ &> P(\text{selecting a defective chip from B}) = \frac{3}{7}. \end{aligned}$$

Inspection of shipment II yields

$$\begin{aligned} P(\text{selecting a defective chip from A}) &= \frac{6}{9} \\ &> P(\text{selecting a defective chip from B}) = \frac{9}{14}. \end{aligned}$$

The engineer will presumably conclude that the manufacturer B is sending better chips than the manufacturer A. Suppose, however, that the engineer mixes the two shipments from A together and similarly for B. A subsequent test leads him to a reverse conclusion since

$$\begin{aligned} P(\text{selecting a defective chip from A}) &= \frac{11}{20} \\ &< P(\text{selecting a defective chip from B}) = \frac{12}{21}. \end{aligned}$$

The problem here is that we are tempted to add the fractions $\frac{5}{11} + \frac{6}{9}$ and compare the sum with $\frac{3}{7} + \frac{9}{14}$; unfortunately, what is called for is adding numerators and adding denominators, which is *not* the way we add fractions.

‡

When n becomes very large, computation using the binomial formula becomes unmanageable. In the limit as n approaches infinity, it can be shown that

$$b(k; n, p) \simeq \frac{1}{\sqrt{2\pi npq}} \cdot e^{-(k-np)^2/(2npq)}. \quad (2.7)$$

This is known as the *Laplace* (or *normal*) *approximation* to the binomial pmf and the agreement between the two formulas depends on the values of n and p . Take $n = 5$ and $p = 0.5$, then

k	<i>Laplace approximation</i> to $b(k; 5, 0.5)$	
	$b(k; 5, 0.5)$	
0	0.03125	0.02929
1	0.15625	0.14507
2	0.31250	0.32287
3	0.31250	0.32287
4	0.15625	0.14507
5	0.03125	0.02929

As p moves away from 0.5, larger values of n are needed. Larson [LARS 1974] suggests that for $n \geq 10$, if

$$\frac{9}{n+9} \leq p \leq \frac{n}{n+9},$$

then the Laplace formula provides a good approximation to the binomial pmf.

Other authors give different advice concerning when to use the normal approximation. For example, Schader and Schmid [SCHA 1989] compared the maximum absolute error in computing the cumulative binomial distribution function using the normal approximation with a continuity correction (see Example 3.7). They consider the two rules for determining whether this approximation should be used: np and $n(1-p)$ are both greater than 5, and $np(1-p) > 9$. Their conclusion is that the relationship between the maximum absolute error and p is approximately linear when considering the smallest possible sample sizes to satisfy the rules. For more information, refer to Leemis and Trivedi [LEEM 1996], Chapter 10 of this book and the Poisson pmf section in this chapter. Yet another approximation to the binomial pmf is the Poisson pmf, which we will study later.

Owing to the importance of the binomial distribution, the binomial CDF

$$B(k; n, p) = \sum_{i=0}^k b(i; n, p)$$

has been tabulated for $n = 2$ to $n = 49$ by the National Bureau of Standards [NBS 1950] and for $n = 50$ to $n = 100$ by Romig [ROMI 1953]. [In Appendix C, we have tabulated $B(k; n, p)$ for $n = 2$ to 20]. For larger values of n , we recommend the use of the Mathematica [WOLF 1999] built-in function **Binomial[n,k]**. Three different possible shapes of binomial pmf's are illustrated in Figures 2.9–2.11. If $p = 0.5$, the bar chart of the binomial pmf is **symmetric**, as in Figure 2.9. If $p < 0.5$, then a **positively skewed** binomial pmf is obtained (see Figure 2.10), and if $p > 0.5$ a **negatively skewed** binomial pmf is obtained (see Figure 2.11). Here, a bar chart is said to be positively skewed if the long “tail” is on the right, and it is said to be negatively skewed if the long “tail” is on the left.

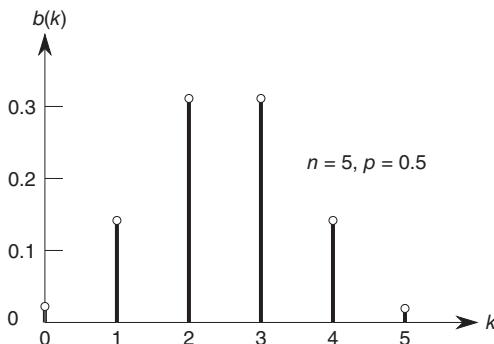


Figure 2.9. Symmetric binomial pmf

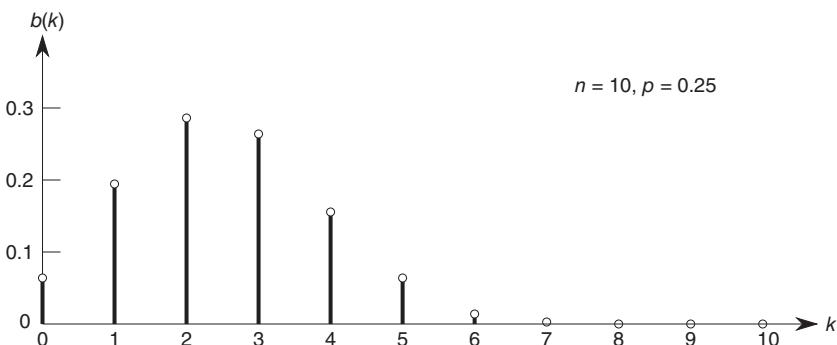


Figure 2.10. Positively skewed binomial pmf

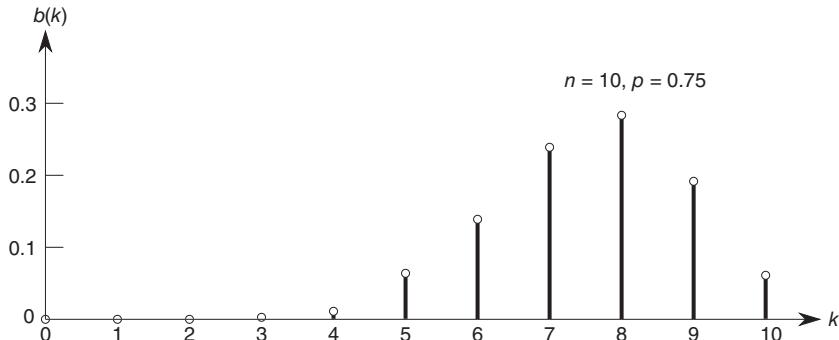


Figure 2.11. Negatively skewed binomial pmf

2.5.3 The Geometric pmf

Once again we consider a sequence of Bernoulli trials, but instead of counting the number of successes in a fixed number n of trials, we count the number of trials until the first “success” occurs. If we let 0 denote a failure and let 1 denote a success then the sample space of this experiment consists of the set of all binary strings with an arbitrary number of 0s followed by a single 1:

$$S = \{0^{i-1}1 | i = 1, 2, 3, \dots\}.$$

Note that this sample space has a countably infinite number of sample points. Define a random variable Z on this sample space so that the value assigned to the sample point $0^{i-1}1$ is i . Thus Z is the number of trials up to and including the first success. Therefore, Z is a random variable with image $\{1, 2, \dots\}$, which is a countably infinite set. To find the pmf of Z , we note that the event $[Z = i]$ occurs if and only if we have a sequence of $i - 1$ failures followed by one success. This is a sequence of independent Bernoulli trials with the probability of success equal to p . Hence, we have

$$\begin{aligned} p_Z(i) &= q^{i-1}p \\ &= p(1-p)^{i-1} \quad \text{for } i = 1, 2, \dots, \end{aligned} \tag{2.8}$$

where $q = 1 - p$. By the formula for the sum of a geometric series, we have

$$\begin{aligned} \sum_{i=1}^{\infty} p_Z(i) &= \sum_{i=1}^{\infty} pq^{i-1} \\ &= \frac{p}{1-q} \\ &= \frac{p}{p} \\ &= 1. \end{aligned}$$

Any random variable Z with the image $\{1, 2, \dots\}$ and pmf given by a formula of the form of equation (2.8) is said to have a **geometric distribution**, and the function given by (2.8) is termed a **geometric pmf** with parameter p . The distribution function of Z is given by

$$F_Z(t) = \sum_{i=1}^{\lfloor t \rfloor} p(1-p)^{i-1} \\ = 1 - (1-p)^{\lfloor t \rfloor} \quad \text{for } t \geq 0. \quad (2.9)$$

Graphs of the geometric pmf for two different values of parameter p are sketched in Figure 2.12.

The random variable Z counts the total number of trials up to and including the first success. We are often interested in counting the number of failures before the first success. Let this number be called the random variable X with the image $\{0, 1, 2, \dots\}$. Clearly, $Z = X + 1$. The random variable X is said to

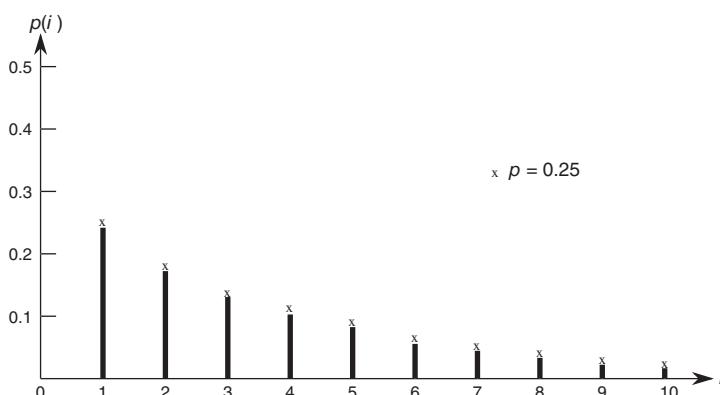
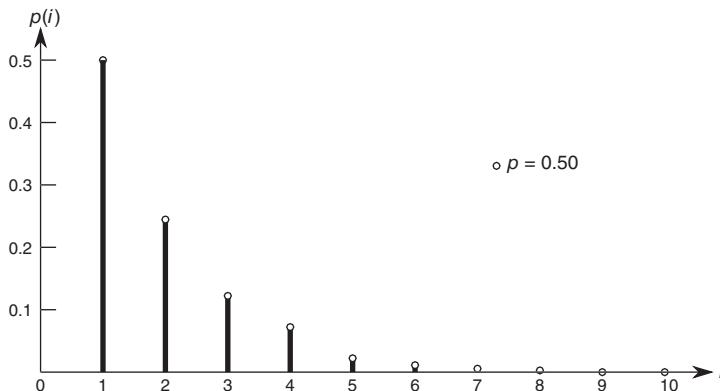


Figure 2.12. Geometric pmf

have a **modified geometric pmf**, specified by

$$p_x(i) = p(1-p)^i \quad \text{for } i = 0, 1, 2, \dots \quad (2.10)$$

The distribution function of X is given by

$$\begin{aligned} F_X(t) &= \sum_{i=0}^{\lfloor t \rfloor} p(1-p)^i \\ &= 1 - (1-p)^{\lfloor t+1 \rfloor} \quad \text{for } t \geq 0. \end{aligned} \quad (2.11)$$

The geometric (and modified geometric) distribution is encountered in some problems in queuing theory. Following are several examples where this distribution occurs:

1. A series of components is made by a certain manufacturer. The probability that any given component is defective is a constant p , which does not depend on the quality of the previous components. The probability that the i th item is the first defective one is given by formula (2.8).
2. Consider the scheduling of a computer system with a fixed time slice (see Figure 18). At the end of a time slice, the program would have completed execution with a probability p ; thus there is a probability $q = 1 - p > 0$ that it needs to perform more computation. The pmf of the random variable denoting the number of time slices needed to complete the execution of a program is given by formula (2.8), if we assume that the operation of the computer satisfies the usual independence assumptions.
3. Consider the following program segment consisting of a **while** loop:

while $\neg B$ **do** S

Assume that the Boolean expression B takes the value **true** with probability p and the value **false** with probability q . If the successive tests on B are independent, then the number of times the body (or the statement group S) of the loop is executed will be a random variable having a modified geometric distribution with parameter p .

4. With the assumptions as in Example 3 above, consider a **repeat** loop:

repeat S **until** B

The number of times the body of the **repeat** loop is executed will be a geometrically distributed random variable with parameter p .

The geometric distribution has an important property, known as the **memoryless property**. Furthermore, it is the only discrete distribution with this

property. To illustrate this property, consider a sequence of Bernoulli trials and let Z represent the number of trials until the first success. Now assume that we have observed a fixed number n of these trials and found them all to be failures. Let Y denote the number of additional trials that must be performed until the first success. Then $Y = Z - n$, and the conditional probability is

$$\begin{aligned} q_i &= P(Y = i | Z > n) \\ &= P(Z - n = i | Z > n) \\ &= P(Z = n + i | Z > n) \\ &= \frac{P(Z = n + i \text{ and } Z > n)}{P(Z > n)} \end{aligned}$$

by using the definition of conditional probability. But for $i = 1, 2, 3, \dots$, $Z = n + i$ implies that $Z > n$. Thus the event $[Z = n + i \text{ and } Z > n]$ is the same as the event $[Z = n + i]$. Therefore

$$\begin{aligned} q_i &= P(Y = i | Z > n) \\ &= \frac{P(Z = n + i)}{P(Z > n)} \\ &= \frac{p_Z(n+i)}{1 - F_Z(n)} \\ &= \frac{pq^{n+i-1}}{1 - (1 - q^n)} \\ &= \frac{pq^{n+i-1}}{q^n} \\ &= pq^{i-1} \\ &= p_Z(i). \end{aligned}$$

Thus we see that, conditioned on $Z > n$, the number of trials remaining until the first success, $Y = Z - n$, has the same pmf as Z had originally. If a run of failures is observed in a sequence of Bernoulli trials, we need not “remember” how long the run was to determine the probabilities for the number of additional trials needed until the first success. The proof that any discrete random variable Z with image $\{1, 2, 3, \dots\}$ and having the memoryless property must have the geometric distribution is left as an exercise.

2.5.4 The Negative Binomial pmf

To obtain the geometric pmf, we observed the number of trials until the first success in a sequence of Bernoulli trials. Now let us observe the number of trials until the r^{th} success, and let T_r be the random variable denoting this

number. It is clear that the image of T_r is $\{r, r+1, r+2, \dots\}$. To compute $p_{T_r}(n)$, define the events:

- $A = "T_r = n."$
- $B = \text{"Exactly } r-1 \text{ successes occur in } n-1 \text{ trials."}$
- $C = \text{"The } n\text{th trial results in a success."}$

Then clearly

$$A = B \cap C$$

and the events B and C are independent. Therefore

$$P(A) = P(B)P(C).$$

To compute $P(B)$, consider a particular sequence of $n-1$ trials with $r-1$ successes and $n-1-(r-1)=n-r$ failures. The probability associated with such a sequence is $p^{r-1}q^{n-r}$ and there are $\binom{n-1}{r-1}$ such sequences. Therefore

$$P(B) = \binom{n-1}{r-1} p^{r-1} q^{n-r}.$$

Now since $P(C) = p$,

$$\begin{aligned} p_{T_r}(n) &= P(T_r = n) \\ &= P(A) \\ &= \binom{n-1}{r-1} p^r q^{n-r} \\ &= \binom{n-1}{r-1} p^r (1-p)^{n-r}, \quad n = r, r+1, r+2, \dots \end{aligned}$$

Using some combinatorial identities [KNUT 1997; p. 57], an alternative form of this pmf can be established:

$$p_{T_r}(n) = p^r \binom{-r}{n-r} (-1)^{n-r} (1-p)^{n-r}, \quad n = r, r+1, r+2, \dots \quad (2.12)$$

This pmf is known as the **negative binomial pmf**, and although we derived it assuming an integral value of r , any positive real value of r is allowed (of course the interpretation of r as a number of successes is no longer applicable). Quite clearly, if we let $r = 1$ in the formula (2.12), then we get the geometric pmf.

To verify that $\sum_{n=r}^{\infty} p_{T_r}(n) = 1$, we recall that the Taylor series expansion of $(1-t)^{-r}$ for $-1 < t < 1$ is

$$(1-t)^{-r} = \sum_{n=r}^{\infty} \binom{-r}{n-r} (-t)^{n-r}.$$

Substituting $t = 1 - p$, we have

$$p^{-r} = \sum_{n=r}^{\infty} \binom{-r}{n-r} (-1)^{n-r} (1-p)^{n-r},$$

which gives us the required result.

As in the case of the geometric distribution, there is a modified version of the negative binomial distribution. Let the random variable Z denote the number of failures before the occurrence of the r^{th} success. Then Z is said to have the **modified negative binomial** distribution with the pmf:

$$p_z(n) = \binom{n+r-1}{r-1} p^r (1-p)^n, \quad n \geq 0. \quad (2.13)$$

The pmf in equation (2.13) reduces the modified geometric pmf when $r = 1$.

2.5.5 The Poisson pmf

Let us consider another problem related to the binomial distribution. Suppose that we are observing the arrival of jobs to a large database server for the time interval $(0, t]$. It is reasonable to assume that for a small interval of duration Δt the probability of a new job arrival is $\lambda \cdot \Delta t$, where λ is a constant that depends upon the user population of the database server. If Δt is sufficiently small, then the probability of two or more jobs arriving in the interval of duration Δt may be neglected. We are interested in calculating the probability of k jobs arriving in the interval of duration t .

Suppose that the interval $(0, t]$ is divided into n subintervals of length t/n , and suppose further that the arrival of a job in any given interval is independent of the arrival of a job in any other interval. Then for a sufficiently large n , we can think of the n intervals as constituting a sequence of Bernoulli trials with the probability of success $p = \lambda t/n$. It follows that the probability of k arrivals in a total of n intervals each with a duration t/n is approximately given by

$$b\left(k; n, \frac{\lambda t}{n}\right) = \binom{n}{k} \left(\frac{\lambda t}{n}\right)^k \left(1 - \frac{\lambda t}{n}\right)^{n-k}, \quad k = 0, 1, \dots, n.$$

Since the assumption that the probability of more than one arrival per interval can be neglected is reasonable if and only if t/n is very small, we will take the limit of the above pmf as n approaches ∞ . Now

$$\begin{aligned} b\left(k; n, \frac{\lambda t}{n}\right) &= \frac{n(n-1)(n-2)\dots(n-k+1)}{k!n^k} (\lambda t)^k \cdot \left(1 - \frac{\lambda t}{n}\right)^{(n-k)} \\ &= \frac{n}{n} \cdot \frac{n-1}{n} \cdot \dots \cdot \frac{n-k+1}{n} \cdot \frac{(\lambda t)^k}{k!} \cdot \left(1 - \frac{\lambda t}{n}\right)^{-k} \cdot \left(1 - \frac{\lambda t}{n}\right)^n. \end{aligned}$$

We are interested in what happens to this expression as n increases, because then the subinterval width approaches zero, and the approximation involved gets better and better. In the limit as n approaches infinity, the first k factors approach unity, the next factor is fixed, the next approaches unity, and the last factor becomes

$$\lim_{n \rightarrow \infty} \left\{ \left[1 - \frac{\lambda t}{n} \right]^{-n/(\lambda t)} \right\}^{-\lambda t}.$$

Setting $-\lambda t/n = h$, this factor is

$$\left[\lim_{h \rightarrow 0} (1 + h)^{1/h} \right]^{-\lambda t} = e^{-\lambda t},$$

since the limit in the brackets is the common definition of e . Thus, the binomial pmf approaches

$$\frac{e^{-\lambda t} (\lambda t)^k}{k!}, \quad k = 0, 1, 2, \dots$$

Now replacing λt by a single parameter α , we get the well-known Poisson pmf:

$$f(k; \alpha) = e^{-\alpha} \frac{\alpha^k}{k!}, \quad k = 0, 1, 2, \dots \quad (2.14)$$

Thus the Poisson pmf can be used as a convenient approximation to the binomial pmf when n is large and p is small:

$$\binom{n}{k} p^k q^{n-k} \simeq e^{-\alpha} \frac{\alpha^k}{k!}, \quad \text{where } \alpha = np.$$

An acceptable rule of thumb is to use the Poisson approximation for binomial probabilities if $n \geq 20$ and $p \leq 0.05$. The table that follows compares $b(k; 5, 0.2)$ and $b(k; 20, 0.05)$ with $f(k; 1)$. Observe that the approximation is better in the case of larger n and smaller p .

k	$b(k; 5, 0.2)$	$b(k; 20, 0.05)$	$f(k; 1)$
0	0.328	0.359	0.368
1	0.410	0.377	0.368
2	0.205	0.189	0.184
3	0.051	0.060	0.061

There are other recommendations from different authors concerning which values of n and p are appropriate. Normal approximation was introduced earlier in this chapter as one way to approximate the binomial pmf. It is useful when n is large and $p \approx 1/2$. The Poisson approximation, although less popular, is good for large values of n and small values of p .

Besides errors in computing binomial probabilities, if the approximation is used in parameter estimation as in Chapter 10, the effect of approximation on the confidence interval should also be considered when using these approximations [LEEM 1996].

Example 2.10

A manufacturer produces VLSI chips, 1% of which are defective. Find the probability that in a box containing 100 chips, no defectives are found.

Since $n = 100$ and $p = 0.01$, the required answer is

$$\begin{aligned} b(0; 100, 0.01) &= \binom{100}{0} \cdot 0.01^0 \cdot 0.99^{100} \\ &= 0.99^{100} \\ &= 0.366. \end{aligned}$$

Using the Poisson approximation, $\alpha = 100 \cdot 0.01 = 1$, and the required answer is

$$\begin{aligned} f(0; 1) &= e^{-1} \\ &= 0.3679. \end{aligned}$$

‡

It is easily verified that the probabilities from equation (2.14) are nonnegative and sum to 1:

$$\begin{aligned} \sum_{k=0}^{\infty} f(k; \alpha) &= \sum_{k=0}^{\infty} \frac{\alpha^k}{k!} e^{-\alpha} \\ &= e^{-\alpha} \sum_{k=0}^{\infty} \frac{\alpha^k}{k!} \\ &= e^{-\alpha} \cdot e^{\alpha} \\ &= 1. \end{aligned}$$

The probabilities $f(k; \alpha)$ are easy to calculate, starting with

$$f(0; \alpha) = e^{-\alpha}$$

and using the recurrence relation

$$f(k+1; \alpha) = \frac{\alpha f(k; \alpha)}{k+1}. \quad (2.15)$$

For very large values of α , special care is necessary to avoid numerical problems in computing Poisson pmf. Fox and Glynn have published an algorithm that is recommended for this purpose [FOX 1988].

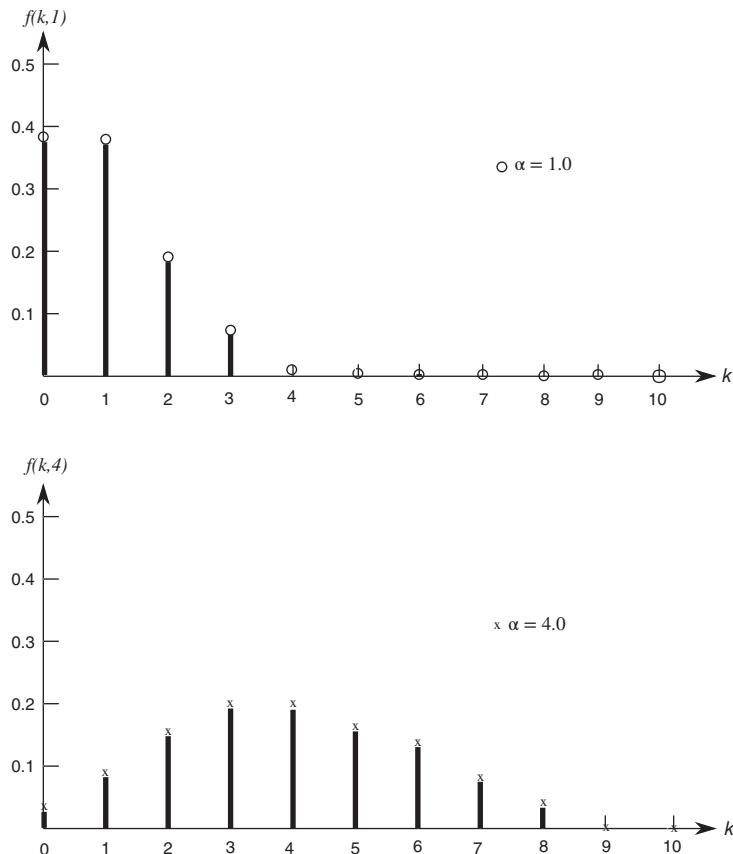


Figure 2.13. Poisson pmf

The Poisson probabilities have been tabulated [PEAR 1966] for $\alpha = 0.1$ to 15, in the increments of 0.1 (in Appendix C, we have tabulated the Poisson CDF). In Figure 2.13, we have plotted the Poisson pmf with parameters $\alpha = 1$ and $\alpha = 4$. Note that this pmf is positively skewed; in fact, it can be shown that the Poisson pmf is positively skewed for any $\alpha > 0$.

Apart from its ability to approximate a binomial pmf, the Poisson pmf is found to be useful in many other situations. In reliability theory, it is quite reasonable to assume that the probability of k components malfunctioning within an interval of time t in a system with a large number of components is given by the Poisson pmf (here λ is known as the *component failure rate*):

$$f(k; \lambda t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad k = 0, 1, 2, \dots \quad (2.16)$$

In studying systems with congestion (or queuing), we find that the number of jobs arriving, the number of jobs completing service, or the number of messages transmitted through a communication channel in a fixed interval of time is approximately Poisson distributed.

2.5.6 The Hypergeometric pmf

We have noted earlier that the binomial pmf is obtained while “sampling with replacement.” The hypergeometric pmf is obtained while “sampling without replacement.” Let us select a random sample of m components from a box containing n components, d of which are known to be defective. For the first component selected, the probability that it is defective is given by d/n , but for the second selection it remains d/n only if the first component selected is replaced. Otherwise, this probability is $(d-1)/(n-1)$ or $d/(n-1)$ depending on whether or not a defective component was selected in the first drawing. Thus the assumption of a constant probability of success, as in a sequence of Bernoulli trials, is not satisfied.

We are interested in computing the **hypergeometric pmf**, $h(k; m, d, n)$, defined as the probability of choosing k defective components in a random sample of m components, chosen without replacement, from a total of n components, d of which are defective. The sample space of this experiment consists of $\binom{n}{m}$ sample points. The k defectives can be selected from d defectives in $\binom{d}{k}$ ways, and the $m - k$ non-defective components may be selected from $n - d$ non-defectives in $\binom{n-d}{m-k}$ ways. Therefore, the whole sample of m components with k defectives can be selected in $\binom{d}{k} \cdot \binom{n-d}{m-k}$ ways. Assuming an equiprobable sample space, the required probability is

$$h(k; m, d, n) = \frac{\binom{d}{k} \cdot \binom{n-d}{m-k}}{\binom{n}{m}}, \quad k = 0, 1, 2, \dots, \min\{d, m\}. \quad (2.17)$$

Example 2.11

Compute the probability of obtaining three defectives in a sample of size 10 taken without replacement from a box of twenty components containing four defectives.

We are required to compute

$$\begin{aligned} h(3; 10, 4, 20) &= \frac{\binom{4}{3} \cdot \binom{16}{7}}{\binom{20}{10}} \\ &= \frac{4 \cdot 11,440}{184,756} \\ &= 0.247678. \end{aligned}$$

If we were to approximate this probability using a binomial distribution with $n = 10$ and $p = 4/20 = 0.20$, we will get $b(3; 10, 0.20) = 0.2013$, a considerable underestimate of the actual probability.

‡

Example 2.12

Return to the TDMA (time division multiple access) wireless system example from Chapter 1 [SUN 1999], where the base transceiver system of each cell has n base repeaters [also called *base radio* (BR)]. Each base repeater provides m time-division-multiplexed channels.

A base repeater is subject to failure. Suppose the channels are allocated randomly to the users. Denote the total number of talking channels in the whole system as k when the failure occurs. Then the probability that i talking channels reside in the failed base repeater, is given by $p_i = h(i; k, m, mn)$.

‡

Example 2.13

A software reliability growth model for estimating the number of residual faults in the software after testing phase based on hypergeometric distribution has been proposed [TOHM 1989].

During the testing phase a software is subjected to a sequence of test instances t_i , $i = 1, 2, \dots, n$. Faults detected by each test instance are assumed to have been removed without introducing new faults before the next test instance is exercised. Assume that the total number of faults initially introduced into the software is m . The test instance t_i senses w_i initial faults out of m initial faults. The sensitization of the faults is distinguished from the detection of the faults in the following way. The number of faults detected by the first test instance t_1 is obviously w_1 . However, the number of faults detected by t_2 is not necessarily w_2 , because some faults may be removed already in t_1 . Similarly, faults detected by t_3 are those that are not yet sensed by t_1 and t_2 .

If the number of faults detected by t_i is denoted by N_i then the cumulative number of faults detected by test instances from t_1 to t_i is given by the random variable:

$$C_i = \sum_{j=1}^i N_j;$$

that is, the number of faults still remaining undetected in the software after the test instance t_i is $m - C_i$. It follows that the probability that k faults are detected by the test instance t_{i+1} given that c_i faults are detected by test instances t_1 through t_i is

$$P(N_{i+1} = k) = h(k; w_{i+1}, m - c_i, m) = \frac{\binom{m - c_i}{k} \binom{c_i}{w_{i+1} - k}}{\binom{m}{w_{i+1}}}.$$

‡

In situations where the sample size m is small compared to the lot size n , the binomial distribution provides a good approximation to the hypergeometric distribution; that is, $h(k; m, d, n) \simeq b(k; m, d/n)$ for large n .

2.5.7 The Discrete Uniform pmf

Let X be a discrete random variable with a finite image $\{x_1, x_2, \dots, x_n\}$. One of the simplest pmf's to consider in this case is one in which each value in the image has equal probability. If we require that $p_x(x_i) = p$ for all i , then, since

$$1 = \sum_{i=1}^n p_x(x_i) = \sum_{i=1}^n p = np,$$

it follows that

$$p_x(x_i) = \begin{cases} \frac{1}{n} & x_i \text{ in the image of } X, \\ 0 & \text{otherwise.} \end{cases}$$

Such a random variable is said to have a **discrete uniform distribution**. This distribution plays an important role in the theory of random numbers and its applications to discrete event simulation. In the average-case analysis of programs, it is often assumed that the input data are uniformly distributed over the input space.

Note that the concept of uniform distribution cannot be extended to a discrete random variable with a countably infinite image, $\{x_1, x_2, \dots\}$. The requirements that $\sum_i p_x(x_i) = 1$ and $p_x(x_i) = \text{constant}$ (for $i = 1, 2, \dots$) are incompatible.

If we let X take on the values $\{1, 2, \dots, n\}$ with $p_x(i) = 1/n, 1 \leq i \leq n$, then its distribution function is given by

$$\begin{aligned} F_X(x) &= \sum_{i=1}^{\lfloor x \rfloor} p_x(i) \\ &= \frac{\lfloor x \rfloor}{n}, \quad 1 \leq x \leq n. \end{aligned}$$

A graph of this distribution with $n = 10$ is given in Figure 2.14.

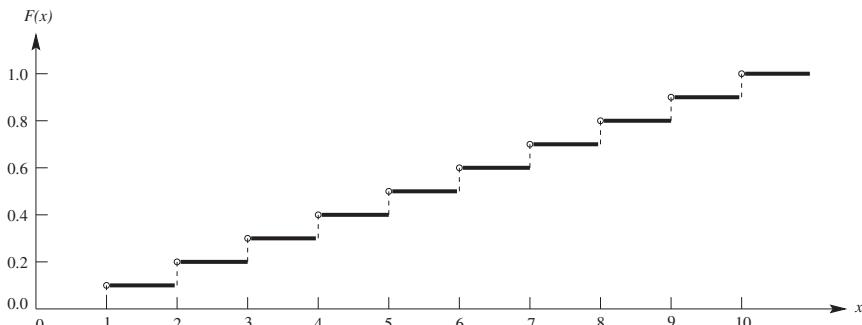


Figure 2.14. Discrete uniform distribution

2.5.8 Constant Random Variable

For a real number c , the function X defined by $X(s) = c$ for all s in S is a discrete random variable. Clearly, $P(X = c) = 1$. Therefore the pmf of this random variable is given by

$$p_X(x) = \begin{cases} 1 & \text{if } x = c, \\ 0 & \text{otherwise.} \end{cases}$$

Such a random variable is called a **constant random variable**.

The distribution function of X is given by

$$F_X(x) = \begin{cases} 0 & \text{for } x < c, \\ 1 & \text{for } x \geq c, \end{cases}$$

and is shown in Figure 2.15.

2.5.9 Indicator Random Variable

Assume that event A partitions the sample space S into two mutually exclusive and collectively exhaustive subsets, A and \bar{A} . The **indicator** of event A is a random variable I_A defined by

$$I_A(s) = \begin{cases} 1, & \text{if } s \in A, \\ 0, & \text{if } s \in \bar{A}. \end{cases}$$

Then event A occurs if and only if $I_A = 1$. This may be visualized as in Figure 2.16. The pmf of I_A is given by

$$\begin{aligned} p_{I_A}(0) &= P(\bar{A}) \\ &= 1 - P(A) \end{aligned}$$

and

$$p_{I_A}(1) = P(A).$$

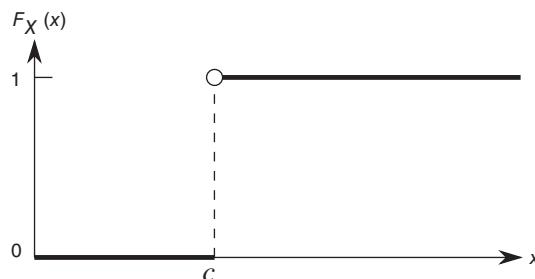


Figure 2.15. CDF of constant random variable

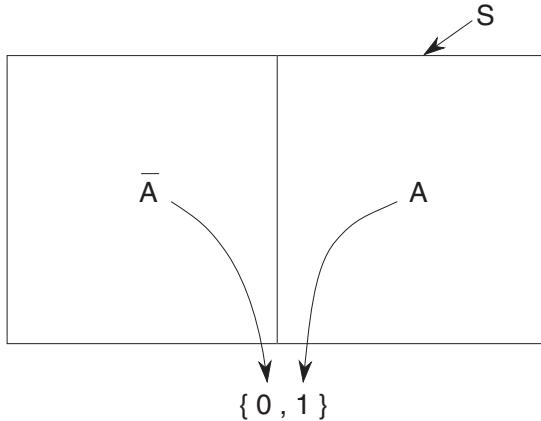


Figure 2.16. Indicator random variable

The concept of the indicator function in certain cases allows us to make efficient computations, without a detailed knowledge of distribution functions. This is quite useful, particularly in cases where the distribution is difficult to calculate. Now if X is a Bernoulli random variable with parameter p and image $\{0, 1\}$, then X is the indicator of the event

$$A = \{s | X(s) = 1\}$$

and

$$\begin{aligned} p_x(0) &= P(\bar{A}) \\ &= 1 - P(A) \\ &= 1 - p \end{aligned}$$

and

$$\begin{aligned} p_x(1) &= P(A) \\ &= p. \end{aligned}$$

Problems

1. Show that the limit as $k \rightarrow \infty$ of $P_c(k)$ is zero in equation (2.6).
2. Out of a job population of ten jobs with six jobs of class 1 and four of class 2, a random sample of size n is selected. Let X be the number of class 1 jobs in the sample. Calculate the pmf of X if the sampling is (a) without replacement, (b) with replacement.
3. A mischievous student wants to break into a computer file, which is password-protected. Assume that there are n equally likely passwords, and that the student chooses passwords independently and at random and tries

them. Let N_n be the number of trials required to break into the file. Determine the pmf of N_n (a) if unsuccessful passwords are not eliminated from further selections, and (b) if they are.

4. A telephone call may pass through a series of trunks before reaching its destination. If the destination is within the caller's own local exchange, then no trunks will be used. Assume that the number of trunks used, X , is a modified geometric random variable with parameter p . Define Z to be the number of trunks used for a call directed to a destination outside the caller's local exchange. What is the pmf of Z ? Given that a call requires at least three trunks, what is the conditional pmf of the number of trunks required?
5. Assume that the probability of error-free transmission of a message over a communication channel is p . If a message is not transmitted correctly, a retransmission is initiated. This procedure is repeated until a correct transmission occurs. Such a channel is often called a **feedback channel**. Assuming that successive transmissions are independent, what is the probability that no retransmissions are required? What is the probability that exactly two retransmissions are required?
6. One percent of faults occurring in a highly available system need the actual repair or replacement of component(s) while the remaining 99% are cleared by a reboot. Find the probability that among a sample of 200 faults there are no faults that require calling the repair person. (*Hint:* You may use the Poisson approximation to the binomial distribution.)
7. Five percent of the disk controllers produced by a plant are known to be defective. A sample of 15 controllers is drawn randomly from each month's production and the number of defectives noted. What proportion of these monthly samples would have at least two defective controllers?
8. The probability of error in the transmission of a bit over a communication channel is $p = 10^{-4}$. What is the probability of more than three errors in transmitting a block of 1000 bits?
9. Assume that the number of messages input to a communication channel in an interval of duration t seconds is Poisson distributed with parameter $0.3t$. Compute the probabilities of the following events:
 - (a) Exactly three messages will arrive during a 10 s interval
 - (b) At most 20 messages arrive in a period of 20 s
 - (c) The number of message arrivals in an interval of 5 s duration is between three and seven.
10. VLSI chips, essential to the running of a computer system, fail in accordance with a Poisson distribution with the rate of one chip in about 5 weeks. If there are two spare chips on hand, and if a new supply will arrive in 8 weeks, what is the probability that during the next 8 weeks the system will be down for a week or more, owing to a lack of chips?

2.6 ANALYSIS OF PROGRAM MAX

We will now apply some of the techniques of the preceding sections to the analysis of a typical algorithm. Given an array of n elements, $B[0], B[1], \dots, B[n - 1]$, we will find m and j such that $m = B[j] = \max\{B[k] \mid 0 \leq k \leq n - 1\}$, and for which j is as large as possible. In other words, the **C program MAX**, shown below, finds the largest element in the given array B . Our discussion here closely parallels that in [KNUT 1997].

```
#define n 100

MAX()
{
    int j, k, m;
    int B[n];

    j = n-1;    k = n - 2;    m = B[n-1];
    while (k >= 0) {
        if (B[k] > m) {
            j = k;
            m = B[k];
        }
        k = k - 1;
    }
    printf(“%d, %d\n”, j, m);
}
```

There are at least two aspects of the analysis of an algorithm: the storage space required and the execution time. Since the storage space required by the program MAX is fixed, we will analyze only the time required for its execution. The execution time depends, in general, on the machine on which it is executed, the compiler used to translate the program, and the input data supplied to it. We are interested in studying the effect of the input data on the execution time. It is convenient to abstract and study the frequency counts for each of the steps. In this way, we need not consider the details of the machine and the compiler used. Counting the number of times each step is executed is facilitated by drawing a flowchart as in Figure 2.17. Noting that the amount of flow into each node must equal the amount of flow out of the node, we obtain the following table:

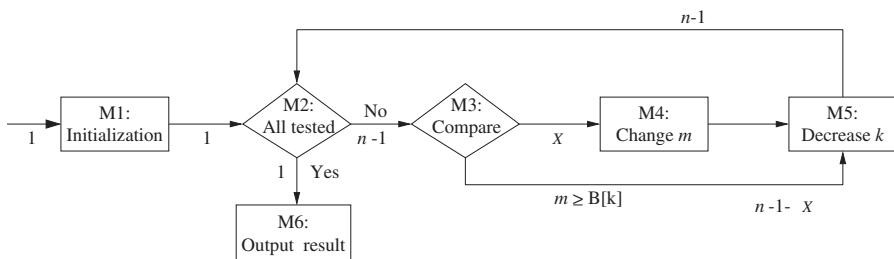


Figure 2.17. Flowchart of MAX

Step number	Frequency count	Number of statements
M1	1	3
M2	n	1
M3	$n - 1$	1
M4	X	2
M5	$n - 1$	1
M6	1	1

This table gives us the information necessary to determine the execution time of program MAX on a given computer. In this table, everything except the quantity X is known. Here X is the number of times we must change the value of the current maximum. The value of X will depend on the pattern of numbers constituting the elements of the array B . As these sets of numbers vary over some specified set, the value of X will also change. Each such pattern of numbers may be considered a sample point with a fixed assigned probability. Then X may be thought of as a random variable over the sample space. We are interested in studying the distribution of the random variable X for a given assignment of probabilities over the sample space.

Clearly, the image of the random variable X is $\{0, 1, \dots, n - 1\}$. The minimum value of X occurs when $B[n - 1] = \max\{B[k] \mid 0 \leq k \leq n - 1\}$, and the maximum value of X occurs when $B[0] > B[1] > \dots > B[n - 1]$.

To simplify our analysis, we will assume that the $B[k]$ are distinct values. Furthermore, without a loss of generality, assume that the vector of elements $(B[0], B[1], \dots, B[n - 1])$ is any one of $n!$ permutations of the integers $\{1, 2, \dots, n\}$. Thus the sample space S_n is the set of all permutations of the n integers $\{1, 2, \dots, n\}$. Finally, we assume that all $n!$ permutations are equally likely. Therefore, for all s in S_n , $P(s) = 1/n!$. We may define the random variable X_n as a function with domain S_n and the image $\{0, 1, \dots, n - 1\}$. As n changes, we have a sequence of random variables X_1, X_2, \dots , where X_i is defined on the sample space S_i .

The probability mass function of X_n , $p_{X_n}(k)$, will be denoted by p_{nk} . Then

$$\begin{aligned} p_{nk} &= P(X_n = k) \\ &= \frac{\text{number of permutations of } n \text{ objects for which } X_n = k}{n!}. \end{aligned}$$

We will establish a recurrence relation for p_{nk} .

Consider a sample point $s = (b_1, b_2, \dots, b_n)$, a permutation on $\{1, 2, \dots, n\}$, and consider two mutually exclusive and collectively exhaustive events:

$$A = "b_1 = n"$$

and

$$\bar{A} = "b_1 \neq n".$$

If event A occurs, then a comparison with b_1 (in program MAX) will force a change in the value of m . Therefore the value obtained for X_n will be one higher than a similar value obtained while examining the previous $n - 1$ elements (b_2, \dots, b_n) . Note that (b_2, \dots, b_n) is a permutation on $\{1, 2, \dots, n - 1\}$. Therefore the number of times the value of m gets changed while examining (b_2, \dots, b_n) is X_{n-1} . From these observations, we conclude that

$$\begin{aligned} P(X_n = k | A) &= P(X_{n-1} = k - 1) \\ &= p_{n-1,k-1}. \end{aligned}$$

On the other hand, the occurrence of event \bar{A} implies that the count of exchanges does not change when we examine b_1 :

$$\begin{aligned} P(X_n = k | \bar{A}) &= P(X_{n-1} = k) \\ &= p_{n-1,k}. \end{aligned}$$

Now, by the assumption of equiprobable sample space, we have $P(A) = 1/n$ and $P(\bar{A}) = (n - 1)/n$. Then by the theorem of total probability, we conclude that

$$\begin{aligned} p_{nk} &= P(X_n = k) \\ &= P(X_n = k | A)P(A) + P(X_n = k | \bar{A})P(\bar{A}) \\ &= \frac{1}{n}p_{n-1,k-1} + \frac{n-1}{n}p_{n-1,k}. \end{aligned} \tag{2.18}$$

This equation will allow us to recursively compute p_{nk} if we provide the initial conditions. Since the image of X_n is $\{0, 1, \dots, n - 1\}$, we know that $p_{nk} = 0$ if $k < 0$. Next consider the random variable X_1 . With $n = 1$, we observe that

the **while** loop in program MAX will never be executed. Therefore $X_1 = 0$; that is, X_1 is a constant random variable, with $P(X_1 = 0) = p_{1,0} = 1$ and $P(X_1 = 1) = p_{1,1} = 0$. Thus the complete specification to evaluate p_{nk} is

$$\begin{aligned} p_{1,0} &= 1, \\ p_{1,1} &= 0, \\ p_{nk} &= \begin{cases} \frac{1}{n}p_{n-1,k-1} + \frac{n-1}{n}p_{n-1,k}, & 0 \leq k \leq n-1, \quad n \geq 2, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (2.19)$$

Generating functions are a convenient tool for evaluation of quantities defined by such recurrence relations. In the context of discrete random variables, these functions will be referred to as **probability generating functions** (PGFs). They will be discussed in the next section. For the moment, let us study the pmf of the random variable X_2 . The image of X_2 is $\{0, 1\}$, and from the preceding recurrence relation,

$$p_{2,0} = 1/2 \text{ and } p_{2,1} = 1/2.$$

Thus X_2 is a Bernoulli random variable with the parameter $p = 1/2$.

Another quantity of interest is the probability $p_{n0} = P(X_n = 0)$. From the preceding recurrence relation

$$\begin{aligned} p_{n0} &= \frac{1}{n}p_{n-1,-1} + \frac{n-1}{n}p_{n-1,0} \\ &= \frac{n-1}{n}p_{n-1,0} \\ &= \frac{n-1}{n} \cdot \frac{n-2}{n-1} \cdots \frac{1}{2}p_{1,0} \\ &= \frac{(n-1)!}{n!} \\ &= \frac{1}{n}. \end{aligned}$$

Alternatively, this result could be obtained by observing that no changes to the value of m will be required if $b_n = B[n] = n$, since m will be set equal to the largest value n before entering the **while** loop. Now out of $n!$ permutations, $(n-1)!$ of them have $b_n = n$, therefore we get the required result.

Problems

1. Explicitly determine the pmf of random variable X_3 , the number of exchanges in program MAX with array size $n = 3$.

2.7 THE PROBABILITY GENERATING FUNCTION

The notion of probability generating functions (PGFs) is a convenient tool that simplifies computations involving integer-valued, discrete random variables. Given a nonnegative integer-valued discrete random variable X with $P(X = k) = p_k$, define the PGF of X by

$$\begin{aligned} G_X(z) &= \sum_{i=0}^{\infty} p_i z^i \\ &= p_0 + p_1 z + p_2 z^2 + \cdots + p_k z^k + \cdots . \end{aligned}$$

$G_X(z)$, also known as the z -transform of X , converges for any complex number z such that $|z| < 1$. It may be easily verified that

$$G_X(1) = 1 = \sum_{i=0}^{\infty} p_i.$$

In many problems we will know the PGF $G_X(z)$, but we will not have explicit knowledge for the pmf of X . Later we will see that we can determine interesting quantities such as the mean and variance of X from the PGF itself. One reason for this is found in the following theorem, which we quote without proof.

THEOREM 2.1. If two discrete random variables X and Y have the same PGFs, then they must have the same distributions and pmf's.

If we can show that a random variable that is under investigation has the same PGF as that of another random variable with a known pmf, then this theorem assures us that the pmf of the original random variable must be the same.

Continuing with our analysis of program MAX, define the PGF of X_n as

$$G_{X_n}(z) = \sum_{k \geq 0} p_{nk} \cdot z^k.$$

$G_{X_n}(z)$ is actually a polynomial, even though an infinite sum is specified for convenience. From equation (2.19), we have

$$\begin{aligned} G_{X_1}(z) &= p_{1,0} + p_{1,1} \cdot z \\ &= 1. \end{aligned}$$

Multiplying the recurrence relation (2.18) by z^k and summing for $k = 1$ to infinity, we obtain

$$\sum_{k \geq 1} p_{nk} \cdot z^k = \frac{z}{n} \sum_{k \geq 1} p_{n-1,k-1} \cdot z^{k-1} + \frac{n-1}{n} \sum_{k \geq 1} p_{n-1,k} \cdot z^k.$$

Thus

$$G_{X_n}(z) - p_{n0} = \frac{z}{n} G_{X_{n-1}}(z) + \frac{n-1}{n} [G_{X_{n-1}}(z) - p_{n-1,0}].$$

Noting that $p_{n0} = 1/n$ and simplifying, we get

$$\begin{aligned} G_{X_n}(z) &= \frac{(z+n-1)}{n} G_{X_{n-1}}(z) \\ &= \frac{(z+n-1)}{n} \cdot \frac{(z+n-2)}{n-1} \cdots \frac{(z+1)}{2} G_{X_1}(z) \\ &= \frac{(z+n-1)(z+n-2)\cdots(z+1)}{n!}. \end{aligned} \quad (2.20)$$

To obtain an explicit expression for p_{nk} , we must expand $G_{X_n}(z)$ into a power series of z . Stirling numbers of the first kind, denoted by $\left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right]$, can be used for this purpose. Stirling numbers are defined by [KNUT 1997, p. 65]:

$$\begin{aligned} x(x-1)\cdots(x-n+1) &= \left[\begin{smallmatrix} n \\ n \end{smallmatrix} \right] x^n - \left[\begin{smallmatrix} n \\ n-1 \end{smallmatrix} \right] x^{n-1} \\ &\quad + \cdots + (-1)^n \left[\begin{smallmatrix} n \\ 0 \end{smallmatrix} \right] \\ &= \sum_{k=0}^n (-1)^{n-k} \left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right] x^k. \end{aligned}$$

Substituting $x = -z$ in this formula, we get

$$z(z+1)\cdots(z+n-1) = \sum_{k=0}^n \left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right] z^k.$$

Then, using (2.20), we have

$$\begin{aligned} G_{X_n}(z) &= \frac{(z+1)(z+2)\cdots(z+n-1)}{n!} \\ &= \frac{1}{n!} \sum_{k=0}^n \left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right] z^{k-1}. \end{aligned}$$

Therefore

$$p_{nk} = \frac{\left[\begin{smallmatrix} n \\ k+1 \end{smallmatrix} \right]}{n!}. \quad (2.21)$$

Thus the pmf of the random variable X_n is described by the Stirling numbers of the first kind.

At this point it is useful to derive the PGFs of some of the distributions studied in Section 2.5. When the random variable X is understood, we will use the abbreviated notation $G(z)$ for its PGF.

1. *The Bernoulli random variable*

$$\begin{aligned} G(z) &= qz^0 + pz^1 \\ &= q + pz \\ &= 1 - p + pz. \end{aligned} \tag{2.22}$$

2. *The binomial random variable*

$$\begin{aligned} G(z) &= \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} z^k \\ &= (pz + 1 - p)^n. \end{aligned} \tag{2.23}$$

3. *The modified geometric random variable*

$$\begin{aligned} G(z) &= \sum_{k=0}^{\infty} p(1-p)^k z^k \\ &= \frac{p}{1 - z(1-p)}. \end{aligned} \tag{2.24}$$

4. *The Poisson random variable*

$$\begin{aligned} G(z) &= \sum_{k=0}^{\infty} \frac{\alpha^k}{k!} e^{-\alpha} z^k \\ &= e^{-\alpha} e^{\alpha z} \\ &= e^{\alpha(z-1)} \\ &= e^{-\alpha(1-z)}. \end{aligned} \tag{2.25}$$

5. *The uniform random variable*

$$\begin{aligned} G(z) &= \sum_{k=1}^n \frac{1}{n} z^k \\ &= \frac{1}{n} \sum_{k=1}^n z^k. \end{aligned} \tag{2.26}$$

6. *The constant random variable.* Let $X = i$ for some $0 \leq i < \infty$; then

$$G(z) = z^i. \tag{2.27}$$

7. *The indicator random variable.* Let $P(I_A = 0) = P(\bar{A}) = 1 - p$ and $P(I_A = 1) = P(A) = p$; then

$$\begin{aligned} G(z) &= (1 - p)z^0 + pz \\ &= 1 - p + pz \\ &= P(\bar{A}) + P(A)z. \end{aligned} \tag{2.28}$$

Problems

- Let X denote the execution time of a job rounded to the nearest second. The charges are based on a linear function $Y = mX + n$ of the execution time for suitably chosen nonnegative integers m and n . Given the PGF of X , find the PGF and pmf of Y .
- Show that the PGF of a geometric random variable with parameter p is given by $pz/(1 - qz)$, where $q = 1 - p$.
- Let X be a negative binomial random variable with parameters n, p , and r . Show that its PGF is given by

$$\left[\frac{p z}{1 - z(1 - p)} \right]^r.$$

2.8 DISCRETE RANDOM VECTORS

Often we may be interested in studying relationships between two or more random variables defined on a given sample space. For example, consider a program consisting of two modules with execution times X and Y , respectively. Since the execution times will depend on input data values and since the execution times will be discrete, we may assume that X and Y are discrete random variables. If the program is organized such that two modules are executed serially, one after the other, then the random variable $Z_1 = X + Y$ gives the total execution time of the program. Alternatively, if the program's two modules were to execute independently and concurrently, then the total program execution time will be given by the random variable $Z_2 = \max\{X, Y\}$, and the time until the completion of the faster module is given by $Z_3 = \min\{X, Y\}$.

Let X_1, X_2, \dots, X_r be r discrete random variables defined on a sample space S . Then, for each sample point s in S , each of the random variables X_1, X_2, \dots, X_r takes on one of its possible values, as

$$X_1(s) = x_1, X_2(s) = x_2, \dots, X_r(s) = x_r.$$

The random vector $\mathbf{X} = (X_1, X_2, \dots, X_r)$ is an r -dimensional vector-valued function $\mathbf{X} : S \rightarrow \Re^r$ with $\mathbf{X}(s) = \mathbf{x} = (x_1, x_2, \dots, x_r)$. Thus, a discrete r -dimensional random vector \mathbf{X} is a function from S to \Re^r taking on a finite or countably infinite set of vector values $\mathbf{x}_1, \mathbf{x}_2, \dots$

Definition (Joint pmf). The **compound** (or **joint**) pmf for a random vector \mathbf{X} is defined to be

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x}) &= P(\mathbf{X} = \mathbf{x}) \\ &= P(X_1 = x_1, X_2 = x_2, \dots, X_r = x_r). \end{aligned}$$

As in the one-dimensional case, the compound pmf has the following four properties:

$$(j1) \quad p_{\mathbf{x}}(\mathbf{x}) \geq 0, \quad \mathbf{x} \in \Re^r.$$

(j2) $\{\mathbf{x} \mid p_{\mathbf{x}}(\mathbf{x}) \neq 0\}$ is a finite or countably infinite subset of \Re^r , which will be denoted by $\{\mathbf{x}_1, \mathbf{x}_2, \dots\}$.

$$(j3) \quad P(\mathbf{X} \in A) = \sum_{\mathbf{x} \in A} p_{\mathbf{x}}(\mathbf{x}).$$

$$(j4) \quad \sum_i p_{\mathbf{x}}(\mathbf{x}_i) = 1.$$

It can be shown that any real-valued function defined on \Re^r having these four properties is the compound pmf of some discrete r -dimensional random vector.

Let us now consider a program with two modules, having module execution times X and Y , respectively. The images of the discrete random variables X and Y are given by $\{1, 2\}$ and $\{1, 2, 3, 4\}$. The compound pmf is described by the following table:

	$y=1$	$y=2$	$y=3$	$y=4$
$x=1$	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$
$x=2$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{16}$

Each possible event $[X = x, Y = y]$ can be pictured as an event point on an (x, y) coordinate system with the value $p_{x,y}(x, y)$ indicated as a bar perpendicular to the (x, y) plane above the event point (x, y) . This can be visualized as in Figure 2.18, where we indicate the value $p_{x,y}(x, y)$ associated with each event by writing it beside the event point (x, y) .

In situations where we are concerned with more than one random variable, the pmf of a single variable, such as $p_x(x)$, is referred to as a **marginal pmf**. Since the eight events shown in Figure 2.18 are collectively exhaustive and mutually exclusive, the marginal pmf is

$$\begin{aligned} p_x(x) &= P(X = x) \\ &= P(\bigcup_j \{X = x, Y = y_j\}) \end{aligned}$$

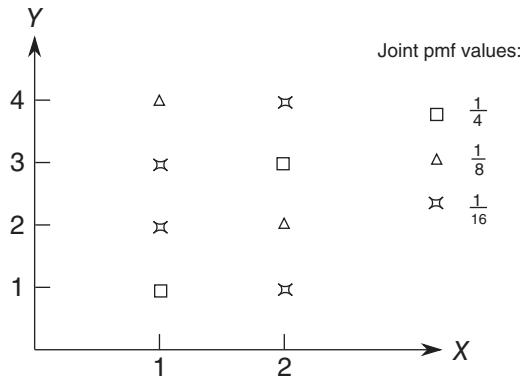


Figure 2.18. Joint pmf for two-module execution problem

$$\begin{aligned}
 &= \sum_j P(X = x, Y = y_j) \\
 &= \sum_j p_{X,Y}(x, y_j).
 \end{aligned}$$

In other words, to obtain the marginal pmf $p_X(x)$, we erect a vertical column at $X = x$ and sum the probabilities of all event points touched by the column. Similarly

$$p_Y(y) = \sum_i p_{X,Y}(x_i, y).$$

In the preceding example, we get

$$\begin{aligned}
 p_X(1) &= \frac{1}{2}, & p_X(2) &= \frac{1}{2}, & p_Y(1) &= \frac{5}{16}, \\
 p_Y(2) &= \frac{3}{16}, & p_Y(3) &= \frac{5}{16}, & p_Y(4) &= \frac{3}{16}.
 \end{aligned}$$

Check that

$$\sum_{i=1}^2 p_X(i) = 1 \quad \text{and} \quad \sum_{i=1}^4 p_Y(i) = 1.$$

The preceding formulas for computing the marginal pmf's from the compound pmf can be easily generalized to the r -dimensional case.

We have seen that the task of obtaining the marginal pmf's from the compound pmf is relatively straightforward. Note that given the marginal pmf's, there is no way to go back, in general, to determine the compound pmf. However, an exception occurs when the random variables are independent. The notion of independence of random variables will be developed in the next section.

Example 2.14

Let X and Y be two random variables, each with image $\{1, 2\}$ and with the compound pmf:

$$\begin{aligned} p_{x,y}(1,1) &= p_{x,y}(2,2) = a, \\ p_{x,y}(1,2) &= p_{x,y}(2,1) = \frac{1}{2} - a, \quad \text{for } 0 \leq a \leq \frac{1}{2}. \end{aligned}$$

It is easy to see that $p_x(1) = p_x(2) = p_y(1) = p_y(2) = \frac{1}{2}$, whatever be the value of a . Thus, we have uncountably many distinct compound pmf's associated with the same marginal pmf's.

‡

An interesting example of a compound pmf is the **multinomial pmf**, which is a generalization of the binomial pmf. Consider a sequence of n generalized Bernoulli trials where there are a finite number r of distinct outcomes having probabilities p_1, p_2, \dots, p_r where $\sum_{i=1}^r p_i = 1$. Define the random vector $\mathbf{X} = (X_1, X_2, \dots, X_r)$ such that X_i is the number of trials that resulted in the i th outcome. Then the compound pmf of \mathbf{X} is given by

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{n}) &= P(X_1 = n_1, X_2 = n_2, \dots, X_r = n_r) \\ &= \binom{n}{n_1 n_2 \cdots n_r} p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r}, \end{aligned} \tag{2.29}$$

where $\mathbf{n} = (n_1, n_2, \dots, n_r)$ and $\sum_{i=1}^r n_i = n$. The marginal pmf of X_i may be computed by

$$\begin{aligned} p_{x_i}(n_i) &= \sum_{\mathbf{n}: \sum_{j \neq i} n_j = n - n_i} \binom{n}{n_1 n_2 \cdots n_r} p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r} \\ &= \frac{n! p_i^{n_i}}{(n - n_i)! (n_i)!} \sum_{\sum_{j \neq i} n_j = n - n_i} \frac{(n - n_i)! p_1^{n_1} \cdots p_{i-1}^{n_{i-1}} p_{i+1}^{n_{i+1}} \cdots p_r^{n_r}}{n_1! n_2! \cdots n_{i-1}! n_{i+1}! \cdots n_r!} \\ &= \binom{n}{n_i} p_i^{n_i} (p_1 + \cdots + p_{i-1} + p_{i+1} + \cdots + p_r)^{n - n_i} \\ &= \binom{n}{n_i} p_i^{n_i} (1 - p_i)^{n - n_i}. \end{aligned}$$

Thus the marginal pmf of each X_i is binomial with parameters n and p_i .

Many practical situations give rise to a multinomial distribution. For example, a program requires I/O service from device i with probability p_i at the end of a CPU burst, with $\sum_{i=1}^r p_i = 1$. This situation is depicted in Figure 2.19. If we observe n CPU burst terminations, then the probability that n_i of these will be directed to I/O device i (for $i = 1, 2, \dots, r$) is given

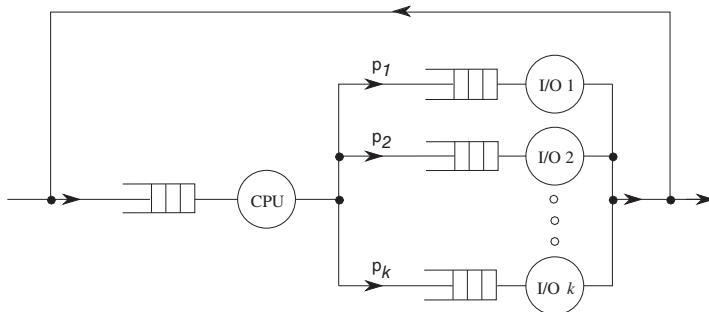


Figure 2.19. I/O queuing at the end of a CPU burst

by the multinomial pmf. Now, if we are just interested in the number of I/O requests (out of n) directed to a specific device j , then it has a binomial distribution with parameters n and p_j . We may also replace the phrase “I/O device” by “file” in this example. Another example of a multinomial distribution occurs when we consider a paging system (or cache memory) and we model a program using the **independent reference model** (see Chapter 7). In this model, we assume that successive page references (or memory references) are independent and the probability of referencing page (or memory block) i is fixed at p_i .

Example 2.15

An inspection plan calls for inspecting five chips and for either accepting each chip, rejecting each chip, or submitting it for reinspection, with probabilities of $p_1 = 0.70$, $p_2 = 0.20$, $p_3 = 0.10$, respectively. What is the probability that all five chips must be reinspected? What is the probability that none of the chips must be reinspected? What is the probability that at least one of the chips must be reinspected?

Let X = number of chips accepted, Y = number of chips rejected; then the remaining $Z = 5 - X - Y$ are sent for reinspection. The compound pmf is

$$p_{X,Y,Z}(i,j,k) = \frac{5!}{i! j! k!} \cdot 0.7^i \cdot 0.2^j \cdot 0.1^k.$$

The answer to the first question is $p_{X,Y,Z}(0,0,5) = 10^{-5}$. The second question pertains to the event

$$\begin{aligned} [X + Y = 5] &= \{s | X(s) + Y(s) = 5\} \\ &= \bigcup_{i+j=5} \{s | X(s) = i \text{ and } Y(s) = j \text{ and } Z(s) = 0\} \end{aligned}$$

and therefore

$$\begin{aligned}
 P(X + Y = 5) &= \sum_{i+j=5} p_{X,Y,Z}(i,j,k) \\
 &= \sum_{i+j=5} \frac{5!}{i! j!} p_1^i p_2^j p_3^0 \\
 &= \sum_{i=0}^5 \binom{5}{i} p_1^i p_2^{5-i} \\
 &= (p_1 + p_2)^5 \\
 &= (0.7 + 0.2)^5 \\
 &= 0.59.
 \end{aligned}$$

To answer the third question, note that the event {“at least one chip reinspected”} = $S - \{\text{“none reinspected”}\}$, but $P(\text{“none reinspected”}) = P(X + Y = 5) = 0.59$. Therefore, the required answer is

$$1 - 0.59 = 0.41.$$

‡

Problems

1. Two discrete random variables X and Y have joint pmf given by the following table:

		Y		
		1	2	3
X	1	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
	2	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{12}$
	3	$\frac{1}{12}$	$\frac{1}{12}$	0

Compute the probability of each of the following events:

- (a) $X \leq 1\frac{1}{2}$.
 - (b) X is odd.
 - (c) XY is even.
 - (d) Y is odd given that X is odd.
2. Each telephone call passes through a number of switching offices before reaching its destination. The number and types of switching offices in the United States in 1971 were estimated to be as follows:

	$i = 1$	2	3	4
<i>Office type</i>	Step-by-step	Panel	Crossbar	ESS ^a
<i>Number</i>	8600	500	5700	286

^aElectronic Switching System

Note that the type of switching office in the local exchange of the originating call is fixed. Let n be the number of switching offices encountered (other than the local exchange) by a telephone call. Assuming that each switching office is randomly and independently chosen from the population, determine the probability that the call passes through exactly n_i ($i = 1, 2, 3, 4$) switching offices of type i , where $n_1 + n_2 + n_3 + n_4 = n$. Determine the marginal pmf for each type of office.

2.9 INDEPENDENT RANDOM VARIABLES

We have noted that the problem of determining the compound pmf given the marginal pmf's does not have a unique solution, unless the random variables are independent.

Definition (Independent Random Variables). Two discrete random variables X and Y are defined to be independent provided their joint pmf is the product of their marginal pmf's:

$$p_{x,y}(x, y) = p_x(x)p_y(y) \quad \text{for all } x \text{ and } y. \quad (2.30)$$

If X and Y are two independent random variables, then for any two subsets A and B of \mathfrak{R} , the events “ X is an element of A ” and “ Y is an element of B ” are independent:

$$P(X \in A \cap Y \in B) = P(X \in A)P(Y \in B).$$

To see this, note that

$$\begin{aligned} P(X \in A \cap Y \in B) &= \sum_{x \in A} \sum_{y \in B} p_{x,y}(x, y) \\ &= \sum_{x \in A} \sum_{y \in B} p_x(x)p_y(y) \\ &= \sum_{x \in A} p_x(x) \sum_{y \in B} p_y(y) \\ &= P(X \in A)P(Y \in B). \end{aligned}$$

To further clarify the notion of independent random variables, assume that on a particular performance of the experiment, the event $[Y = y]$ has been observed, and we want to know the probability that a specific value of X will occur. We write

$$\begin{aligned} P(X = x | Y = y) &= \frac{P(X = x \cap Y = y)}{p_y(y)} \\ &= \frac{p_{x,y}(x, y)}{p_y(y)} \end{aligned}$$

$$\begin{aligned}
 &= \frac{p_x(x)p_y(y)}{p_y(y)} \quad \text{by independence} \\
 &= p_x(x).
 \end{aligned}$$

Thus, if X, Y are independent, then the knowledge that a particular value of Y has been observed does not affect the probability of observing a particular value of X . The notion of independence of two random variables can be easily generalized to r random variables.

Definition. Let X_1, X_2, \dots, X_r be r discrete random variables with pmf's $p_{x_1}, p_{x_2}, \dots, p_{x_r}$, respectively. These random variables are said to be **mutually independent** if their compound pmf p is given by

$$p_{x_1, x_2, \dots, x_r}(x_1, x_2, \dots, x_r) = p_{x_1}(x_1)p_{x_2}(x_2) \cdots p_{x_r}(x_r).$$

In situations involving many random variables, the assumption of mutual independence usually leads to considerable simplification. We note that it is possible for every pair of random variables in the set $\{X_1, X_2, \dots, X_r\}$ to be independent (**pairwise independent**) without the entire set being mutually independent.

Example 2.16

Consider a sequence of two Bernoulli trials and define X_1 and X_2 as the number of successes on the first and second trials respectively. Let X_3 define the number of matches on the two trials. Then it can be shown that the pairs (X_1, X_2) , (X_1, X_3) , and (X_2, X_3) are each independent, but that the set $\{X_1, X_2, X_3\}$ is not mutually independent.

‡

Returning to our earlier example of a program with two modules, let us determine the pmf's of the random variables Z_1, Z_2 , and Z_3 , given that X and Y are independent. Consider the event $[Z_1 = X + Y = t]$. On a two-dimensional (x, y) event space, this event is represented by all the event points on the line $X + Y = t$ (see Figure 2.20). The probability of this event may be computed by adding the probabilities of all the event points on this line. Therefore

$$\begin{aligned}
 P(Z_1 = t) &= \sum_{x=0}^t P(X = x, X + Y = t) \\
 &= \sum_{x=0}^t P(X = x, Y = t - x) \\
 &= \sum_{x=0}^t P(X = x)P(Y = t - x)
 \end{aligned}$$

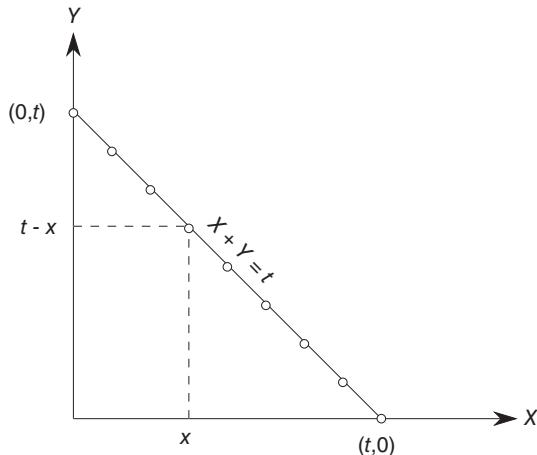


Figure 2.20. Computing the pmf of the random variable $Z_1 = X + Y$

by independence. Thus

$$\begin{aligned} p_{Z_1}(t) &= p_{X+Y}(t) \\ &= \sum_{x=0}^t p_X(x)p_Y(t-x). \end{aligned}$$

This summation is said to represent the **discrete convolution**, and it gives the formula for the pmf of the sum of two nonnegative independent discrete random variables. In the case that X and Y are allowed to take negative values as well, the lower index of summation is changed from 0 to $-\infty$.

Restricting attention to nonnegative integer-valued random variables and recalling the definition of the probability generating function, the PGF of the sum of two independent random variables is the product of their PGFs:

$$\begin{aligned} G_{Z_1}(z) &= G_{X+Y}(z) \\ &= G_X(z)G_Y(z). \end{aligned}$$

To see this, note that

$$\begin{aligned} G_{Z_1}(z) &= \sum_{t=0}^{\infty} p_{Z_1}(t)z^t \\ &= \sum_{t=0}^{\infty} z^t \sum_{x=0}^t p_X(x)p_Y(t-x) \end{aligned}$$

$$\begin{aligned}
&= \sum_{x=0}^{\infty} p_X(x) z^x \sum_{t=x}^{\infty} p_Y(t-x) z^{t-x} \\
&= \sum_{x=0}^{\infty} p_X(x) z^x \sum_{y=0}^{\infty} p_Y(y) z^y \\
&= G_X(z)G_Y(z),
\end{aligned}$$

which is the desired result.

It follows by induction that if X_1, X_2, \dots, X_r are mutually independent nonnegative integer-valued random variables, then

$$G_{X_1+X_2+\dots+X_r}(z) = G_{X_1}(z)G_{X_2}(z)\cdots G_{X_r}(z). \quad (2.31)$$

This result is useful in proving the following theorem.

THEOREM 2.2. Let X_1, X_2, \dots, X_r be mutually independent.

- (a) If X_i has the binomial distribution with parameters n_i and p , then $\sum_{i=1}^r X_i$ has the binomial distribution with parameters $n_1 + n_2 + \dots + n_r$ and p .
- (b) If X_i has the (modified) negative binomial distribution with parameters α_i and p , then $\sum_{i=1}^r X_i$ has the (modified) negative binomial distribution with parameters $\alpha_1 + \alpha_2 + \dots + \alpha_r$ and p .
- (c) If X_i has the Poisson distribution with parameter α_i , then $\sum_{i=1}^r X_i$ has the Poisson distribution with parameter $\sum_{i=1}^r \alpha_i$.

Proof: First note that

$$G_{\sum_{i=1}^r X_i}(z) = G_{X_1}(z)G_{X_2}(z)\cdots G_{X_r}(z).$$

- (a) If X_i obeys $b(k; n_i, p)$, then

$$G_{X_i}(z) = (pz + 1 - p)^{n_i}.$$

Therefore

$$G_{\sum X_i}(z) = (pz + 1 - p)^{\sum_{i=1}^r n_i}.$$

But this implies that $\sum_{i=1}^r X_i$ obeys $b(k; \sum_{i=1}^r n_i, p)$, as was to be shown. The proofs of parts (b) and (c) are left as an exercise.

This theorem can be visualized with Figure 2.21.

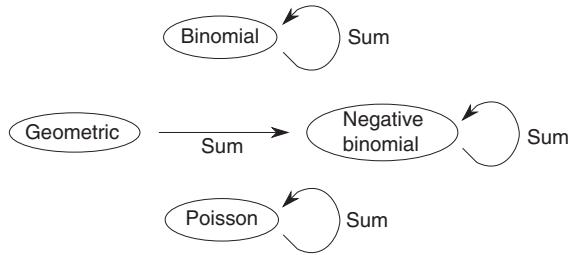


Figure 2.21. Theorem 2.2

Returning now to our example of a program with two modules, if we assume that X and Y are both geometrically distributed with parameter p , then we know that the total serial execution time $Z_1 = X + Y$ is negative binomially distributed with parameters 2 and p (note that the geometric distribution is a negative binomial distribution with parameters 1 and p).

Let us proceed to compute the distribution of $Z_2 = \max\{X, Y\}$ for the above example. Note that the event

$$\begin{aligned} [Z_2 \leq t] &= \{s | \max\{X(s), Y(s)\} \leq t\} \\ &= \{s | X(s) \leq t \text{ and } Y(s) \leq t\} = [X \leq t \text{ and } Y \leq t]. \end{aligned}$$

Therefore

$$\begin{aligned} F_{Z_2}(t) &= P(Z_2 = \max\{X, Y\} \leq t) \\ &= P(X \leq t \text{ and } Y \leq t) \\ &= P(X \leq t)P(Y \leq t) \quad \text{by independence} \\ &= F_X(t)F_Y(t). \end{aligned} \tag{2.32}$$

Thus the CDF of $\max\{X, Y\}$ of two independent random variables X, Y is the product of their CDFs.

Next consider the random variable $Z_3 = \min\{X, Y\}$. First compute

$$\begin{aligned} P(Z_3 = \min\{X, Y\} > t) &= P(X > t \text{ and } Y > t) \\ &= P(X > t)P(Y > t) \quad \text{by independence.} \end{aligned} \tag{2.33}$$

But $P(X > t) = 1 - F_X(t)$. Therefore

$$1 - F_{Z_3}(t) = [1 - F_X(t)][1 - F_Y(t)]$$

or

$$F_{Z_3}(t) = F_X(t) + F_Y(t) - F_X(t)F_Y(t). \tag{2.34}$$

This last expression can be alternatively derived by first defining the events $A = [X \leq t]$, $B = [Y \leq t]$, and $C = [Z_3 \leq t]$ and noting that

$$P(C) = P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

If we assume that X and Y are (modified) geometrically distributed with parameter p , then

$$p_x(i) = p(1-p)^i$$

and

$$p_y(j) = p(1-p)^j.$$

Also

$$\begin{aligned} F_X(k) &= \sum_{i=0}^k p(1-p)^i \\ &= \frac{p[1 - (1-p)^{k+1}]}{[1 - (1-p)]} \\ &= 1 - (1-p)^{k+1} \end{aligned}$$

and

$$F_Y(k) = 1 - (1-p)^{k+1}. \quad (2.35)$$

Then by (2.34) we have

$$\begin{aligned} F_{Z_3}(k) &= 2[1 - (1-p)^{k+1}] - [1 - 2(1-p)^{k+1} + (1-p)^{2(k+1)}] \\ &= 1 - (1-p)^{2(k+1)} \\ &= 1 - [(1-p)^2]^{k+1}. \end{aligned} \quad (2.36)$$

From this, we conclude that Z_3 is also (modified) geometrically distributed with parameter $1 - (1-p)^2 = 2p - p^2$. In general, $\min\{X_1, X_2, \dots, X_r\}$ is geometrically distributed if each X_i ($1 \leq i \leq r$) is geometrically distributed, given that X_1, X_2, \dots, X_r are mutually independent.

Let us consider the event that the module 2 takes longer to finish than module 1; that is, $[Y \geq X]$. Then, from Figure 2.22, we obtain

$$\begin{aligned} P(Y \geq X) &= \sum_{x=0}^{\infty} P(X = x, Y \geq x) \\ &= \sum_{x=0}^{\infty} P(X = x)P(Y \geq x) \quad \text{by independence.} \end{aligned}$$

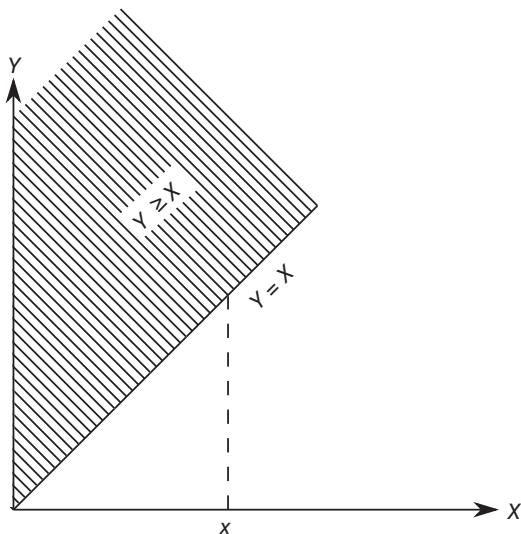


Figure 2.22. Graph for two-module execution problem

Now, since Y has a modified geometric distribution, we have

$$P(Y \geq x) = 1 - F_Y(x-1) = (1-p)^x.$$

Therefore

$$\begin{aligned}
 P(Y \geq X) &= \sum_{x=0}^{\infty} p(1-p)^x(1-p)^x \\
 &= p \sum_{x=0}^{\infty} [(1-p)^2]^x \\
 &= \frac{p}{1-(1-p)^2} \\
 &= \frac{p}{2p-p^2} \\
 &= \frac{1}{2-p}.
 \end{aligned} \tag{2.37}$$

We may also compute

$$\begin{aligned}
 P(Y = X) &= \sum_{x=0}^{\infty} P(X = x, Y = x) \\
 &= \sum_{x=0}^{\infty} P(X = x)P(Y = x),
 \end{aligned}$$

since X and Y are independent, it follows that

$$\begin{aligned} P(Y = X) &= \sum_{x=0}^{\infty} p(1-p)^x p(1-p)^x \\ &= \frac{p^2}{2p - p^2} \\ &= \frac{p}{2 - p}. \end{aligned}$$

Thus if $p = 1/2$, then there is a 33% chance that both modules will take exactly the same time to finish. Similar events will be seen to occur with probability zero when X and Y are continuous random variables.

Finally, consider the conditional probability of the event $[Y = y]$ given that $[X + Y = t]$:

$$\begin{aligned} P(Y = y|X + Y = t) &= \frac{P(Y = y \text{ and } X + Y = t)}{P(X + Y = t)} \\ &= \frac{P(X = t - y, Y = y)}{P(X + Y = t)} \\ &= \frac{P(X = t - y)P(Y = y)}{P(X + Y = t)} \quad \text{by independence.} \end{aligned}$$

Recall that $X + Y$ has a modified negative binomial pmf with parameters 2 and p [see formula (2.13)] while X and Y have the pmf given by (2.10). Therefore

$$\begin{aligned} P(Y = y|X + Y = t) &= \frac{p(1-p)^{t-y}p(1-p)^y}{p^2(1+t)(1-p)^t} \\ &= \frac{1}{t+1}. \end{aligned}$$

Thus given that the total serial execution time was t units, the execution time of the second module is distributed uniformly over $\{0, 1, \dots, t\}$.

Problems

1. Consider two program segments:

```
 $S_1$ : while ( $B_1$ ) {
    printf("hey you!\n");
    printf("finished\n");
}
```

and

```

 $S_2:$  if ( $B_2$ )
    printf("hey you!\n");
else
    printf("finished\n");

```

Assuming that B_1 is true with probability p_1 and B_2 is true with probability p_2 , compute the pmf of the number of times “hey you!” is printed and compute the pmf of the number of times “finished” is printed by the following program:

$\{S_1; S_2;\}$

2. Complete the proofs of parts (b) and (c) of Theorem 2.2.
3. Prove Theorem 2.2 for $r = 2$ without using generating functions—that is, directly using the convolution formula for the pmf of the sum of two independent random variables.
4. Reconsider the example of a program with two modules and assume that respective module execution times X and Y are independent random variables uniformly distributed over $\{1, 2, \dots, n\}$. Find
 - (a) $P(X \geq Y)$.
 - (b) $P(X = Y)$.
 - (c) The pmf and the PGF of $Z_1 = X + Y$.
 - (d) The pmf of $Z_2 = \max\{X, Y\}$.
 - (e) The pmf of $Z_3 = \min\{X, Y\}$.
5. Compute the pmf and the CDF of $\max\{X, Y\}$ where X and Y are independent random variables such that X and Y are both Poisson distributed with parameter α .
6. * Consider a program that needs two stacks. We want to compare two different ways to allocate storage for the two stacks. The first method is to separately allocate n locations to each stack. The second is to let the two stacks grow toward each other in a common area of memory consisting of N locations. If the required value of N is smaller than $2n$, then the latter solution is preferable to the former. Determine the required values of n and N so as to keep the probability of overflow below 5%, assuming:
 - (a) The size of each stack is geometrically distributed with parameter p (use $p = \frac{1}{4}, \frac{1}{2},$ and $\frac{3}{4}$).
 - (b) The size of each stack is Poisson distributed with parameter $\alpha = 0.5$.
 - (c) The size of each stack is uniformly distributed over $\{1, 2, \dots, 20\}$.

Review Problems

1. Consider the combinational switching circuit shown in Figure 2.P.1, with four inputs and one output. The switching function realized by the circuit is easily shown to be

$$y = (x_1 \text{ and } x_2) \text{ or } \overline{(x_3 \text{ and } x_4)}.$$

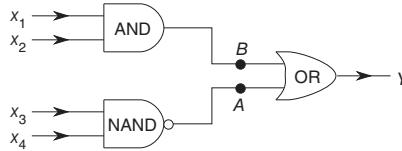


Figure 2.P.1. A combinational circuit

Associate the random variable X_i with the switching variable x_i . Assuming that X_i ($i = 1, 2, 3, 4$) is a Bernoulli random variable with parameter p_i , compute the pmf of the output random variable Y . If a fault develops, then the pmf of Y will change. Assume that only single faults of stuck-at-1 or stuck-at-0 type occur at any one of the input points, at the output point, or at internal points A and B . Compute the pmf of Y for each one of these 14 faulty conditions, assuming that $p_i = b$ for each $i = 1, 2, 3, 4$.

2. See Wetherell [WETH 1980]. Consider a context-free language $L = \{a^n b^n | n \geq 0\}$ as the sample space of an experiment and define the random variable X that maps the string (sample point) $a^n b^n$ into the integer n . Determine the value of the constant k such that the pmf of X is $p_X(n) = k/n!$.
3. * See Burks et al. [BURK 1963]. In designing a parallel binary adder, we are interested in analyzing the length of the longest carry sequence. Assume that the two n -bit integer operands X_n and Y_n are independent random variables, uniformly distributed over $\{0, 1, \dots, 2^n - 1\}$. Let the random variable V_n denote the length of the longest carry sequence while adding X_n and Y_n . Let the pmf of V_n be denoted by $p_n(v)$ and let $R_n(v) = \sum_{j=v}^n p_n(j)$. Define $R_n(v) = 0$ if $v > n$. Show that

$$R_n(v) = R_{n-1}(v) + \frac{1 - R_{n-v}(v)}{2^{v+1}}, \quad v \leq n.$$

Further show that $R_n(v) \leq \min\{1, (n - v + 1)/2^{v+1}\}$.

REFERENCES

- [BURK 1963] A. W. Burks, H. H. Goldstine, and J. von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” in A. H. Taub (ed.), *Collected Works of John von Neumann*, Vol. 5, Macmillan, New York, 1963.
- [DEME 1999] C. Demetrescu, “LLC-MAC analysis of general packet radio service in GSM,” *Bell Labs Technical J.*, **4** (3), 37–50 (July–Sept. 1999).
- [FOX 1988] B. L. Fox and P. W. Glynn, “Computing Poisson probabilities,” *Commun. ACM*, **31** (4), 440–445 (April 1988).
- [KNUT 1997] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*, 3rd ed., Addison-Wesley, Reading, MA, 1997.

- [LARS 1974] H. J. Larson, *Introduction to Probability Theory and Statistical Inference*, J Wiley, New York, 1974.
- [LEEM 1996] L. M. Leemis and K. S. Trivedi, “A comparison of approximate interval estimators for the Bernoulli parameter,” *Am. Statistician* **50** (1), 63–68 (Feb. 1996).
- [NBS 1950] National Bureau of Standards, *Tables of the Binomial Distribution*, U.S. Government Printing Office, Washington, DC, 1950.
- [PEAR 1966] E. S. Pearson and H. O. Hartley, *Biometrika Tables for Statisticians*, Cambridge Univ. Press, Cambridge, UK, 1966.
- [ROMI 1953] H. G. Romig, *50-100 Binomial Tables*, J Wiley, New York, 1953.
- [SCHA 1989] M. Schader and F. Schmid, “Two rules of thumb for the approximation of the binomial distribution by the normal distribution,” *Am. Statistician* **43**, 23–24 (1989).
- [SUN 1999] H.-R. Sun, Y. Cao, and K. S. Trivedi, “Availability and performance evaluation for automatic protection switching in TDMA wireless system,” *Pacific Rim Int. Symp. Dependable Computing*, 1999.
- [TOHM 1989] Y. Tohma, K. Tokunaga, S. Nagase, and Y. Murata, “Structural approach to the estimation of the number of residual software faults based on the hyper-geometric distribution,” *IEEE Trans. Software Eng.*, **15** (3), 345–355 (March 1989).
- [WETH 1980] C. J. Wetherell, “Probabilistic languages: A review and some open questions,” *ACM Comput. Surv.* **12** (4), 361–380 (Dec. 1980).
- [WOLF 1999] S. Wolfram, *The Mathematica Book*, 4th ed., Wolfram Media / Cambridge Univ. Press, 1999.

Chapter 3

Continuous Random Variables

3.1 INTRODUCTION

So far, we have considered discrete random variables and their distributions. In applications, such random variables denote the number of objects of a certain type, such as the number of job arrivals to a file server in one minute or the number of calls into a telephone exchange in one minute.

Many situations, both applied and theoretical, require the use of random variables that are “continuous” rather than discrete. As described in the last chapter, a random variable is a real-valued function on the sample space S . When the sample space S is nondenumerable (as mentioned in Section 1.7), not every subset of the sample space is an event that can be assigned a probability. As before, let \mathcal{F} denote the class of measurable subsets of S . Now if X is to be a random variable, it is natural to require that $P(X \leq x)$ be well defined for every real number x . In other words, if X is to be a random variable defined on a probability space (S, \mathcal{F}, P) , we require that $\{s | X(s) \leq x\}$ be an event (i.e., a member of \mathcal{F}). We are, therefore, led to the following extension of our earlier definition.

Definition (Random Variable). A random variable X on a probability space (S, \mathcal{F}, P) is a function $X : S \rightarrow \mathbb{R}$ that assigns a real number $X(s)$ to each sample point $s \in S$, such that for every real number x , the set of sample points $\{s | X(s) \leq x\}$ is an event, that is, a member of \mathcal{F} .

Definition (Distribution Function). The (cumulative) distribution function or CDF F_X of a random variable X is defined to be the function

$$F_X(x) = P(X \leq x), \quad -\infty < x < \infty.$$

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

The subscript X is used here to indicate the random variable under consideration. When there is no ambiguity, the subscript will be dropped and $F_X(x)$ will be denoted by $F(x)$.

As we saw in Chapter 2, the distribution function of a discrete random variable grows only by jumps. By contrast, the distribution function of a continuous random variable has no jumps but grows continuously. Thus, a **continuous random variable** X is characterized by a distribution function $F_X(x)$ that is a continuous function of x for all $-\infty < x < \infty$. Most continuous random variables that we encounter will have an absolutely continuous distribution function, $F(x)$, that is, one for which the derivative, $dF(x)/dx$, exists everywhere (except perhaps at a finite number of points). Such a random variable is called **absolutely continuous**. Thus, for instance, the continuous uniform distribution, given by

$$F(x) = \begin{cases} 0, & x < 0, \\ x, & 0 \leq x < 1, \\ 1, & x \geq 1, \end{cases}$$

possesses a derivative at all points except at $x = 0$ and $x = 1$. Therefore, it is an absolutely continuous distribution. All continuous random variables that we will study are absolutely continuous and hence the adjective will be dropped.

Definition (Probability Density Function). For a continuous random variable, X , $f(x) = dF(x)/dx$ is called the **probability density function** (pdf or density function) of X .

The pdf enables us to obtain the CDF by integrating the pdf:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt, \quad -\infty < x < \infty.$$

The analogy with (2.2) is clear, with the sum being replaced by an integral. We can also obtain other probabilities of interest such as

$$\begin{aligned} P(X \in (a, b]) &= P(a < X \leq b) \\ &= P(X \leq b) - P(X \leq a) \\ &= \int_{-\infty}^b f_X(t) dt - \int_{-\infty}^a f_X(t) dt \\ &= \int_a^b f_X(t) dt. \end{aligned}$$

The pdf, $f(x)$, satisfies the following properties:

(f1) $f(x) \geq 0$ for all x .

(f2) $\int_{-\infty}^{\infty} f(x) dx = 1$.

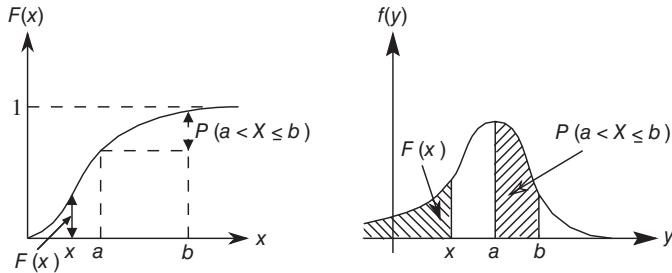


Figure 3.1. Relation between CDF and pdf

It should be noted that, unlike the pmf, the values of the pdf are not probabilities, and thus it is perfectly acceptable if $f(x) > 1$ at a point x .

As is the case for the CDF of a discrete random variable, the CDF of a continuous random variable, $F(x)$, satisfies the following properties:

$$\text{(F1)} \quad 0 \leq F(x) \leq 1, \quad -\infty < x < \infty.$$

$$\text{(F2)} \quad F(x) \text{ is a monotone increasing function of } x.$$

$$\text{(F3)} \quad \lim_{x \rightarrow -\infty} F(x) = 0 \text{ and } \lim_{x \rightarrow +\infty} F(x) = 1.$$

Unlike the CDF of a discrete random variable, the CDF of a continuous random variable does not have any jumps. Therefore, the probability associated with the event $[X = c] = \{s | X(s) = c\}$ is zero:

$$\text{(F4')} \quad P(X = c) = P(c \leq X \leq c) = \int_c^c f_X(y) dy = 0.$$

This does not imply that the set $\{s | X(s) = c\}$ is empty, but that the probability assigned to this set is zero. As a consequence of the fact that $P(X = c) = 0$, we have

$$\begin{aligned} P(a \leq X \leq b) &= P(a < X \leq b) = P(a \leq X < b) \\ &= P(a < X < b) \\ &= \int_a^b f_X(x) dx \\ &= F_X(b) - F_X(a). \end{aligned} \tag{3.1}$$

The relation between the functions f and F is illustrated in Figure 3.1. Probabilities are represented by areas under the pdf curve. The total area under the curve is unity.

Example 3.1

The time (measured in years), X , required to complete a software project has a pdf of the form:

$$f_X(x) = \begin{cases} kx(1-x), & 0 \leq x \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Since f_X satisfies property (f1), we know $k \geq 0$. In order for f_X to be a pdf, it must also satisfy property (f2); hence

$$\int_0^1 kx(1-x) dx = k \left(\frac{x^2}{2} - \frac{x^3}{3} \right) \Big|_0^1 = 1.$$

Therefore

$$k = 6.$$

Now the probability that the project will be completed in less than four months is given by

$$P(X < \frac{4}{12}) = F_X(\frac{1}{3}) = \int_0^{1/3} f_X(x) dx = \frac{7}{27}$$

or about 26 percent chance.

#

Most random variables we consider will either be discrete (as in Chapter 2) or continuous, but *mixed* random variables do occur sometimes. For example, there may be a nonzero probability, p_0 , of initial failure of a component at time 0 due to manufacturing defects. In this case, the time to failure, X , of the component is neither discrete nor a continuous random variable. The CDF of such a modified exponential random variable X with a mass at origin (shown in Figure 3.2) is then

$$F_X(x) = \begin{cases} 0, & x < 0, \\ p_0, & x = 0, \\ p_0 + (1 - p_0)(1 - e^{-\lambda x}), & x > 0. \end{cases} \quad (3.2)$$

The CDF of a mixed random variable satisfies properties (F1)–(F3) but it does not satisfy property (F4) of Chapter 2 or the property (F4') above.

The distribution function of a mixed random variable can be written as a linear combination of two distribution functions, denoted by $F^{(d)}(.)$ and $F^{(c)}(.)$, which are discrete and continuous, respectively, so that for every real number x

$$F_X(x) = \alpha_d F^{(d)}(x) + \alpha_c F^{(c)}(x)$$

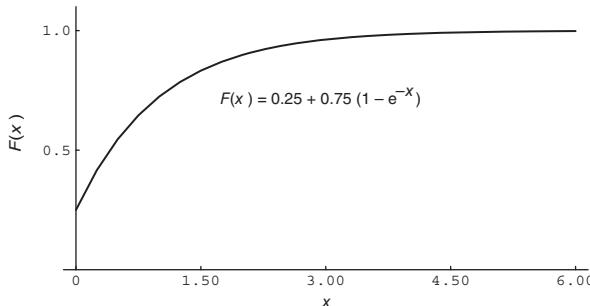


Figure 3.2. CDF of a mixed random variable

where $0 \leq \alpha_d$, $\alpha_c \leq 1$ and $\alpha_d + \alpha_c = 1$. Thus the mixed distribution (3.2) can be represented in this way if we let $F^{(d)}(x)$ as the unit step function, $F^{(c)}(x) = 1 - e^{-\lambda x}$, $\alpha_d = p_0$, and $\alpha_c = 1 - p_0$. (A unified treatment of discrete, continuous, and mixed random variables can also be given through the use of Riemann–Stieltjes integrals [BREI 1968, RUDI 1964].)

Problems

- Find the value of the constant k so that

$$f(x) = \begin{cases} kx^2(1-x^3), & 0 < x < 1 \\ 0, & \text{otherwise} \end{cases}$$

is a proper density function of a continuous random variable.

- Let X be a continuous random variable denoting the time to failure of a component. Suppose the distribution function of X is $F(x)$. Use this distribution function to express the probability of the following events:
 - $9 < X < 90$.
 - $X < 90$.
 - $X > 90$, given that $X > 9$.
- Consider a random variable X defined by the CDF:

$$F_X(x) = \begin{cases} 0, & x < 0, \\ \frac{1}{2}\sqrt{x} + \frac{1}{2}(1 - e^{-\sqrt{x}}), & 0 \leq x \leq 1, \\ \frac{1}{2} + \frac{1}{2}(1 - e^{-\sqrt{x}}), & x > 1. \end{cases}$$

Show that this function satisfies properties (F1)–(F3) and (F4'). Note that $F_X(x)$ is a continuous function but it does not have a derivative at $x = 1$. (That is, the pdf of X has a discontinuity at $x = 1$.) Plot the CDF and the pdf of X .

- See Hamming [HAMM 1973]. Consider a normalized floating-point number in base (or radix) β so that the mantissa, X , satisfies the condition $1/\beta \leq X < 1$. Experience shows that X has the following **reciprocal density**:

$$f_X(x) = \frac{k}{x}, \quad k > 0.$$

Determine

- The value of k .
- The distribution function of X .
- The probability that the leading digit of X is i for $1 \leq i < \beta$.

3.2 THE EXPONENTIAL DISTRIBUTION

This distribution, sometimes called the **negative exponential distribution**, occurs in applications such as reliability theory and queuing theory. Reasons for its use include its memoryless property (and resulting analytical

tractability) and its relation to the (discrete) Poisson distribution. Thus the following random variables will often be modeled as exponential:

1. Time between two successive job arrivals to a file server (often called **interarrival time**).
2. Service time at a server in a queuing network; the server could be a resource such as a CPU, an I/O device, or a communication channel.
3. Time to failure (lifetime) of a component.
4. Time required to repair a component that has malfunctioned.

Note that the assertion “Above distributions are exponential” is not a given fact but an assumption. Experimental verification of this assumption must be sought before relying on the results of the analysis (see Chapter 10 for further elaboration on this topic).

The **exponential distribution function**, shown in Figure 3.3, is given by

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & \text{if } 0 \leq x < \infty, \\ 0, & \text{otherwise.} \end{cases} \quad (3.3)$$

If a random variable X possesses CDF given by equation (3.3), we use the notation $X \sim EXP(\lambda)$, for brevity. The pdf of X has the shape shown in Figure 3.4 and is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.4)$$

While specifying a pdf, usually we state only the nonzero part, and it is understood that the pdf is zero over any unspecified region. Since $\lim_{x \rightarrow \infty} F(x) = 1$, it follows that the total area under the exponential pdf is unity. Also

$$\begin{aligned} P(X \geq t) &= \int_t^{\infty} f(x) \, dx \\ &= e^{-\lambda t} \end{aligned} \quad (3.5)$$

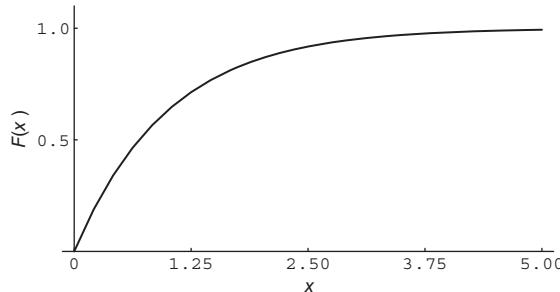


Figure 3.3. The CDF of an exponentially distributed random variable ($\lambda = 1$)

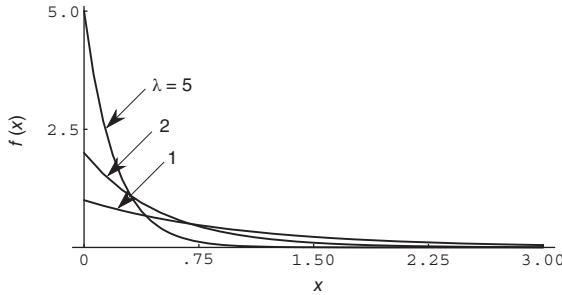


Figure 3.4. Exponential pdf

and

$$\begin{aligned} P(a \leq X \leq b) &= F(b) - F(a) \\ &= e^{-\lambda a} - e^{-\lambda b}. \end{aligned}$$

Now let us investigate the **memoryless property** of the exponential distribution. Suppose we know that X exceeds some given value t ; that is, $X > t$. For example, let X be the lifetime of a component, and suppose we have observed that this component has already been operating for t hours. We may then be interested in the distribution of $Y = X - t$, the remaining (residual) lifetime. Let the conditional probability of $Y \leq y$, given that $X > t$, be denoted by $G_Y(y|t)$. Thus, for $y \geq 0$, we have

$$\begin{aligned} G_Y(y|t) &= P(Y \leq y | X > t) \\ &= P(X - t \leq y | X > t) \\ &= P(X \leq y + t | X > t) \\ &= \frac{P(X \leq y + t \text{ and } X > t)}{P(X > t)} \\ &\quad (\text{by the definition of conditional probability}) \\ &= \frac{P(t < X \leq y + t)}{P(X > t)}. \end{aligned}$$

Thus (see Figure 3.5)

$$\begin{aligned} G_Y(y|t) &= \frac{\int_t^{y+t} f(x) dx}{\int_t^\infty f(x) dx} \\ &= \frac{\int_t^{y+t} \lambda e^{-\lambda x} dx}{\int_t^\infty \lambda e^{-\lambda x} dx} \\ &= \frac{e^{-\lambda t} (1 - e^{-\lambda y})}{e^{-\lambda t}} \\ &= 1 - e^{-\lambda y}. \end{aligned}$$

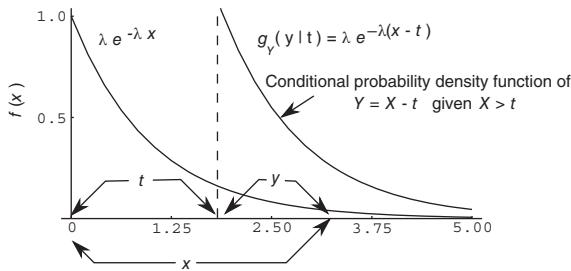


Figure 3.5. Memoryless property of the exponential distribution ($\lambda = 1$)

Thus, $G_Y(y|t)$ is independent of t and is identical to the original exponential distribution of X . The distribution of the remaining life does not depend on how long the component has been operating. The component does not “age” (it is as good as new or it “forgets” how long it has been operating), and its eventual breakdown is the result of some suddenly appearing failure, not of gradual deterioration.

If the interarrival times are exponentially distributed, then the memoryless property implies that the time we must wait for a new arrival is statistically independent of how long we have already spent waiting for it.

If X is a nonnegative continuous random variable with the memoryless property, then we can show that the distribution of X must be exponential:

$$\frac{P(t < X \leq y + t)}{P(X > t)} = P(X \leq y) = P(0 < X \leq y),$$

or

$$F_X(y + t) - F_X(t) = [1 - F_X(t)][F_X(y) - F_X(0)].$$

Since $F_X(0) = 0$, we rearrange this equation to get

$$\frac{F_X(y + t) - F_X(y)}{t} = \frac{F_X(t)[1 - F_X(y)]}{t}.$$

Taking the limit as t approaches zero, we get

$$F'_X(y) = F'_X(0)[1 - F_X(y)],$$

where F'_X denotes the derivative of F_X . Let $R_X(y) = 1 - F_X(y)$; then the preceding equation reduces to

$$R'_X(y) = R'_X(0)R_X(y).$$

The solution to this differential equation is given by

$$R_X(y) = K e^{R'_X(0)y},$$

where K is a constant of integration and $-R'_X(0) = F'_X(0) = f_X(0)$, the pdf evaluated at 0. Noting that $R_X(0) = 1$, and denoting $f_X(0)$ by the constant λ , we get

$$R_X(y) = e^{-\lambda y}$$

and hence

$$F_X(y) = 1 - e^{-\lambda y}, \quad y > 0.$$

Therefore X must have the exponential distribution.

The exponential distribution can be obtained from the Poisson distribution by considering the interarrival times rather than the number of arrivals.

Example 3.2

Let the discrete random variable N_t denote the number of jobs arriving to a file server in the interval $(0, t]$. Let X be the time of the next arrival. Further assume that N_t is Poisson distributed with parameter λt , so that λ is the arrival rate. Then

$$\begin{aligned} P(X > t) &= P(N_t = 0) \\ &= \frac{e^{-\lambda t} (\lambda t)^0}{0!} \\ &= e^{-\lambda t} \end{aligned}$$

and

$$F_X(t) = 1 - e^{-\lambda t}.$$

Therefore, the time to the next arrival is exponentially distributed. More generally, it can be shown that the interarrival times of Poisson events are exponentially distributed [BHAT 1984, p. 197].

#

Example 3.3

Consider a Web server with an average rate of requests $\lambda = 0.1$ jobs per second. Assuming that the number of arrivals per unit time is Poisson distributed, the interarrival time, X , is exponentially distributed with parameter λ . The probability that an interval of 10 seconds elapses without requests is then given by

$$\begin{aligned} P(X \geq 10) &= \int_{10}^{\infty} 0.1e^{-0.1t} dt = \lim_{t \rightarrow \infty} [-e^{-0.1t}] - (-e^{-1}) \\ &= e^{-1} = 0.368. \end{aligned}$$

#

Problems

- Jobs arriving to a compute server have been found to require CPU time that can be modeled by an exponential distribution with parameter $1/140 \text{ ms}^{-1}$. The CPU scheduling discipline is quantum-oriented so that a job not completing within a quantum of 100 ms will be routed back to the tail of the queue of waiting jobs. Find the probability that an arriving job is forced to wait for a second quantum. Of the 800 jobs coming in during a day, how many are expected to finish within the first quantum?

3.3 THE RELIABILITY AND FAILURE RATE

Let the random variable X be the lifetime or the time to failure of a component. The probability that the component survives until some time t is called the **reliability** $R(t)$ of the component. Thus, $R(t) = P(X > t) = 1 - F(t)$, where F is the distribution function of the component lifetime X . The component is normally (but not always) assumed to be working properly at time $t = 0$ [i.e., $R(0) = 1$], and no component can work forever without failure [i.e., $\lim_{t \rightarrow +\infty} R(t) = 0$]. Also, $R(t)$ is a monotone decreasing function of t . For t less than zero, reliability has no meaning, but we let $R(t) = 1$ for $t < 0$. $F(t)$ will often be called the **unreliability**.

Consider a fixed number of identical components, N_0 , under test. After time t , $N_f(t)$ components have failed and $N_s(t)$ components have survived with $N_f(t) + N_s(t) = N_0$. The estimated probability of survival may be written (using the frequency interpretation of probability) as

$$\hat{P}(\text{survival}) = \frac{N_s(t)}{N_0}.$$

In the limit as $N_0 \rightarrow \infty$, we expect $\hat{P}(\text{survival})$ to approach $R(t)$. As the test progresses, $N_s(t)$ gets smaller and $R(t)$ decreases:

$$\begin{aligned} R(t) &\simeq \frac{N_s(t)}{N_0} \\ &= \frac{N_0 - N_f(t)}{N_0} \\ &= 1 - \frac{N_f(t)}{N_0}. \end{aligned}$$

The total number of components N_0 is constant, while the number of failed components N_f increases with time. Taking derivatives on both sides of the preceding equation, we get

$$R'(t) \simeq -\frac{1}{N_0} N'_f(t). \quad (3.6)$$

In this equation, $N'_f(t)$ is the rate at which components fail. Therefore, as $N_0 \rightarrow \infty$, the right-hand side may be interpreted as the negative of the failure density function, $f_X(t)$:

$$R'(t) = -f_X(t). \quad (3.7)$$

Note that $f(t)\Delta t$ is the (unconditional) probability that a component will fail in the interval $(t, t + \Delta t]$. However, if we have observed the component functioning up to some time t , we expect the (conditional) probability of its failure to be different from $f(t)\Delta t$. This leads us to the notion of instantaneous failure rate as follows.

Notice that the conditional probability that the component does not survive for an (additional) interval of duration x given that it has survived until time t can be written as

$$G_Y(x|t) = \frac{P(t < X \leq t + x)}{P(X > t)} = \frac{F(t + x) - F(t)}{R(t)}. \quad (3.8)$$

Definition (Instantaneous Failure Rate). The instantaneous failure rate $h(t)$ at time t is defined to be

$$h(t) = \lim_{x \rightarrow 0} \frac{1}{x} \frac{F(t + x) - F(t)}{R(t)} = \lim_{x \rightarrow 0} \frac{R(t) - R(t + x)}{xR(t)},$$

so that

$$h(t) = \frac{f(t)}{R(t)}. \quad (3.9)$$

Thus, $h(t)\Delta t$ represents the conditional probability that a component having survived to age t will fail in the interval $(t, t + \Delta t]$. Alternate terms for $h(t)$ are *hazard rate*, *force of mortality*, *intensity rate*, *conditional failure rate*, or simply *failure rate*. Failure rates in practice are so small that expressing them as failures per hour is not appropriate. Often, $h(t)$ is expressed in failures per 10,000 hours. Another commonly used unit is FIT (failures in time), which expresses failures per 10^9 or a billion hours.

It should be noted that the exponential distribution is characterized by a constant failure rate, since

$$h(t) = \frac{f(t)}{R(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda.$$

By integrating both sides of equation (3.9), we get

$$\begin{aligned} \int_0^t h(x) dx &= \int_0^t \frac{f(x)}{R(x)} dx \\ &= \int_0^t -\frac{R'(x)}{R(x)} dx \quad \text{using equation (3.7)} \\ &= - \int_{R(0)}^{R(t)} \frac{dR}{R}, \end{aligned}$$

or

$$\int_0^t h(x) dx = -\ln R(t)$$

using the boundary condition, $R(0) = 1$. Therefore

$$R(t) = \exp \left[- \int_0^t h(x) dx \right]. \quad (3.10)$$

This formula holds even when the distribution of the time to failure is not exponential.

The cumulative failure rate, $H(t) = \int_0^t h(x)dx$, is referred to as the **cumulative hazard**. Equation (3.10) gives a useful theoretical representation of reliability as a function of the failure rate. An alternate representation gives the reliability in terms of cumulative hazard:

$$R(t) = e^{-H(t)}. \quad (3.11)$$

Note that if the lifetime is exponentially distributed, then $H(t) = \lambda t$, and we obtain the exponential reliability function.

We should note the difference between $f(t)$ and $h(t)$. The quantity $f(t)\Delta t$ is the unconditional probability that the component will fail in the interval $(t, t + \Delta t]$, whereas $h(t)\Delta t$ is the conditional probability that the component will fail in the same time interval, given that it has survived until time t . Also, $h(t) = f(t)/R(t)$ is always greater than or equal to $f(t)$, because $R(t) \leq 1$. Function $f(t)$ represents probability density whereas $h(t)$ does not. By analogy, the probability that a newborn child will die at an age between 99 and 100 years [corresponding to $f(t)\Delta t$] is quite small because few of them will survive that long. But the probability of dying in that same period, provided that the child has survived until age 99 (corresponding to $h(t)\Delta t$) is much greater.

To further see the difference between the failure rate $h(t)$ and failure density $f(t)$, we need the notion of conditional probability density. Let $V_X(x|t)$ denote the conditional distribution of the lifetime X given that the component has survived past fixed time t . Then

$$\begin{aligned} V_X(x|t) &= \frac{\int_t^x f(y)dy}{P(X > t)} \\ &= \begin{cases} \frac{F(x) - F(t)}{1 - F(t)}, & x \geq t, \\ 0, & x < t. \end{cases} \end{aligned}$$

[Note that $V_X(x|t) = G_Y(x-t|t)$.]

Then the conditional failure density is

$$v_X(x|t) = \begin{cases} \frac{f(x)}{1-F(t)}, & x \geq t, \\ 0, & x < t. \end{cases}$$

The conditional density $v_X(x|t)$ satisfies properties (f1) and (f2) and hence is a probability density while the failure rate $h(t)$ does not satisfy property (f2) since

$$0 = \lim_{t \rightarrow \infty} R(t) = \exp \left[- \int_0^\infty h(t)dt \right].$$

[Note that $h(t) = v_X(t|t)$.]

Define the **conditional reliability** $R_t(y)$ to be the probability that the component survives an (additional) interval of duration y given that it has survived until time t . Thus

$$R_t(y) = \frac{R(t+y)}{R(t)}. \quad (3.12)$$

[Note that $R_t(y) = 1 - G_Y(y|t)$.]

Now consider a component that does not age stochastically. In other words, its survival probability over an additional period of length y is the same regardless of its present age:

$$R_t(y) = R(y) \quad \text{for all } y, t \geq 0.$$

Then, using formula (3.12), we get

$$R(y+t) = R(t) R(y), \quad (3.13)$$

and rearranging, we get

$$\frac{R(y+t) - R(y)}{t} = \frac{[R(t) - 1]R(y)}{t}.$$

Taking the limit as t approaches zero and noting that $R(0) = 1$, we obtain

$$R'(y) = R'(0)R(y).$$

So $R(y) = e^{yR'(0)}$. Letting $R'(0) = -\lambda$, we get

$$R(y) = e^{-\lambda y}, \quad y > 0,$$

which implies that the lifetime $X \sim EXP(\lambda)$. In this case, the failure rate $h(t)$ is equal to λ , which is a constant, independent of component age t . Conversely, the exponential lifetime distribution is the only distribution with a constant failure rate [BARL 1981]. If a component has exponential lifetime distribution, then it follows that

1. Since a used component is (stochastically) as good as new, a policy of a scheduled replacement of used components (known to be still functioning) does not accrue any benefit.
2. In estimating mean life, reliability, and other such quantities, data may be collected consisting only of the number of hours of observed life and of the number of observed failures; the ages of components under observation are of no concern.

Now consider a component that ages adversely in the sense that the conditional survival probability decreases with the age t ; that is, $R_t(y)$ is decreasing in $0 < t < \infty$ for all $y \geq 0$. As a result

$$h(t) = \lim_{y \rightarrow 0} \frac{R(t) - R(t+y)}{yR(t)}$$

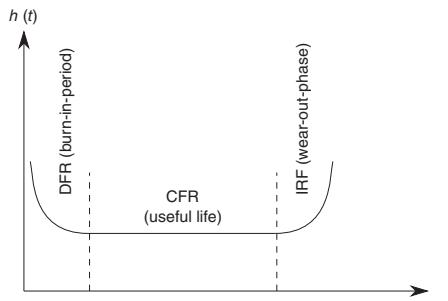


Figure 3.6. Failure rate as a function of time

is an increasing function of t for $t \geq 0$. The corresponding distribution function $F(t)$ is known as an **increasing failure rate (IFR) distribution**.

Alternately, if aging is beneficial in the sense that the conditional survival probability increases with age, then the failure rate will be a decreasing function of age, and the corresponding distribution is known as a **decreasing failure rate (DFR) distribution**.

The behavior of the failure rate as a function of age is known as the *mortality curve*, *hazard function*, *life characteristic*, or *lambda characteristic*. The mortality curve is empirically observed to have the so-called bathtub shape shown in Figure 3.6. During the early life period (infant mortality phase, burnin period, debugging period, or breakin period), failures are of the **endogenous** type and arise from inherent defects in the system attributed to faulty design, manufacturing, or assembly. During this period, the failure rate is expected to drop with age.

When the system has been debugged, it is prone to chance or random failure (also called **exogenous** failure). Such failures are usually associated with environmental conditions under which the component is operating. They are the results of severe, unpredictable stresses arising from sudden environmental shocks; the failure rate is determined by the severity of the environment. During this useful-life phase, failure rate is approximately constant and the exponential model is usually acceptable.

The rationale for the choice of exponential failure law is provided by assuming that the component is operating in an environment that subjects it to a stress varying in time. A failure occurs when the applied stress exceeds the maximum allowable stress, S_{\max} (see Figure 3.7). Such “peak” stresses may be assumed to follow a Poisson distribution with parameter λt , where λ is a constant rate of occurrence of peak loads. Denoting the number of peak stresses in the interval $(0, t]$ by N_t , we get

$$P(N_t = r) = \frac{e^{-\lambda t} (\lambda t)^r}{r!}, \quad \lambda > 0, \quad r = 0, 1, 2, \dots$$

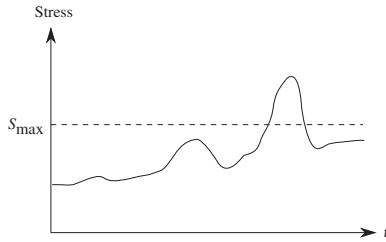


Figure 3.7. Stress as a function of time

Now the event $[X > t]$, where X is the component lifetime, corresponds to the event $[N_t = 0]$, and thus

$$\begin{aligned} R(t) &= P(X > t) \\ &= P(N_t = 0) \\ &= e^{-\lambda t}, \end{aligned}$$

the exponential reliability function.

When components begin to reach their “rated life,” the system failure rate begins to increase and it is said to have entered the wearout phase. The wearout failure is the outcome of accumulated wear and tear, a result of a depletion process due to abrasion, fatigue, creep, and the like.

Problems

- The failure rate of a certain component is $h(t) = \lambda_0 t$, where $\lambda_0 > 0$ is a given constant. Determine the reliability, $R(t)$, of the component. Repeat for $h(t) = \lambda_0 t^{1/2}$.
- The failure rate of a computer system for onboard control of a space vehicle is estimated to be the following function of time:

$$h(t) = \alpha \mu t^{\alpha-1} + \beta \gamma t^{\beta-1}.$$

Derive an expression for the reliability $R(t)$ of the system. Plot $h(t)$ and $R(t)$ as functions of time with parameter values $\alpha = \frac{1}{4}$, $\beta = \frac{1}{7}$, $\mu = 0.0004$, and $\gamma = 0.0007$.

3.4 SOME IMPORTANT DISTRIBUTIONS

3.4.1 Hypoexponential Distribution

Many processes in nature can be divided into sequential phases. If the time the process spends in each phase is independent and exponentially distributed,

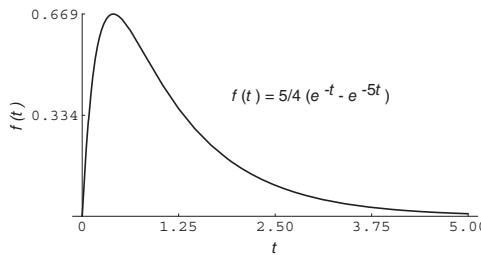


Figure 3.8. The pdf of the hypoexponential distribution

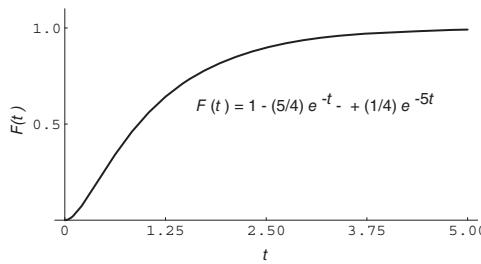


Figure 3.9. The CDF of the hypoexponential distribution

then it can be shown that the overall time is hypoexponentially distributed. It has been empirically observed that the service times for input–output operations in a computer system often possess this distribution. The distribution has r parameters, one for each of its distinct phases. A two-stage hypoexponential random variable, X , with parameters λ_1 and λ_2 ($\lambda_1 \neq \lambda_2$), will be denoted by $X \sim \text{HYPO}(\lambda_1, \lambda_2)$, and its pdf is given by (see Figure 3.8)

$$f(t) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t}), \quad t > 0. \quad (3.14)$$

The corresponding distribution function is (see Figure 3.9)

$$F(t) = 1 - \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 t}, \quad t \geq 0. \quad (3.15)$$

The hazard rate of this distribution is given by

$$h(t) = \frac{\lambda_1 \lambda_2 (e^{-\lambda_1 t} - e^{-\lambda_2 t})}{\lambda_2 e^{-\lambda_1 t} - \lambda_1 e^{-\lambda_2 t}}. \quad (3.16)$$

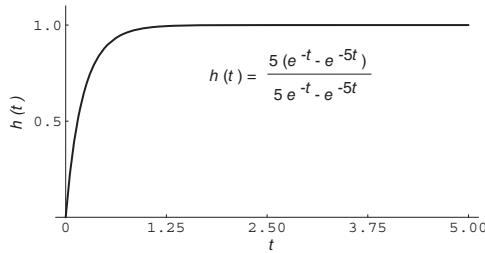


Figure 3.10. The failure rate of the hypoexponential distribution

It is not difficult to see that this is an IFR distribution with the failure rate increasing from 0 up to $\min\{\lambda_1, \lambda_2\}$ (see Figure 3.10).

3.4.2 Erlang and Gamma Distribution

When r sequential phases have identical exponential distributions, then the resulting density is known as r -stage (or r -phase) Erlang and is given by

$$f(t) = \frac{\lambda^r t^{r-1} e^{-\lambda t}}{(r-1)!}, \quad t > 0, \quad \lambda > 0, \quad r = 1, 2, \dots \quad (3.17)$$

The distribution function is

$$F(t) = 1 - \sum_{k=0}^{r-1} \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad t \geq 0, \quad \lambda > 0, \quad r = 1, 2, \dots \quad (3.18)$$

Also

$$h(t) = \frac{\lambda^r t^{r-1}}{(r-1)! \sum_{k=0}^{r-1} \frac{(\lambda t)^k}{k!}}, \quad t > 0, \quad \lambda > 0, \quad r = 1, 2, \dots \quad (3.19)$$

The exponential distribution is a special case of the Erlang distribution with $r = 1$. The physical interpretation of this distribution is a natural extension of that for the exponential. Consider a component subjected to an environment so that N_t , the number of peak stresses in the interval $(0, t]$, is Poisson distributed with parameter λt . Suppose further that the component can withstand $(r-1)$ peak stresses and the r th occurrence of a peak stress causes a failure. Then the component lifetime X is related to N_t so that the following two events are equivalent:

$$[X > t] = [N_t < r]$$

thus

$$\begin{aligned}
 R(t) &= P(X > t) \\
 &= P(N_t < r) \\
 &= \sum_{k=0}^{r-1} P(N_t = k) \\
 &= e^{-\lambda t} \sum_{k=0}^{r-1} \frac{(\lambda t)^k}{k!}.
 \end{aligned}$$

Then $F(t) = 1 - R(t)$ yields formula (3.18). We conclude that the component lifetime has an r -stage Erlang distribution.

If we let r (call it α) take nonintegral values, then we get the gamma density

$$f(t) = \frac{\lambda^\alpha t^{\alpha-1} e^{-\lambda t}}{\Gamma(\alpha)}, \quad \alpha > 0, t > 0. \quad (3.20)$$

where the **gamma function** is defined by the following integral

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx, \quad \alpha > 0. \quad (3.21)$$

The following properties of the gamma function will be useful in the sequel. Integration by parts shows that for $\alpha > 1$

$$\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1). \quad (3.22)$$

In particular, if α is a positive integer, denoted by n , then

$$\Gamma(n) = (n - 1)! \quad (3.23)$$

Other useful formulas related to the gamma function are

$$\Gamma(\frac{1}{2}) = \sqrt{\pi} \quad (3.24)$$

and

$$\int_0^\infty x^{\alpha-1} e^{-\lambda x} dx = \frac{\Gamma(\alpha)}{\lambda^\alpha}. \quad (3.25)$$

A random variable X with pdf (3.20) will be denoted by $X \sim \text{GAM}(\lambda, \alpha)$. This distribution has two parameters. The parameter α is called a **shape parameter** since, as α increases, the density becomes more peaked. The parameter λ is a **scale parameter**; that is, the distribution depends on λ only through the product λt . The gamma distribution is DFR for $0 < \alpha < 1$ and IFR for $\alpha > 1$ (see Figure 3.11). For $\alpha = 1$, the distribution degenerates to the exponential distribution; that is, $EXP(\lambda) = \text{GAM}(\lambda, 1)$.

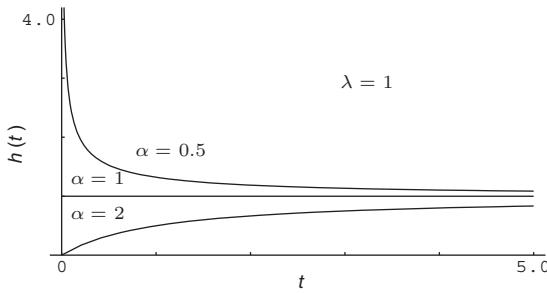


Figure 3.11. The failure rate of the gamma distribution

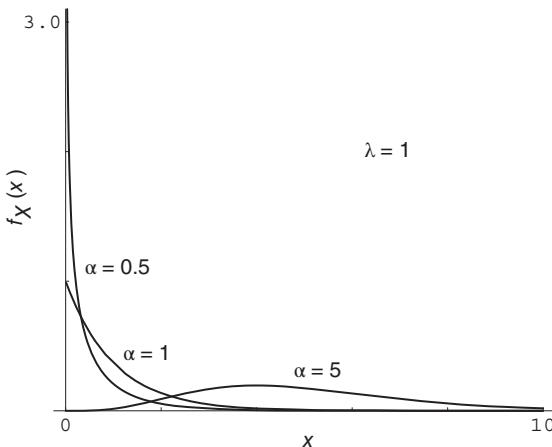


Figure 3.12. The gamma pdf

The gamma distribution is the continuous counterpart of the (discrete) negative binomial distribution. The chi-square distribution, useful in mathematical statistics, is a special case of the gamma distribution with $\alpha = n/2$ (n is a positive integer) and $\lambda = \frac{1}{2}$. Thus, if $X \sim \text{GAM}(\frac{1}{2}, n/2)$, then it is said to have a chi-square distribution with n degrees of freedom. Figure 3.12 illustrates possible shapes of gamma density with $\alpha = 0.5, 1$, and 5 .

In Chapter 4 we will show that if a sequence of k random variables X_1, X_2, \dots, X_k are mutually independent and identically distributed as $\text{GAM}(\lambda, \alpha)$, then their sum $\sum_{i=1}^k X_i$ is distributed as $\text{GAM}(\lambda, k\alpha)$.

3.4.3 Hyperexponential Distribution

A process with sequential phases gives rise to a hypoexponential or an Erlang distribution, depending on whether the phases have identical distributions. Instead, if a process consists of alternate phases—that is, during any single experiment, the process experiences one and only one of the many alternate

phases—and these phases have exponential distributions, then the overall distribution is hyperexponential. The density function of a k -phase hyperexponential random variable is

$$f(t) = \sum_{i=1}^k \alpha_i \lambda_i e^{-\lambda_i t}, \quad t > 0, \lambda_i > 0, \alpha_i > 0, \sum_{i=1}^k \alpha_i = 1, \quad (3.26)$$

and the distribution function is

$$F(t) = \sum_i \alpha_i (1 - e^{-\lambda_i t}), \quad t \geq 0. \quad (3.27)$$

The failure rate is

$$h(t) = \frac{\sum \alpha_i \lambda_i e^{-\lambda_i t}}{\sum \alpha_i e^{-\lambda_i t}}, \quad t \geq 0, \quad (3.28)$$

which is a decreasing failure rate from $\sum \alpha_i \lambda_i$ down to $\min\{\lambda_1, \lambda_2, \dots\}$.

The hyperexponential distribution exhibits more variability than the exponential. Lee et al. have found that the time to failure distributions of VAXcluster and Tandem software are captured well by the two-phase hyperexponential [LEE 1993]. CPU service time in a computer system has often been observed to possess such a distribution (see Figure 3.13). Similarly, if a product is manufactured in several parallel assembly lines and the outputs are merged, the failure density of the overall product is likely to be hyperexponential. The hyperexponential is a special case of mixture distributions that often arise in practice—that is, of the form:

$$F(x) = \sum_i \alpha_i F_i(x), \quad \sum_i \alpha_i = 1, \quad \alpha_i \geq 0. \quad (3.29)$$

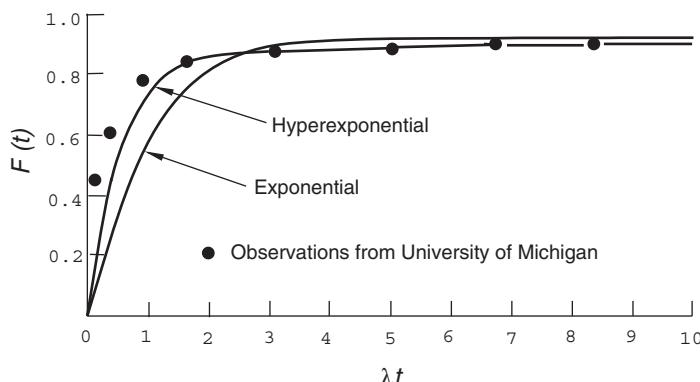


Figure 3.13. The CPU service time distribution compared with the hyperexponential distribution. (Reproduced from R. F. Rosin, “Determining a computing center environment,” *CACM*, 1965; reprinted with permission of the Association of Computing Machinery.)

3.4.4 Weibull Distribution

The Weibull distribution has been used to describe fatigue failure, electronic component failure, and ballbearing failure. At present, it is perhaps the most widely used parametric family of failure distributions. The reason is that by a proper choice of its shape parameter α , an IFR, a DFR, or a constant failure rate distribution can be obtained. Therefore, it can be used for all three phases of the mortality curve. The density is given by

$$f(t) = \lambda \alpha t^{\alpha-1} e^{-\lambda t^\alpha}, \quad (3.30)$$

the distribution function by

$$F(t) = 1 - e^{-\lambda t^\alpha}, \quad (3.31)$$

and the failure rate by

$$h(t) = \lambda \alpha t^{\alpha-1}, \quad (3.32)$$

and the cumulative hazard is a power function, $H(t) = \lambda t^\alpha$. For all these formulas, $t \geq 0$, $\lambda > 0$, $\alpha > 0$. Figure 3.14 shows $h(t)$ plotted as a function of t , for various values of α .

Often a third parameter is added to obtain a three-parameter Weibull distribution:

$$F(t) = 1 - e^{-\lambda(t-\theta)^\alpha}, \quad t \geq \theta \quad (3.33)$$

where θ is the location parameter.

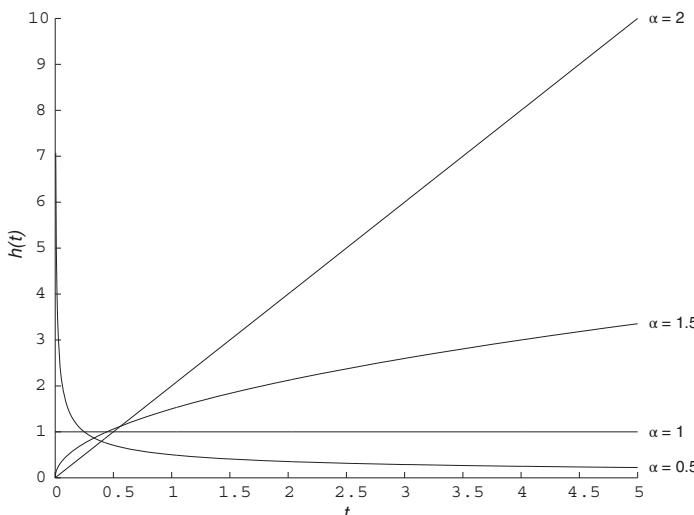


Figure 3.14. Failure rate of the Weibull distribution with various values of α and $\lambda = 1$

Example 3.4

The lifetime X in hours of a component is modeled by a Weibull distribution with shape parameter $\alpha = 2$. Starting with a large number of components, it is observed that 15% of the components that have lasted 90 h fail before 100 h. Determine the scale parameter λ .

Note that

$$F_X(x) = 1 - e^{-\lambda x^2}$$

and we are given that

$$P(X < 100 | X > 90) = 0.15.$$

Also

$$\begin{aligned} P(X < 100 | X > 90) &= \frac{P(90 < X < 100)}{P(X > 90)} \\ &= \frac{F_X(100) - F_X(90)}{1 - F_X(90)} \\ &= \frac{e^{-\lambda(90)^2} - e^{-\lambda(100)^2}}{e^{-\lambda(90)^2}}. \end{aligned}$$

Equating the two expressions and solving for λ , we get

$$\lambda = -\frac{\ln(0.85)}{1900} = \frac{0.1625}{1900} = 0.00008554.$$

#

3.4.5 Log-Logistic Distribution

Although Weibull and gamma distributions are widely used, they are limited in their modeling capability. As outlined earlier, they are appropriate for modeling constant, strictly increasing, and strictly decreasing failure rate.

The log-logistic distribution has scale parameter $\lambda > 0$ and shape parameter $\kappa > 0$. Depending on the choice of its shape parameter, a decreasing or increasing/decreasing behavior of failure rate can be obtained. The density function of a log-logistic random variable is

$$f(t) = \frac{\lambda \kappa (\lambda t)^{\kappa-1}}{[1 + (\lambda t)^\kappa]^2}, \quad t \geq 0, \quad (3.34)$$

the distribution function is

$$F(t) = 1 - \frac{1}{1 + (\lambda t)^\kappa}, \quad (3.35)$$

the hazard rate is

$$h(t) = \frac{\lambda \kappa (\lambda t)^{\kappa-1}}{1 + (\lambda t)^\kappa}, \quad (3.36)$$

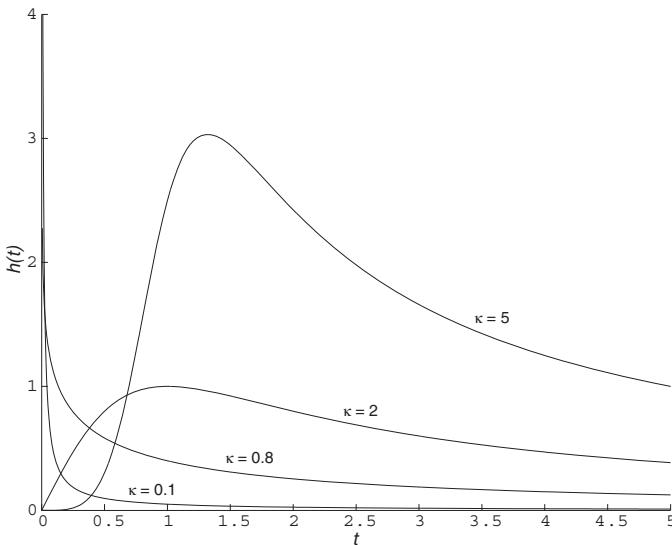


Figure 3.15. Failure rate of log-logistic distribution with $\lambda = 1$

and the cumulative hazard is $H(t) = \ln[1 + (\lambda t)^\kappa]$. When $\kappa \leq 1$ the log-logistic distribution is a DFR distribution. For $\kappa > 1$ the hazard rate initially increases and then decreases. The corresponding distribution function in this case is known as a UBT (upside-down bathtub) distribution (see Figure 3.15).

3.4.6 Normal or Gaussian Distribution

This distribution is extremely important in statistical applications because of the central-limit theorem, which states that, under very general assumptions, the mean of a sample of n mutually independent random variables (having distributions with finite mean and variance) is normally distributed in the limit $n \rightarrow \infty$. It has been observed that errors of measurement often possess this distribution. Experience also shows that during the wearout phase, component lifetime follows a normal distribution.

The normal density has the well-known bell-shaped curve (see Figure 3.16) and is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], \quad -\infty < x < \infty, \quad (3.37)$$

where $-\infty < \mu < \infty$ and $\sigma > 0$ are two parameters of the distribution. (We will see in Chapter 4 that these parameters are, respectively, the mean and the standard deviation of the distribution.) If a random variable X has the pdf (3.37), then we write $X \sim N(\mu, \sigma^2)$.

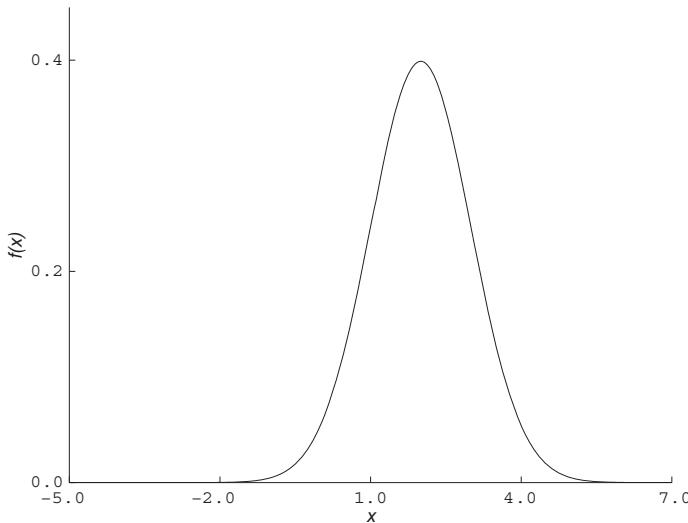


Figure 3.16. Normal density with parameters $\mu = 2$ and $\sigma = 1$

Since the distribution function $F(x)$ has no closed form, between every pair real numbers of limits a and b , probabilities relating to normal distributions are usually obtained numerically and recorded in special tables (see Appendix C). Such tables pertain to the **standard normal distribution** [$Z \sim N(0, 1)$]—a normal distribution with parameters $\mu = 0$ and $\sigma = 1$ —and their entries are the values of

$$F_Z(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt. \quad (3.38)$$

Since the standard normal density is clearly symmetric, it follows that for $z > 0$ we have

$$\begin{aligned} F_Z(-z) &= \int_{-\infty}^{-z} f_Z(t) dt \\ &= \int_z^{\infty} f_Z(-t) dt \\ &= \int_z^{\infty} f_Z(t) dt \\ &= \int_{-\infty}^{\infty} f_Z(t) dt - \int_{-\infty}^z f_Z(t) dt \\ &= 1 - F_Z(z). \end{aligned} \quad (3.39)$$

Therefore, the tabulations of the normal distribution are made only for $z \geq 0$. To find $P(a \leq Z \leq b)$, we use $F(b) - F(a)$.

For a particular value, x , of a normal random variable X , the corresponding value of the standardized variable Z is given by $z = (x - \mu)/\sigma$. The distribution function of X can now be found by using the following relation

$$\begin{aligned} F_Z(z) &= P(Z \leq z) \\ &= P\left(\frac{X - \mu}{\sigma} \leq z\right) \\ &= P(X \leq \mu + z\sigma) \\ &= F_X(\mu + z\sigma). \end{aligned}$$

Alternatively

$$F_X(x) = F_Z\left(\frac{x - \mu}{\sigma}\right). \quad (3.40)$$

Example 3.5

An analog signal received at a detector (measured in microvolts) may be modeled as a Gaussian random variable $N(200, 256)$ at a fixed point in time. What is the probability that the signal will exceed $240 \mu\text{V}$? What is the probability that the signal is larger than $240 \mu\text{V}$, given that it is greater than $210 \mu\text{V}$?

$$\begin{aligned} P(X > 240) &= 1 - P(X \leq 240) \\ &= 1 - F_Z\left(\frac{240 - 200}{16}\right), \quad \text{using equation (3.40)} \\ &= 1 - F_Z(2.5) \\ &\simeq 0.00621. \end{aligned}$$

Next

$$\begin{aligned} P(X > 240 | X > 210) &= \frac{P(X > 240)}{P(X > 210)} \\ &= \frac{1 - F_Z\left(\frac{240 - 200}{16}\right)}{1 - F_Z\left(\frac{210 - 200}{16}\right)} \\ &= \frac{0.00621}{0.26599} \\ &\simeq 0.02335. \end{aligned}$$

#

If X denotes the measured quantity in a certain experiment, then the probability of an event such as $\mu - k\sigma \leq X \leq \mu + k\sigma$ is an indicator of the measurement error

$$\begin{aligned}
 P(\mu - k\sigma \leq X \leq \mu + k\sigma) &= F_X(\mu + k\sigma) - F_X(\mu - k\sigma) \\
 &= F_Z(k) - F_Z(-k) \\
 &= 2F_Z(k) - 1 \\
 &= \frac{2}{\sqrt{2\pi}} \int_{-\infty}^k e^{-t^2/2} dt - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-t^2/2} dt \\
 &= \frac{2}{\sqrt{2\pi}} \int_{-\infty}^k e^{-t^2/2} dt - \frac{2}{\sqrt{2\pi}} \int_{-\infty}^0 e^{-t^2/2} dt \\
 &= \frac{2}{\sqrt{2\pi}} \int_0^k e^{-t^2/2} dt.
 \end{aligned}$$

By the variable transformation, $t = \sqrt{2y}$, we get

$$P(\mu - k\sigma \leq X \leq \mu + k\sigma) = \frac{2}{\sqrt{\pi}} \int_0^{k/\sqrt{2}} e^{-y^2} dy.$$

The **error function** (or error integral) is defined by

$$\operatorname{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-y^2} dy. \quad (3.41)$$

Thus

$$P(\mu - k\sigma \leq X \leq \mu + k\sigma) = \operatorname{erf}\left(\frac{k}{\sqrt{2}}\right). \quad (3.42)$$

For example, for $k = 3$, we obtain

$$P(\mu - 3\sigma \leq X \leq \mu + 3\sigma) = 0.997.$$

Thus, a Gaussian random variable deviates from its mean by more than ± 3 standard deviations in only 0.3% of the trials, on the average. We often find tables of the error function rather than that of the CDF of the standard normal random variable.

Many physical experiments result in a nonnegative random variable, whereas the normal random variable takes negative values, as well. Therefore, it may be of interest to define a **truncated normal density**:

$$f(x) = \begin{cases} 0, & x < 0, \\ \frac{1}{\alpha\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right], & x \geq 0, \end{cases} \quad (3.43)$$

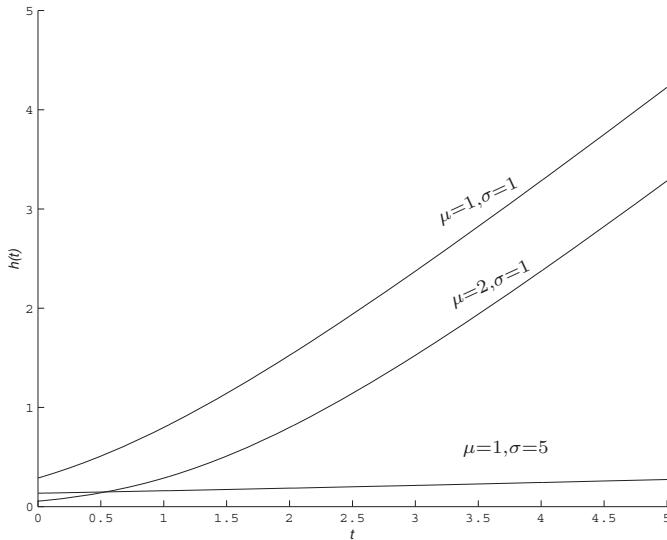


Figure 3.17. Failure rate of the normal distribution

where

$$\alpha = \int_0^\infty \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(t-\mu)^2}{2\sigma^2}\right] dt.$$

The introduction of α insures that $\int_{-\infty}^\infty f(t)dt = 1$, so that f is the density of a nonnegative random variable. For $\mu > 3\sigma$, the value of α is close to 1, and for most practical purposes it may be omitted, so that the truncated normal density reduces to the usual normal density.

The normal distribution is IFR (see Figure 3.17), which implies that it can be used to model the behavior of components during the wearout phase.

Example 3.6

Assuming that the life of a given subsystem, in the wearout phase, is normally distributed with $\mu = 10,000$ h and $\sigma = 1000$ h, determine the reliability for an operating time of 500 h given that (a) the age of the component is 9000 h, (b) the age of the component is 11,000 h.

The required quantity under (a) is $R_{9000}(500)$ and under (b) is $R_{11,000}(500)$. Note that with the usual exponential assumption, these two quantities will be identical. But in the present case

$$\begin{aligned} R_{9000}(500) &= \frac{R(9500)}{R(9000)} \\ &= \frac{\int_{9500}^\infty f(t)dt}{\int_{9000}^\infty f(t)dt}. \end{aligned}$$

Noting that $\mu - 0.5\sigma = 9500$ and $\mu - \sigma = 9000$, we have

$$\begin{aligned}
 R_{9000}(500) &= \frac{\int_{\mu-0.5\sigma}^{\infty} f(t) dt}{\int_{\mu-\sigma}^{\infty} f(t) dt} \\
 &= \frac{1 - F_X(\mu - 0.5\sigma)}{1 - F_X(\mu - \sigma)} \\
 &= \frac{1 - F_Z(-0.5)}{1 - F_Z(-1)} \\
 &= \frac{F_Z(0.5)}{F_Z(1)} \\
 &= \frac{0.6915}{0.8413} \quad [\text{Table C.3 (from Appendix C)}] \\
 &= 0.8219.
 \end{aligned}$$

Similarly, since $\mu + 1.5\sigma = 11,500$ and $\mu + \sigma = 11,000$, we have

$$\begin{aligned}
 R_{11,000}(500) &= \frac{1 - F_X(\mu + 1.5\sigma)}{1 - F_X(\mu + \sigma)} \\
 &= \frac{0.0668}{0.1587} \quad [\text{Table C.3}] \\
 &= 0.4209.
 \end{aligned}$$

Thus, unlike the exponential assumption, $R_{11,000}(500) < R_{9000}(500)$; that is, the subsystem has aged.

‡

It can be shown that the normal distribution is a good approximation to the (discrete) binomial distribution for large n , provided p is not close to 0 or 1. The corresponding parameters are $\mu = np$ and $\sigma^2 = np(1-p)$.

Example 3.7

Twenty percent of VLSI chips made in a certain plant are defective. Assuming that a binomial model is acceptable, the probability of at most 13 rejects in a lot of 100 chosen for inspection may be computed by

$$\sum_{x=0}^{13} b(x; 100, 0.20) = B(13; 100, 0.20).$$

Let us approximate this probability by using the normal distribution with $\mu = np = 20$ and $\sigma^2 = np(1-p) = 16$. From Figure 3.18, observe that we are actually approximating the sum of the areas of the first 14 rectangles of the histogram of the binomial pmf by means of the shaded area under the continuous curve. Thus, it is preferable

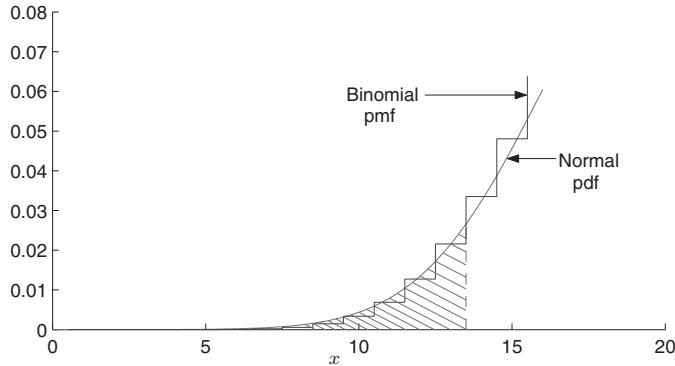


Figure 3.18. Normal approximation to the binomial pmf

to compute the area under the curve between -0.5 and 13.5 rather than between 0 and 13 . Making this **continuity correction**, we get

$$\begin{aligned} B(13; 100, 0.2) &\simeq F_X(13.5) - F_X(-0.5) \\ &= F_Z\left(\frac{13.5 - 20}{4}\right) - F_Z\left(\frac{-0.5 - 20}{4}\right) \\ &= F_Z(-1.625) - F_Z(-5.125) \\ &= 0.0521 - 0, \end{aligned}$$

which compares favorably to the exact value of 0.046912 .

‡

3.4.7 The Uniform or Rectangular Distribution

A continuous random variable X is said to have a uniform distribution over the interval (a, b) if its density is given by (see Figure 3.19):

$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b, \\ 0, & \text{otherwise} \end{cases} \quad (3.44)$$

and the distribution function is given by (see Figure 3.20)

$$F(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a \leq x < b, \\ 1, & x \geq b. \end{cases} \quad (3.45)$$

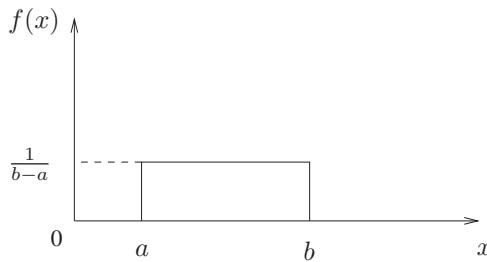


Figure 3.19. The pdf of a uniform distribution

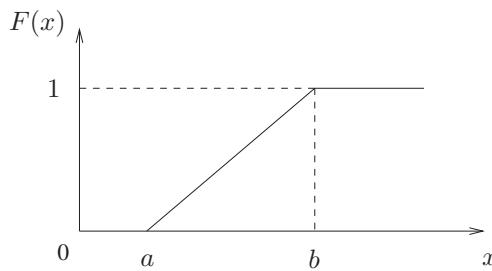


Figure 3.20. The CDF of a uniform distribution

3.4.8 Pareto Distribution

The Pareto distribution, also referred to as the **double-exponential distribution**, the **hyperbolic distribution**, and the **power-law distribution**, has been used to model the amount of CPU time consumed by an arbitrary process [LELA 1986], the Web file size on the Internet servers [CROV 1997, DENG 1996] the thinking time of the Web browser [CROV 1997, DENG 1996], the number of data bytes in FTP (File Transfer Protocol) bursts [PAXS 1995], and the access frequency of Web traffic [NABE 1998].

The density is given by (see Figure 3.21)

$$f(x) = \alpha k^\alpha x^{-\alpha-1}, x \geq k, \alpha, k > 0, \quad (3.46)$$

the distribution function by (see Figure 3.22)

$$F(x) = \begin{cases} 1 - (\frac{k}{x})^\alpha, & x \geq k \\ 0, & x < k \end{cases} \quad (3.47)$$

and the failure rate by (see Figure 3.23)

$$h(x) = \begin{cases} \frac{\alpha}{x}, & x \geq k, \\ 0, & x < k. \end{cases} \quad (3.48)$$

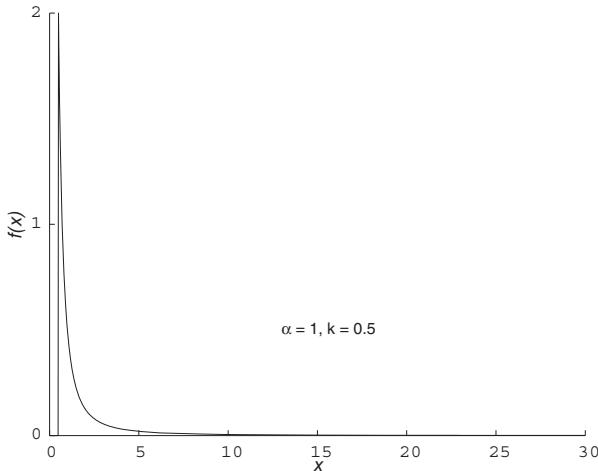


Figure 3.21. The pdf of a Pareto distribution

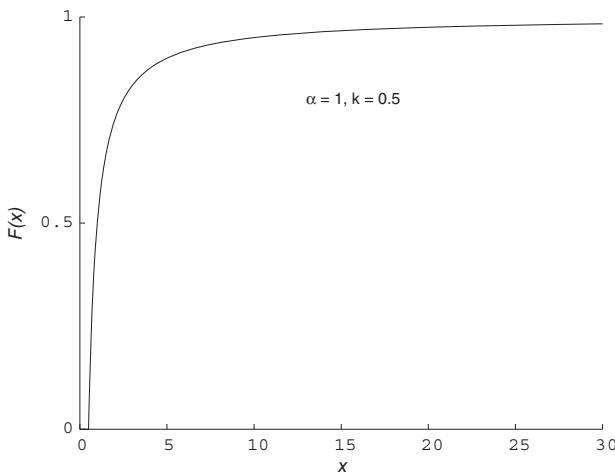


Figure 3.22. The CDF of a Pareto distribution

The location parameter k represents the smallest possible value of the random variable. From the abovementioned papers, the shape parameter α is found to be in the interval $[1.05, 1.25]$ for the amount of CPU time consumed by an arbitrary process, $[0.58, 0.9]$ for the thinking time of the Web browser, $[1.1, 1.3]$ for the Web file size, and $[0.9, 1.1]$ for the number of data bytes in FTP bursts, respectively.

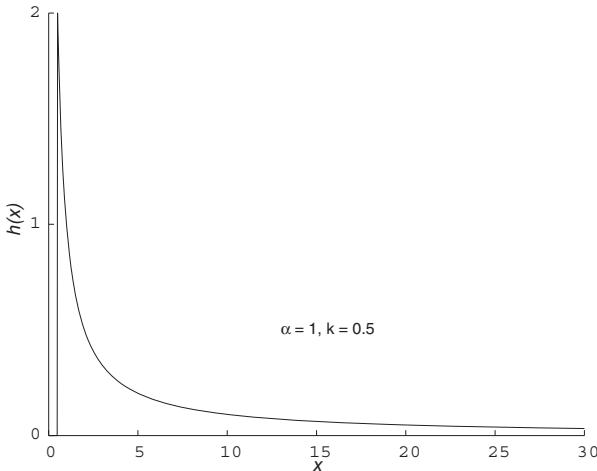


Figure 3.23. The failure rate of a Pareto distribution

3.4.9 Defective Distribution

In many programs there exists a case when the algorithm fails to converge after running for a very long time (e.g., a **while** loop in a C program without any exit condition). Sometimes the hardware underlying the software fails before the program finishes. The running times for such processes follow the defective (or improper) distribution which can be defined using the distribution function $F_X(t)$ such that

$$\lim_{t \rightarrow \infty} F_X(t) < 1.$$

The defect in the random variable X is given by $p_\infty = 1 - \lim_{t \rightarrow \infty} F_X(t)$ and can be thought of as the mass of X at infinity. Note that the pdf $f(x)$ does not satisfy the property (f2).

An extreme case of the defective distribution is $F_X(t) = 0$, $t < \infty$. This implies an event that can *never* occur.

As an example of the defective distribution, consider the distribution function given by (see Figure 3.24)

$$F_X(x) = p_c (1 - e^{-\lambda x}). \quad (3.49)$$

Thus, $\lim_{x \rightarrow \infty} F_X(x) = p_c < 1$. Hence, X is a defective random variable with defect $p_\infty = 1 - p_c$.

For further reading on this topic, the reader is encouraged to study exponential polynomial distributions used in the SHARPE software package [SAHN 1996].

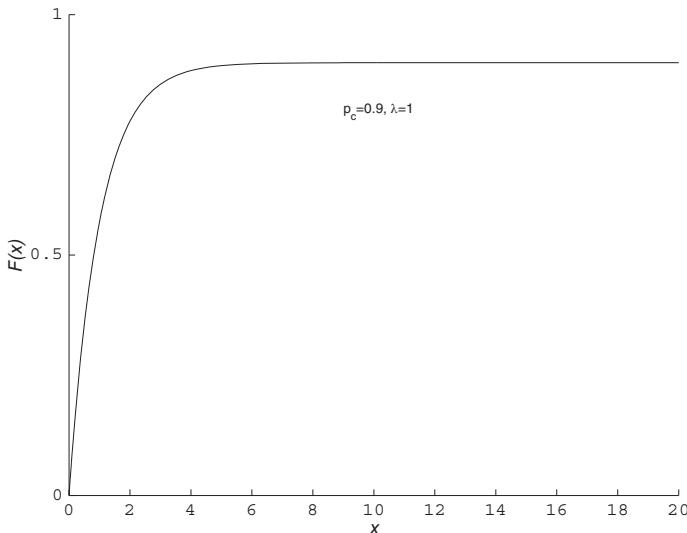


Figure 3.24. Defective distribution function

Problems

- Lifetimes of VLSI chips manufactured by a semiconductor manufacturer are approximately normally distributed with $\mu = 5 \times 10^6$ h and $\sigma = 5 \times 10^5$ h. A computer manufacturer requires that at least 95% of a batch should have a lifetime greater than 4×10^6 h. Will the deal be made?
- Errors occur in data transmission over a binary communication channel due to Gaussian white noise. The probability of an error P_e , can be shown to be

$$P_e = \frac{1}{2} - \frac{1}{\sqrt{\pi}} \int_0^u e^{-y^2} dy = \frac{1}{2} [1 - \text{erf}(u)],$$

where $z = u^2$ is a measure of the ratio of the signal power to noise power. The variable u is usually specified in terms of $10 \log_{10} z$ in decibel units (dB). Plot P_e as a function of $10 \log_{10} z$.

- Show that the failure rate $h(t)$ of the hypoexponential distribution has the property

$$\lim_{t \rightarrow +\infty} h(t) = \min\{\lambda_1, \lambda_2\}.$$

- Show that a two-stage Erlang pdf is the limiting case of two-stage hypoexponential pdf. In other words, show that

$$\lim_{\lambda_1 \rightarrow \lambda_2} \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t}) = \lambda_2^2 t e^{-\lambda_2 t}.$$

(Hint: Use l'Hôpital's rule.)

5. The CPU time requirement of a typical program measured in minutes is found to follow a three-stage Erlang distribution with $\lambda = \frac{1}{2}$. What is the probability that the CPU demand of a program will exceed 1 min?
6. Plot the three-parameter Weibull distribution function [equation (3.33)] for $\lambda = 0.0001$, $\theta = 10$, and $\alpha = 0.5, 1, 2$. Repeat for the density function and the failure rate function.

3.5 FUNCTIONS OF A RANDOM VARIABLE

Situations often arise in systems analysis where knowledge of some characteristic of the system, together with the knowledge of the input, will allow some estimate of the behavior at the output. For example, the input random variable X and its density $f(x)$ are known and the input-output behavior is characterized by

$$Y = \Phi(X).$$

We are interested in computing the density of the random variable Y . Note that for a given random variable X and a function Φ , Y may not satisfy the definition of a random variable. But if we assume that Φ is continuous or piecewise-continuous, then $Y = \Phi(X)$ will be a random variable [ASH 1970].

Example 3.8

Let $Y = \Phi(X) = X^2$. As an example, X could denote the measurement error in a certain physical experiment and Y would then be the square of the error (recall the method of least squares).

Note that $F_Y(y) = 0$ for $y \leq 0$. For $y > 0$,

$$\begin{aligned} F_Y(y) &= P(Y \leq y) \\ &= P(X^2 \leq y) \\ &= P(-\sqrt{y} \leq X \leq \sqrt{y}) \\ &= F_X(\sqrt{y}) - F_X(-\sqrt{y}), \end{aligned}$$

and by differentiation the density of Y is

$$f_Y(y) = \begin{cases} \frac{1}{2\sqrt{y}}[f_X(\sqrt{y}) + f_X(-\sqrt{y})], & y > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.50)$$

#

Example 3.9

As a special case of Example 3.8, let X have the standard normal distribution $[N(0, 1)]$ so that

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad -\infty < x < \infty.$$

Then

$$f_Y(y) = \begin{cases} \frac{1}{2\sqrt{y}} \left(\frac{1}{\sqrt{2\pi}} e^{-y/2} + \frac{1}{\sqrt{2\pi}} e^{-y/2} \right), & y > 0, \\ 0, & y \leq 0, \end{cases}$$

or

$$f_Y(y) = \begin{cases} \frac{1}{\sqrt{2\pi y}} e^{-y/2}, & y > 0, \\ 0, & y \leq 0. \end{cases}$$

By comparing this formula with formula (3.20) and remembering (3.24), we conclude that Y has a gamma distribution with $\alpha = \frac{1}{2}$ and $\lambda = \frac{1}{2}$. Now, since $\text{GAM}(\frac{1}{2}, n/2) = \chi_n^2$, it follows that if X is standard normal then $Y = X^2$ is chi-square distributed with one degree of freedom.

#

Example 3.10

Let X be uniformly distributed on $(0, 1)$. We show that $Y = -\lambda^{-1} \ln(1 - X)$ has an exponential distribution with parameter $\lambda > 0$.

Observe that Y is a nonnegative random variable implying $F_Y(y) = 0$ for $y \leq 0$. For $y > 0$, we have

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P[-\lambda^{-1} \ln(1 - X) \leq y] \\ &= P[\ln(1 - X) \geq -\lambda y] \\ &= P[(1 - X) \geq e^{-\lambda y}] \quad (\text{since } e^x \text{ is an increasing function of } x,) \\ &= P(X \leq 1 - e^{-\lambda y}) \\ &= F_X(1 - e^{-\lambda y}). \end{aligned}$$

But since X is uniform over $(0, 1)$, $F_X(x) = x, 0 \leq x \leq 1$. Thus

$$F_Y(y) = 1 - e^{-\lambda y}.$$

Therefore Y is exponentially distributed with parameter λ .

#

This fact can be used in a distribution-driven simulation. In such simulation programs it is important to be able to generate values of variables with known distribution functions. Such values are known as **random deviates** or **random variates**. Most computer systems provide built-in functions to generate random deviates from the uniform distribution over $(0, 1)$, say, u . Such random deviates are called **random numbers**. For a discussion of random number generation, the reader is referred to Knuth [KNUT 1997].

Examples 3.9 and 3.10 are special cases of problems that can be solved using the following theorem.

THEOREM 3.1. Let X be a continuous random variable with density f_X that is nonzero on a subset I of real numbers [i.e., $f_X(x) > 0, x \in I$ and $f_X(x) = 0, x \notin I$]. Let Φ be a differentiable monotone function whose domain is I and whose range is the set of reals. Then $Y = \Phi(X)$ is a continuous random variable with the density, f_Y , given by

$$f_Y(y) = \begin{cases} f_X[\Phi^{-1}(y)][(\Phi^{-1})'(y)], & y \in \Phi(I), \\ 0, & \text{otherwise,} \end{cases} \quad (3.51)$$

where Φ^{-1} is the uniquely defined inverse of Φ and $(\Phi^{-1})'$ is the derivative of the inverse function.

Proof: We prove the theorem assuming that $\Phi(x)$ is an increasing function of x . The proof for the other case follows in a similar way.

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P[\Phi(X) \leq y] \\ &= P[X \leq \Phi^{-1}(y)], \quad (\text{since } \Phi \text{ is monotone increasing}) \\ &= F_X[\Phi^{-1}(y)]. \end{aligned}$$

Taking derivatives and using the chain rule, we get the required result.

Example 3.11

Now let Φ be the distribution function, F , of a random variable X , with density f . Applying Theorem 3.1, $Y = F(X)$ and $F_Y(y) = F_X(F_X^{-1}(y)) = y$. Therefore the random variable $Y = F(X)$ has the density given by

$$f_Y(y) = \begin{cases} 1, & 0 < y < 1, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, if X is a continuous random variable with CDF F , then the new random variable $Y = F(X)$ is uniformly distributed over the interval $(0, 1)$.

This idea can be used to generate a random deviate x of X by first generating a random number u from a uniform distribution over $(0, 1)$ and then using the relation $x = F^{-1}(u)$ as illustrated in Figure 3.25. For example, if we want to generate an exponentially distributed random deviate, we should first generate a uniformly distributed random number u over $(0, 1)$, and then by the relation given in Example 3.10, which is

$$x = -\lambda^{-1} \ln(1 - u), \quad (3.52)$$

we can obtain a random deviate x of an $\text{EXP}(\lambda)$ random variable from the random number u . ‡

Example 3.12

As another example, we consider the problem of generating a Weibull distributed random deviate x with shape parameter α . The distribution function of X is given by

$$F_X(x) = 1 - e^{-\lambda x^\alpha}. \quad (3.53)$$

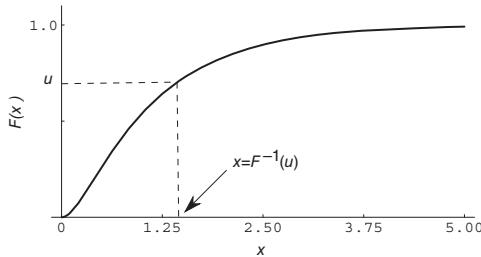


Figure 3.25. Generating a random deviate

We first generate a random number u , then by the relation $x = F_X^{-1}(u)$, we get

$$x = \left(\frac{-\ln(1-u)}{\lambda} \right)^{1/\alpha}. \quad (3.54)$$

This method of generating random deviates is called the **inverse transform method**. For this method, the generation of the $(0, 1)$ random number can be accomplished rather easily. The real question, however, is whether $F^{-1}(u)$ can be expressed in a closed mathematical form. This inversion is possible for distributions such as the exponential and the Weibull, as shown above and for distributions such as Pareto and log-logistic. While for distributions such as the normal, other techniques must be used. For a detailed discussion of random deviate generation, the reader is referred to Fishman [FISH 1995].

#

Another interesting special case of Theorem 3.1 occurs when Φ is linear, that is, when $Y = aX + b$. In other words, Y differs from X only in origin and scale of measurement. In fact, we have already made use of such a transformation when relating $N(\mu, \sigma^2)$ to the standard normal distribution, $N(0, 1)$. The use of Theorem 3.1 yields

$$f_Y(y) = \begin{cases} \frac{1}{|a|} f_X \left(\frac{y-b}{a} \right), & y \in aI + b, \\ 0, & \text{otherwise,} \end{cases} \quad (3.55)$$

where I is the interval over which $f(x) \neq 0$.

Example 3.13

Let X be exponentially distributed, that is, $X \sim \text{EXP}(\lambda)$. Consider a random variable $Y = rX$ where r is a positive real number. Note that the interval I over which $f_X \neq 0$ is $(0, \infty)$. Also

$$f_X(x) = \lambda e^{-\lambda x}.$$

Using formula (3.55), we have

$$f_Y(y) = \frac{1}{r} \lambda e^{-\lambda y/r}, \quad y > 0.$$

It follows that Y is exponentially distributed with parameter λ/r . This result will be used later in this book.

‡

Example 3.14

Let X be normally distributed and consider $Y = e^X$. Since

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right],$$

and

$$\Phi^{-1}(y) = \ln(y) \text{ implies that } [\Phi^{-1}]'(y) = \frac{1}{y};$$

then, using Theorem 3.1, the density of Y is

$$\begin{aligned} f_Y(y) &= \frac{f(\ln y)}{y} \\ &= \frac{1}{\sigma y \sqrt{2\pi}} \exp\left[-\frac{(\ln y - \mu)^2}{2\sigma^2}\right], \quad y > 0. \end{aligned}$$

The random variable Y is said to have log-normal distribution. The importance of this distribution stems from another form of the central-limit theorem, which states that the product of n mutually independent random variables has a log-normal distribution in the limit $n \rightarrow \infty$.

‡

Problems

1. Show that if X has the k -stage Erlang distribution with parameter λ , then

$$Y = 2\lambda X$$

has the chi-square distribution with $2k$ degrees of freedom.

2. Consider a nonlinear amplifier whose input X and the output Y are related by its transfer characteristic:

$$Y = \begin{cases} X^{1/2}, & X \geq 0, \\ -|X|^{1/2}, & X < 0. \end{cases}$$

Assuming that $X \sim N(0, 1)$ compute the pdf of Y and plot your result.

3. The phase X of a sine wave is uniformly distributed over $(-\pi/2, \pi/2)$:

$$f_X(x) = \begin{cases} \frac{1}{\pi}, & -\frac{\pi}{2} < x < \frac{\pi}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

Let $Y = \sin X$ and show that

$$f_Y(y) = \frac{1}{\pi} \frac{1}{\sqrt{1-y^2}}, \quad -1 < y < 1.$$

4. Let X be a chi-square random variable with n (≥ 1) degrees of freedom with the pdf:

$$f(x) = \begin{cases} \frac{x^{n/2-1}}{2^{n/2}\Gamma(n/2)}e^{-x/2}, & x > 0, \\ 0, & \text{elsewhere.} \end{cases}$$

Find the pdf of the random variable $Y = \sqrt{X/n}$.

5. Consider an IBM DTCA-23240 type disk [IBM 1997]. Assume that the number of cylinders N to be traversed between two disk requests is a normally distributed random variable with $\mu = n/3$ and $\sigma^2 = n^2/18$, where n is the total number of cylinders. The seek time T is a random variable related to the seek distance N by

$$T = a + bN.$$

In the particular case of the IBM DTCA-23240, experimental data show $a = 8$ ms, $b = 0.00215$ ms per cylinder, and $n = 6,976$. Determine the pdf of T . Recalling that T is a nonnegative random variable whereas the normal model allows negative values, make appropriate corrections.

6. Consider a normalized floating-point number in base (or radix) β so that the mantissa X satisfies the condition $1/\beta \leq X < 1$. Assume that the density of the continuous random variable X is $1/(x \ln \beta)$.
- Show that a random deviate of X is given by the formula, β^{u-1} where u is a random number.
 - Determine the pdf of the normalized reciprocal $Y = 1/(\beta X)$.
7. Write down formulas to generate random deviates of the log-logistic, the Pareto and the defective exponential [equation (3.49)] distributions.

3.6 JOINTLY DISTRIBUTED RANDOM VARIABLES

So far, we were concerned with the properties of a single random variable. In many practical problems, however, it is important to consider two or more random variables defined on the same probability space.

Let X and Y be two random variables defined on the same probability space (S, \mathcal{F}, P) . The event $[X \leq x, Y \leq y] = [X \leq x] \cap [Y \leq y]$ consists of all sample points $s \in S$ such that $X(s) \leq x$ and $Y(s) \leq y$.

Definition (Joint Distribution Function). The joint (or compound) distribution function of random variables X and Y is defined by

$$F_{X,Y}(x, y) = P(X \leq x, Y \leq y), \quad -\infty < x < \infty, -\infty < y < \infty.$$

The subscripts will be dropped whenever the two random variables under consideration are clear from the context; that is, $F_{X,Y}(x, y)$ will be written as $F(x, y)$. Such a function satisfies the following properties:

(J1) $0 \leq F(x, y) \leq 1, -\infty < x < \infty, -\infty < y < \infty$. This is evident since $F(x, y)$ is a probability.

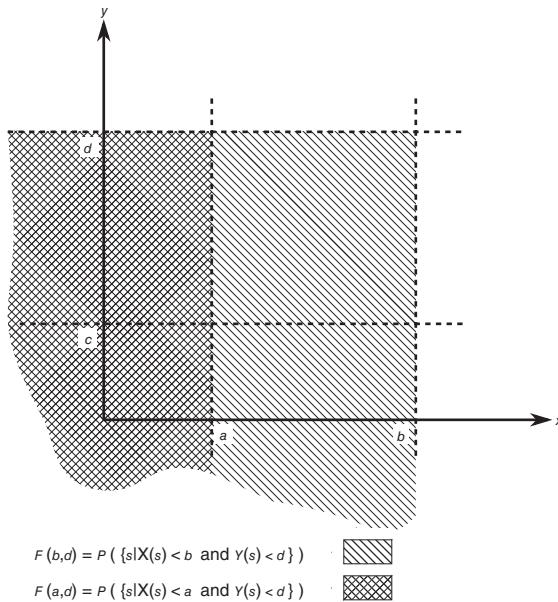


Figure 3.26. Properties of joint CDF

(J2) $F(x,y)$ is monotone increasing in both the variables; that is, if $x_1 \leq x_2$ and $y_1 \leq y_2$, then $F(x_1, y_1) \leq F(x_2, y_2)$. This follows since the event $[X \leq x_1 \text{ and } Y \leq y_1]$ is contained in the event $[X \leq x_2 \text{ and } Y \leq y_2]$.

(J3) If either x or y approaches $-\infty$, then $F(x,y)$ approaches 0, and if both x and y approach $+\infty$, then $F(x,y)$ approaches 1.

(J4) $F(x,y)$ is right continuous in general, and if X and Y are continuous random variables, then $F(x,y)$ is continuous.

(J5) $P(a < X \leq b \text{ and } c < Y \leq d) = F(b,d) - F(a,d) - F(b,c) + F(a,c)$. This relation follows from Figure 3.26.

Note that in the limit $y \rightarrow \infty$, the event $[X \leq x, Y \leq y]$ approaches the event $[X \leq x, Y < \infty] = [X \leq x]$. Therefore, $\lim_{y \rightarrow \infty} F_{X,Y}(x,y) = F_X(x)$. Also $\lim_{x \rightarrow \infty} F_{X,Y}(x,y) = F_Y(y)$. These two formulas show how to compute the **individual or marginal distribution functions** of X and Y given their joint distribution function.

If both X and Y are continuous random variables, then we can often find a function $f(x,y)$ such that

$$F(x,y) = \int_{-\infty}^y \int_{-\infty}^x f(u,v) \, du \, dv. \quad (3.56)$$

Such a function is known as the **joint** or the **compound probability density function** of X and Y . It follows that

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) \ du \ dv = 1 \quad (3.57)$$

and

$$P(a < X \leq b, \ c < Y \leq d) = \int_a^b \int_c^d f(x, y) \ dy \ dx. \quad (3.58)$$

Also

$$f(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y}, \quad (3.59)$$

assuming that the partial derivative exists. Now, since

$$F_X(x) = \int_{-\infty}^x \int_{-\infty}^{\infty} f(u, y) \ dy \ du,$$

we obtain the marginal density f_X as

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y) \ dy. \quad (3.60)$$

Similarly

$$f_Y(y) = \int_{-\infty}^{\infty} f(x, y) \ dx. \quad (3.61)$$

Thus the marginal densities, $f_X(x)$ and $f_Y(y)$, can easily be determined from the knowledge of the joint density, $f(x, y)$. However, the knowledge of the marginal densities does not, in general, uniquely determine the joint density. The exception occurs when the two random variables are independent.

Intuitively, if X and Y are independent random variables, then we expect that events such as $[X \leq x]$ and $[Y \leq y]$ will be independent events.

Definition (Independent Random Variables). We define two random variables X and Y to be **independent** if

$$F(x, y) = F_X(x)F_Y(y), \ -\infty < x < \infty, \ -\infty < y < \infty.$$

Thus the independence of random variables X and Y implies that their joint CDF factors into the product of the marginal CDFs. This definition applies to all types of random variables. In case X and Y are discrete, the preceding definition of independence is equivalent to definition given in Chapter 2:

$$p(x, y) = p_X(x) p_Y(y).$$

In the case that X and Y are continuous, the preceding definition of independence is equivalent to the condition

$$f(x, y) = f_X(x) f_Y(y), \ -\infty < x < \infty, \ -\infty < y < \infty,$$

assuming that $f(x, y)$ exists. We will also have occasion to consider the joint distribution of X and Y when one of them is a discrete random variable while the other is a continuous random variable. In case X is discrete and Y is continuous, the condition for their independence becomes

$$P(X = x, Y \leq y) = p_X(x) F_Y(y), \quad \text{all } x \text{ and } y.$$

The definition of joint distribution, joint density, and independence of two random variables can be easily generalized to a set of n random variables X_1, X_2, \dots, X_n .

Example 3.15

Assume that the lifetime X and the brightness Y of a lightbulb are being modeled as continuous random variables. Let the joint pdf be given by

$$f(x, y) = \lambda_1 \lambda_2 e^{-(\lambda_1 x + \lambda_2 y)}, \quad 0 < x < \infty, 0 < y < \infty,$$

(this is known as the **bivariate exponential density**). The marginal density of X is

$$\begin{aligned} f_X(x) &= \int_{-\infty}^{\infty} f(x, y) dy \\ &= \int_0^{\infty} \lambda_1 \lambda_2 e^{-(\lambda_1 x + \lambda_2 y)} dy \\ &= \lambda_1 e^{-\lambda_1 x}, \quad 0 < x < \infty. \end{aligned}$$

Similarly

$$f_Y(y) = \lambda_2 e^{-\lambda_2 y}, \quad 0 < y < \infty.$$

It follows that X and Y are independent random variables. The joint distribution function can be computed to be

$$\begin{aligned} F(x, y) &= \int_{-\infty}^x \int_{-\infty}^y f(u, v) dv du \\ &= \int_0^x \int_0^y \lambda_1 \lambda_2 e^{-(\lambda_1 u + \lambda_2 v)} dv du \\ &= (1 - e^{-\lambda_1 x})(1 - e^{-\lambda_2 y}), \quad 0 < x < \infty, 0 < y < \infty. \end{aligned}$$

#

Problems

- A batch of 1M RAM chips are purchased from two different semiconductor houses. Let X and Y denote the times to failure of the chips purchased from the two suppliers. The joint probability density of X and Y is estimated by

$$f(x, y) = \begin{cases} \lambda \mu e^{-(\lambda x + \mu y)}, & x > 0, y > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Assume $\lambda = 10^{-5}$ per hour and $\mu = 10^{-6}$ per hour.

Determine the probability that the time to failure is greater for chips characterized by X than it is for chips characterized by Y .

2. Let X and Y have joint pdf

$$f(x, y) = \begin{cases} \frac{1}{\pi}, & x^2 + y^2 \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Determine the marginal pdf's of X and Y . Are X and Y independent?

3. Consider a series connection of two components, with respective lifetimes X and Y . The joint pdf of the lifetimes is given by

$$f(x, y) = \begin{cases} \frac{1}{200}, & (x, y) \in A, \\ 0, & \text{elsewhere,} \end{cases}$$

where A is the triangular region in the (x, y) plane with the vertices $(100, 100)$, $(100, 120)$, and $(120, 120)$. Find the reliability expression for the entire system.

4. If the random variables B and C are independent and uniformly distributed over $(0,1)$, compute the probability that the roots of the equation

$$x^2 + 2Bx + C = 0$$

are real.

5. Let the joint pdf of X and Y be given by

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right],$$

where $|\rho| < 1$. Show that the marginal pdf's are

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}.$$

The random variables X and Y are said to have a two-dimensional (or bivariate) normal pdf. Also note that X and Y are *not* independent unless $\rho = 0$.

3.7 ORDER STATISTICS

Let X_1, X_2, \dots, X_n be mutually independent, identically distributed continuous random variables, each having the distribution function F and density f . Let Y_1, Y_2, \dots, Y_n be random variables obtained by permuting the set X_1, X_2, \dots, X_n so as to be in increasing order. To be specific

$$Y_1 = \min\{X_1, X_2, \dots, X_n\},$$

and

$$Y_n = \max\{X_1, X_2, \dots, X_n\}.$$

The random variable Y_k is called the **k th-order statistic**. Since X_1, X_2, \dots, X_n are continuous random variables, it follows that $Y_1 < Y_2 < \dots < Y_n$ (as opposed to $Y_1 \leq Y_2 \leq \dots \leq Y_n$) with a probability of one.

As examples of the use of order statistics, let X_i be the lifetime of the i th component in a system of n components. If the system is a series system, then Y_1 will denote the overall system lifetime. Similarly, Y_n will denote the lifetime of a parallel system and Y_{n-k+1} will denote the lifetime of an k -out-of- n system.

To derive the distribution function of Y_k , we note that the probability that exactly j of the X_i values lie in $(-\infty, y]$ and $(n-j)$ lie in (y, ∞) is

$$\binom{n}{j} F^j(y) [1 - F(y)]^{n-j},$$

since the binomial pmf with parameters n and $p = F(y)$ is applicable. Then

$$\begin{aligned} F_{Y_k}(y) &= P(Y_k \leq y) \\ &= P(\text{"at least } k \text{ of the } X_i \text{'s lie in the interval } (-\infty, y] \text{"}) \\ &= \sum_{j=k}^n \binom{n}{j} F^j(y) [1 - F(y)]^{n-j}, \quad -\infty < y < \infty. \end{aligned} \tag{3.62}$$

(For a generalization of this formula to the case of when $\{X_i\}$ are not identically distributed, see Sahner et al. [SAHN 1996].) In particular, the distribution functions of Y_n and Y_1 can be obtained from (3.62) as

$$F_{Y_n}(y) = [F(y)]^n, \quad -\infty < y < \infty,$$

and

$$F_{Y_1}(y) = 1 - [1 - F(y)]^n, \quad -\infty < y < \infty.$$

From this we obtain

$$\begin{aligned} R_{\text{series}}(t) &= R_{Y_1}(t) \\ &= 1 - F_{Y_1}(t) \\ &= [1 - F(t)]^n \\ &= [R(t)]^n, \end{aligned}$$

and

$$\begin{aligned} R_{\text{parallel}}(t) &= R_{Y_n}(t) \\ &= 1 - F_{Y_n}(t) \\ &= 1 - [F(t)]^n \\ &= 1 - [1 - R(t)]^n. \end{aligned}$$

Both these formulas can be easily generalized to the case when the lifetime distributions of individual components are distinct:

$$R_{\text{series}}(t) = \prod_{i=1}^n R_i(t), \quad (3.63)$$

and

$$R_{\text{parallel}}(t) = 1 - \prod_{i=1}^n (1 - R_i(t)). \quad (3.64)$$

Example 3.16

Let the lifetime distribution of the i th component be exponential with parameter λ_i . Then equation (3.63) reduces to

$$R_{\text{series}}(t) = \exp \left[- \left(\sum_{i=1}^n \lambda_i \right) t \right],$$

so that the lifetime distribution of a series system whose components have independent exponentially distributed lifetimes is itself exponentially distributed with parameter $\sum_{i=1}^n \lambda_i$.

This fact is responsible for the “parts count method” of system reliability analysis often used in practice. Using this method, the analyst counts the number n_i of parts of type i each with a failure rate λ_i . Now if there are k such part types, then the system failure rate λ is computed by

$$\lambda = \sum_{i=1}^k \lambda_i n_i. \quad (3.65)$$

‡

Clary and others [CLAR 1978] present the following example of the parts count method of reliability analysis.

Example 3.17

One implementation of the CPU–cache–main memory subsystem of computer consists of the following chip types

<i>Chip type</i>	<i>Number of chips,</i> n_i	<i>Failure rate per chip</i> (number of failures/ 10^6 h), λ_i
SSI	1,202	0.1218
MSI	668	0.242
ROM	58	0.156
RAM	414	0.691
MOS	256	1.0602
BIP	2,086	0.1588

It is assumed that times to failure of all chip types are exponentially distributed with the failure rate shown above. All the chips are required to be fault-free in order for the system as a whole to be fault-free (i.e., a series system). The system time to failure is then exponentially distributed with parameter:

$$\begin{aligned}\lambda &= \sum_{\text{all chip types}} n_i \lambda_i \\ &= 146.40 + 161.66 + 9.05 + 286.07 + 261.41 + 331.27 \\ &= 1205.85 \text{ failures per } 10^6 \text{ hours.}\end{aligned}$$

‡

Example 3.18

We have seen that the lifetime of a series system is exponentially distributed, provided the component lifetimes are exponentially distributed. Thus a series system whose components have constant failure rate itself has a constant failure rate. *This does not apply to a parallel system.* The failure rate of a parallel system is a function of its age, even though the failure rates of individual components are constant. It can be shown that the corresponding distribution is IFR. In particular, the reliability of a parallel system of n components, each with an exponential failure law (parameter λ), is given by

$$\begin{aligned}R_p(t) &= 1 - (1 - e^{-\lambda t})^n \\ &= \binom{n}{1} e^{-\lambda t} - \binom{n}{2} e^{-2\lambda t} + \dots + (-1)^{n-1} e^{-n\lambda t}.\end{aligned}\quad (3.66)$$

Figure 3.27 shows the reliability improvement obtained by parallel redundancy.

‡

Example 3.19

Another interesting case of order statistics occurs when we consider the triple modular redundant (TMR) system ($n = 3$ and $k = 2$). Y_2 then denotes the time until

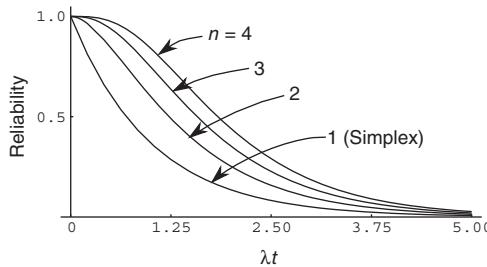


Figure 3.27. Reliability of a parallel-redundant system

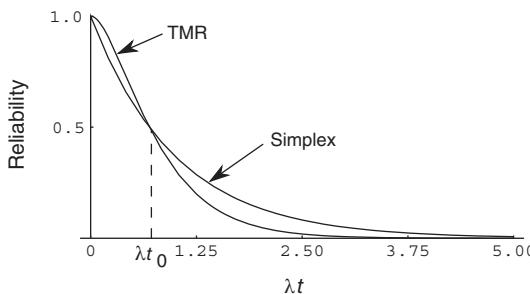


Figure 3.28. Comparison of TMR and simplex reliabilities

the second failure. From equation (3.62), we get

$$R_{\text{TMR}}(t) = 3R^2(t) - 2R^3(t). \quad (3.67)$$

Assuming that the reliability of a single component is given by $R(t) = e^{-\lambda t}$, we get

$$R_{\text{TMR}}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}. \quad (3.68)$$

In Figure 3.28, we have plotted $R_{\text{TMR}}(t)$ against t as well as $R(t)$ against t . Note that

$$R_{\text{TMR}}(t) \geq R(t), \quad 0 \leq t \leq t_0,$$

and

$$R_{\text{TMR}}(t) \leq R(t), \quad t_0 \leq t < \infty,$$

where t_0 is the solution to the equation

$$3e^{-2\lambda t_0} - 2e^{-3\lambda t_0} = e^{-\lambda t_0},$$

which is

$$t_0 = \frac{\ln 2}{\lambda} \simeq \frac{0.7}{\lambda}.$$

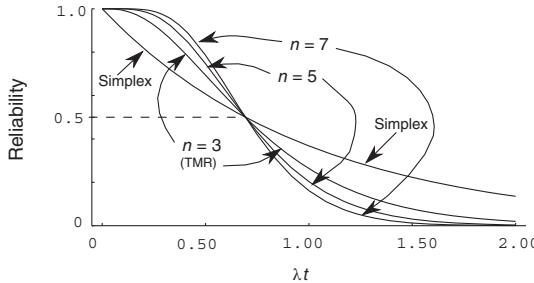


Figure 3.29. Comparison of k -out-of- n and simplex reliabilities

Thus, if we define a “short” mission by the mission time $t \leq t_0$, then it is clear that TMR type of redundancy improves reliability only for short missions. For long missions, this type of redundancy actually degrades reliability. The same type of behavior is exhibited by any k -out-of- n system, with $n = 2k - 1$ (see Figure 3.29). \sharp

Example 3.20

Consider a computer system with jobs arriving from several independent sources. Let $N_i(t)$ ($1 \leq i \leq n$) denote the number of jobs arriving in the interval $(0, t]$ from source i . Assume that $N_i(t)$ is Poisson distributed with parameter $\lambda_i t$. Then the time between two arrivals from source i , denoted by X_i , is exponentially distributed with parameter λ_i (refer to Example 3.2). Also from Theorem 2.2(c), the total number of jobs arriving from all sources in the interval $(0, t]$, $N(t) = \sum_{i=1}^n N_i(t)$, is Poisson distributed with parameter $\sum_{i=1}^n \lambda_i t$. Then, again recalling Example 3.2, the interarrival time, Y_1 , for jobs from all sources will be exponentially distributed with parameter $\sum_{i=1}^n \lambda_i$. But this could also be derived using order statistics. Note that

$$Y_1 = \min\{X_1, X_2, \dots, X_n\},$$

since Y_1 is the time until the next arrival from any one of the sources. Now, since

$$F_{X_i}(t) = 1 - e^{-\lambda_i t},$$

it follows that

$$\begin{aligned} F_{Y_1}(t) &= 1 - \prod_{i=1}^n [1 - F_{X_i}(t)] \\ &= 1 - \prod_{i=1}^n e^{-\lambda_i t} \\ &= 1 - \exp\left[-\sum_{i=1}^n \lambda_i t\right]. \end{aligned}$$

Thus Y_1 is exponentially distributed with parameter $\sum_{i=1}^n \lambda_i$. \sharp

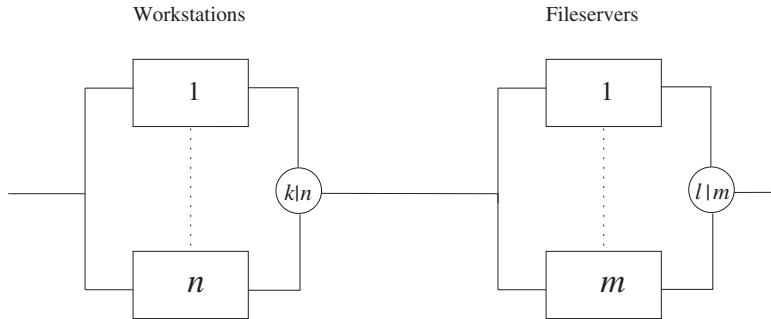


Figure 3.30. Reliability block diagram for the workstation–file server (WFS) example

Example 3.21

We consider a system consisting of n workstations and m file servers. The network connecting these devices is assumed to be fault-free. The system is considered to be operational so long as at least k workstations and l file servers are operational. Reliability block diagram for the system is shown in Figure 3.30. Let $R_w(t)$ denote the reliability of a single workstation and $R_f(t)$ the reliability of a single file server. Assuming that all devices fail independently of each other, system reliability is given by

$$R(t) = \sum_{j=k}^n \binom{n}{j} [R_w(t)]^j [1 - R_w(t)]^{n-j} \sum_{j=l}^m \binom{m}{j} [R_f(t)]^j [1 - R_f(t)]^{m-j}.$$

‡

Example 3.22

An $N \times N$ shuffle exchange network (SEN) with $N = 2^n$ inputs consists of $(N/2) \log_2 N$ switching elements, that is, there are $N/2$ switching elements in each of the $\log_2 N$ stages. Each switching element either transmits the inputs directly through itself or exchanges the inputs as shown in Figure 3.31. Unique path exists between each source–destination pair of the SEN. An 8×8 SEN is shown in Figure 3.32.

The SEN is a self-routing network. A message from any source to a given destination is routed through the network according to the binary representation of the destination’s address. For example, if S = 000 wants to send a message to D = 101,

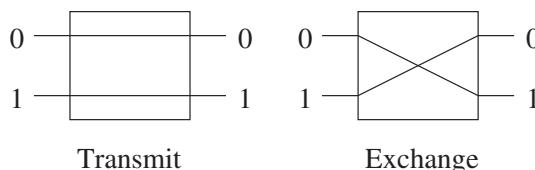


Figure 3.31. Transmit and exchange operations of a switching element

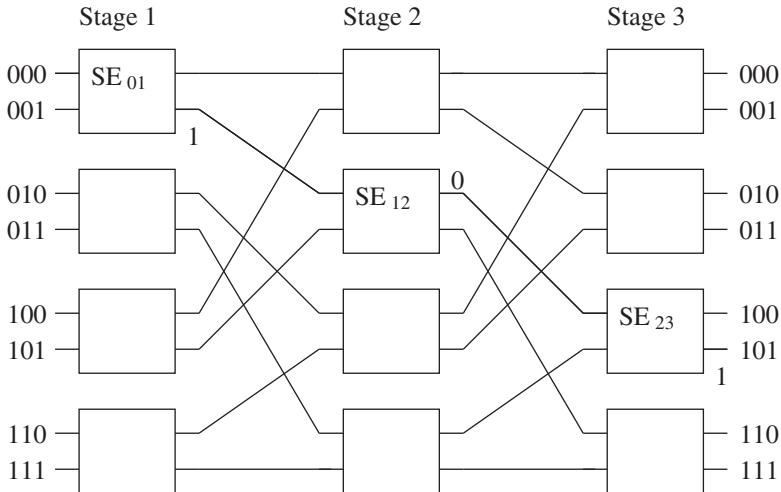


Figure 3.32. An 8×8 SEN

the message is routed as follows. The first bit of the destination address (1) is used and the output link 1 of SE_{01} passes the message to SE_{12} , now the second bit of D is used and the output link 0 of SE_{12} passes the message to SE_{23} , finally, the message appears at the output link 1 of SE_{23} .

Given that $r_{SE}(t)$ is the time-dependant reliability of each switching element, we calculate the reliability of the $N \times N$ SEN. The SEN is operational as long as every source is able to communicate with every destination. Since a unique path exists between each source–destination pair of the SEN, a failure in any switching element will lead to the SEN failure. Thus, from a reliability point of view, we have $(N/2)\log_2 N$ switching elements in series [BLAK 1989].

$$\begin{aligned}
 \text{Reliability of the SEN} &= R_{\text{SEN}}(t) \\
 &= \text{Reliability of a series system of } (N/2)\log_2 N \text{ elements} \\
 &= [r_{SE}(t)]^{(N/2)\log_2 N}
 \end{aligned}$$

#

Example 3.23

As an extension to the SEN, called SEN+, consists of an extra stage consisting of $N/2$ switching elements. This introduces an additional path between each source and destination of the network. The paths in the first and last stages are not disjoint,

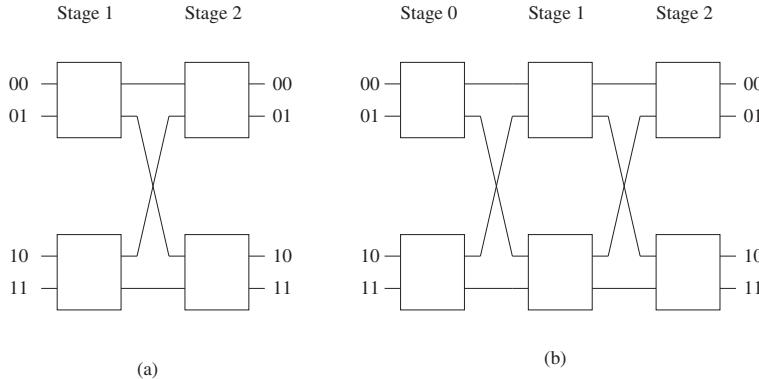


Figure 3.33. A 4×4 SEN and a SEN+.

but those in the intermediate stages are disjoint. As an example, $S = 000$ can reach $D = 101$ by two different paths.

Let $r_{\text{SE}}(t)$ be the time-dependant reliability of the switching element. We compare the reliability of the 4×4 SEN+ with that of the 4×4 SEN. First, since the SEN corresponds to a series system consisting of $(N/2)\log_2 N = 4$ elements:

$$R_{\text{SEN}}(t) = [r_{\text{SE}}(t)]^4.$$

In the 4×4 SEN+ network shown in Figure 3.33, there are six SEs, two in each of the three stages. The SEs in the first and last stages are required to work for full connectivity. The intermediate stages can tolerate 1 failure, which from a reliability point of view is 2 SEs in parallel. Thus, the 4×4 SEN+ can be considered to be a series-parallel system with 4 elements in series and two elements in parallel as shown in Figure 3.34.

Calculating the reliability of this series-parallel system, we get

$$R_{\text{SEN+}}(t) = [r_{\text{SE}}(t)]^4[1 - (1 - [r_{\text{SE}}(t)]^2)].$$

Although an additional stage has been introduced in the 4×4 SEN+, the SEN has a better reliability than the SEN+. This is because the SEN+ is fault tolerant with respect to switches in the intermediate stages only. It can be shown that for $N \geq 8$, the SEN+ is strictly more reliable than the SEN [BLAK 1989]. For example, the reliability of the 8×8 SEN+ (see Figure 3.35) is determined (using Markov chain methods of Chapter 8) to be

$$R_{\text{SEN+}}(t) = 2[r_{\text{SE}}(t)]^{12} + 4[r_{\text{SE}}(t)]^{14} - 8[r_{\text{SE}}(t)]^{15} + 3[r_{\text{SE}}(t)]^{16} \geq [r_{\text{SE}}(t)]^{12} = R_{\text{SEN}}(t).$$

廿

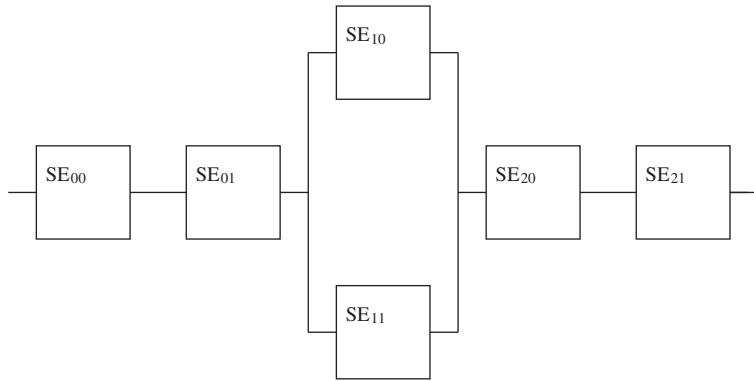


Figure 3.34. Series–parallel reliability block diagram of 4×4 SEN+

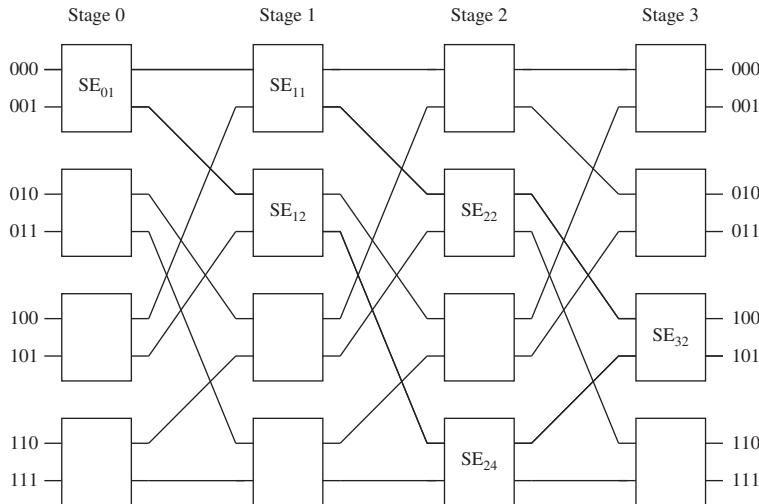


Figure 3.35. An 8×8 SEN+

Problems

1. A system with three independent components works correctly if at least one component is functioning properly. Failure rates of the individual components are $\lambda_1 = 0.0001$, $\lambda_2 = 0.0002$, and $\lambda_3 = 0.0004$ (assume exponential lifetime distributions).
 - (a) Determine the probability that the system will work for 1000 h.
 - (b) Determine the density function of the lifetime X of the system.
2. A multiprocessor system has n processors. Service time of a process executing on the i th processor is exponentially distributed with parameter μ_i ($i = 1, 2, \dots, n$).

Given that all n processors are active and that they are executing mutually independent processes, what is the distribution of time until a processor becomes idle?

3. Consider a series system consisting of n independent components. Assuming that the lifetime of the i th component is Weibull distributed with parameter λ_i and α , show that the system lifetime also has a Weibull distribution.

As a concrete example, consider a liquid cooling cartridge system that is used in enterprise-class servers made by Sun Microsystems [KOSL 2001]. The series system consists of a blower, a water pump and a compressor. The following table gives the Weibull data for the three components.

<i>Component</i>	<i>L10 (h)</i>	<i>Shape parameter (α)</i>
Blower	70,000	3.0
Water pump	100,000	3.0
Compressor	100,000	3.0

$L10$ is the *rating life* of the component, which is the time at which 10 % of the components are expected to have failed or $R(L10) = 0.9$. Derive the system reliability expression.

4. Consider a random access memory card consisting of d VLSI chips, each containing w bits. Each chip supplies one bit position in a d -bit word for a total of w , d -bit words. Assuming a failure rate of λ per chip (and an exponential lifetime distribution), derive the reliability expression $R_0(t)$ for the memory card. Suppose that we now introduce single error correction, so that up to one chip may fail without a system failure. Note that this will require c extra chips where c must satisfy the relation $c \geq \log_2(c + d + 1)$. (See Rao and Fujiwara [RAO 1989].) Derive the reliability expression $R_1(t)$. Plot $R_0(t)$ and $R_1(t)$ as functions of t on the same plot. Derive expressions for the hazard functions $h_0(t)$ and $h_1(t)$, and plot these as functions of time t . For these plots assume $d = 16$ (hence $c = 5$) and $\lambda = 10$ per million hours.
5. The memory requirement distribution for jobs in a computer system is exponential with parameter λ . The memory scheduler scans all eligible jobs on a job queue and loads the job with the smallest memory requirement first, then the job with the next smallest memory requirement, etc. Given that there are n jobs on the job queue, write down the distribution function for the memory requirement of the job with the smallest memory requirement, and that of the job with the largest memory requirement.
6. Redo the above example assuming that the memory requirement distribution for jobs is three-parameter Weibull [equation (3.33)].
7. A series system has n independent components. For $i = 1, 2, \dots, n$, the lifetime X_i of the i th component is exponentially distributed with parameter λ_i . Compute the probability that a given component $j = (1, 2, \dots, n)$ is the cause of the system failure.

8. Consider a system with n independent components each having the modified exponential lifetime distribution with initial mass p_0 at origin and rate λ [see equation (3.2)]. Derive reliability expression for a k -out-of- n system and specialize to the cases of $k = 1$ (parallel) and $k = n$ (series).
9. Repeat problem 8 above with the defective exponential distribution (3.49).

3.8 DISTRIBUTION OF SUMS

Let X and Y be continuous random variables with joint density f . In many situations we are interested in the density of a random variable Z that is a function of X and Y —that is, $Z = \Phi(X, Y)$. The distribution function of Z may be computed by

$$\begin{aligned} F_Z(z) &= P(Z \leq z) \\ &= \iint_{A_z} f(x, y) \, dx \, dy \end{aligned} \quad (3.69)$$

where A_z is a subset of \mathbb{R}^2 given by

$$\begin{aligned} A_z &= \{(x, y) \mid \Phi(x, y) \leq z\} \\ &= \Phi^{-1}((-\infty, z]). \end{aligned}$$

One function of special interest is $Z = X + Y$ with

$$A_z = \{(x, y) \mid x + y \leq z\},$$

which is the half-plane to the lower left of the line $x + y = z$ (see Figure 3.36). Then

$$\begin{aligned} F_Z(z) &= \iint_{A_z} f(x, y) \, dx \, dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{z-x} f(x, y) \, dy \, dx. \end{aligned}$$

Making a change of variable $y = t - x$, we get

$$\begin{aligned} F_Z(z) &= \int_{-\infty}^{\infty} \int_{-\infty}^z f(x, t - x) \, dt \, dx \\ &= \int_{-\infty}^z \int_{-\infty}^{\infty} f(x, t - x) \, dx \, dt \\ &= \int_{-\infty}^z f_Z(t) dt \end{aligned}$$

by the definition of density. Thus the density of $Z = X + Y$ is given by

$$f_Z(z) = \int_{-\infty}^{\infty} f(x, z - x) dx, \quad -\infty < z < \infty. \quad (3.70)$$

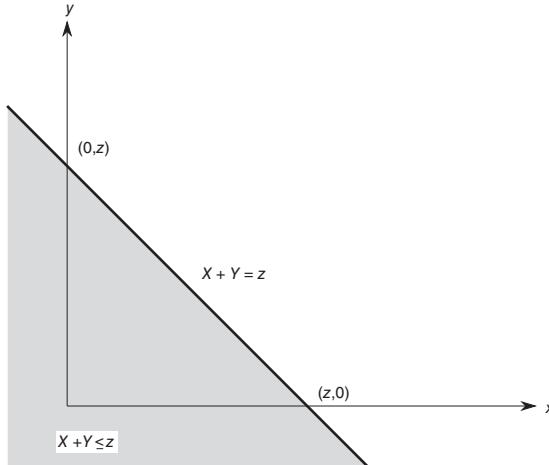


Figure 3.36. Area of integration for the convolution of X and Y

Now if X and Y are assumed to be independent, then $f(x, y) = f_X(x)f_Y(y)$, and formula (3.70) reduces to

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x)f_Y(z-x)dx, \quad -\infty < z < \infty. \quad (3.71)$$

Furthermore, if X and Y are nonnegative random variables, then

$$f_Z(z) = \int_0^z f_X(x)f_Y(z-x)dx, \quad 0 < z < \infty. \quad (3.72)$$

This integral is often called the **convolution** of f_X and f_Y . Thus the density of the sum of two nonnegative independent random variables is the convolution of the individual densities.

Example 3.24

Consider a job consisting of three tasks. Tasks 1 and 2 are noninterfering and hence can be executed in parallel. Task 3 cannot be started until both task 1 and task 2 have completed. If T_1 , T_2 , and T_3 , respectively, denote the times of execution of three tasks, then the time of execution of the entire job is given by

$$\begin{aligned} T &= \max\{T_1, T_2\} + T_3 \\ &= M + T_3. \end{aligned}$$

Assume that T_1 and T_2 are continuous random variables with uniform distribution over $(t_1 - t_0, t_1 + t_0)$ and T_3 is a continuous random variable uniformly distributed

over $(t_3 - t_0, t_3 + t_0)$. Also assume that T_1 , T_2 , and T_3 are mutually independent. We are asked to compute the probability that $T > t_1 + t_3$.

Note that

$$\begin{aligned} f_{T_1}(t) &= f_{T_2}(t) \\ &= \begin{cases} \frac{1}{2t_0}, & t_1 - t_0 < t < t_1 + t_0, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

and

$$f_{T_3}(t) = \begin{cases} \frac{1}{2t_0}, & t_3 - t_0 < t < t_3 + t_0, \\ 0, & \text{otherwise.} \end{cases}$$

First compute the distribution of the random variable M :

$$\begin{aligned} F_M(m) &= P(M \leq m) \\ &= P(\max\{T_1, T_2\} \leq m) \\ &= P(T_1 \leq m \text{ and } T_2 \leq m) \\ &= P(T_1 \leq m)P(T_2 \leq m) \quad \text{by independence} \\ &= F_{T_1}(m)F_{T_2}(m). \end{aligned}$$

Now observe that

$$F_{T_1}(t) = \begin{cases} 0, & t < t_1 - t_0, \\ \frac{t - t_1 + t_0}{2t_0}, & t_1 - t_0 \leq t < t_1 + t_0 \\ 1, & \text{otherwise.} \end{cases}$$

Thus

$$F_M(m) = \begin{cases} 0, & m < t_1 - t_0, \\ \frac{(m - t_1 + t_0)^2}{4t_0^2}, & t_1 - t_0 \leq m < t_1 + t_0, \\ 1, & \text{otherwise.} \end{cases}$$

Also

$$f_M(m) = \begin{cases} \frac{m - t_1 + t_0}{2t_0^2}, & t_1 - t_0 < m < t_1 + t_0, \\ 0, & \text{otherwise.} \end{cases}$$

Now consider the (M, T_3) plane as shown in Figure 3.37 and let A denote the shaded region. Then

$$\begin{aligned} P(T > t_1 + t_3) &= \int \int_A f_{M, T_3}(m, t) \ dm \ dt \\ &= \int \int_A f_M(m)f_{T_3}(t) \ dm \ dt \end{aligned}$$

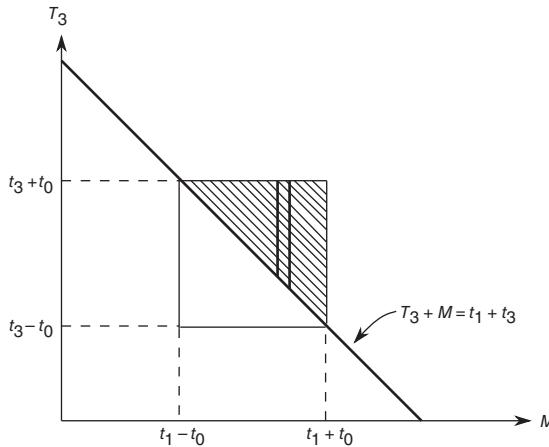


Figure 3.37. The area of integration for Example 3.24

since M and T_3 are independent, it follows that

$$P(T > t_1 + t_3) = \int_{t_1 - t_0}^{t_1 + t_0} \left(\int_{t_1 + t_3 - m}^{t_3 + t_0} \frac{m - t_1 + t_0}{4t_0^3} dt \right) dm = \frac{2}{3}.$$

‡

For further examples of performance analysis of jobs consisting of multiple tasks, see Sahner et al. [SAHN 1996].

In the rest of this section we will concentrate on sums of independent exponentially distributed random variables.

Example 3.25

Consider a system with two statistically identical components, each with an exponential failure law with parameter λ . Only one component is required to be operative for the system to function properly. One method of utilizing the second component is to use parallel redundancy, in which both components are initially operative simultaneously. Alternately, we could initially keep the spare component in a powered-off state (deenergized) and later, on the failure of the operative component, replace it by the spare. Assuming that a deenergized component does not fail (this is sometimes known as a “cold spare”) and that the failure detection and switching equipment is perfect, we can characterize the lifetime Z of such a system in terms of the lifetimes X and Y of individual components by $Z = X + Y$. Such a system is known to possess **standby redundancy** in contrast to a system with parallel redundancy. Now if the random variables X and Y are assumed to be independent, then the density of Z is the convolution of the densities of X and Y .

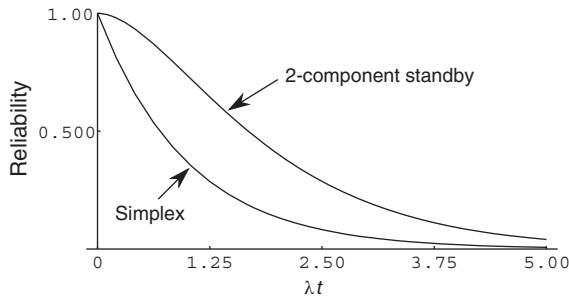


Figure 3.38. Reliabilities of simplex and two-component standby systems

Let X and Y have exponential distributions with parameter λ ; that is, $f_X(x) = \lambda e^{-\lambda x}$, $x > 0$, and $f_Y(y) = \lambda e^{-\lambda y}$, $y > 0$. Then using the convolution formula (3.72) we have

$$\begin{aligned} f_Z(z) &= \int_0^z \lambda e^{-\lambda x} \lambda e^{-\lambda(z-x)} dx \\ &= \lambda^2 e^{-\lambda z} \int_0^z dx \\ &= \lambda^2 z e^{-\lambda z}, \quad z > 0. \end{aligned}$$

Thus Z has a gamma density with parameters λ and $\alpha = 2$ (or equivalently, Z has a two-stage Erlang distribution). An expression for the reliability of a two-component standby redundant system is obtained by using equation (3.18):

$$\begin{aligned} R(t) &= 1 - F(t) = \sum_{k=0}^1 \frac{(\lambda t)^k}{k!} e^{-\lambda t} \\ &= (1 + \lambda t)e^{-\lambda t}, \quad t \geq 0. \end{aligned} \tag{3.73}$$

Figure 3.38 compares the simplex reliability with a two-component standby system reliability.

#

This example is a special case of the following theorem, which will be proved in Chapter 4.

THEOREM 3.2. If X_1, X_2, \dots, X_r are mutually independent, identically distributed random variables so that $X_i \sim EXP(\lambda)$ for each i , then the random variable $X_1 + X_2 + \dots + X_r$ has an r -stage Erlang distribution with parameter λ .

As a consequence of this theorem, the reliability expression for a standby redundant system with a total of n components, each of which has an exponentially distributed lifetime with parameter λ , is given by

$$R_{\text{standby}}(t) = \sum_{k=0}^{n-1} \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad t \geq 0. \quad (3.74)$$

Example 3.26

Consider a system consisting of n processors. One way to operate the system is to use the concept of standby sparing, and the system reliability will be given by the expression (3.74) above. Note that in such a case only one processor is active at a time, while the others are idle. Now let us consider another way of utilizing the system. In the beginning we let all n processors be active. This is then a parallel redundant system so that its reliability is given by equation (3.66). Given our assumptions, it is easy to show that $R_{\text{standby}}(t) \geq R_{\text{parallel}}(t)$. But the above comparison ignores the fact that the parallel redundant system delivers more computation capacity. Initially when all n processors are active, performing different computations so that the total computing capacity is n (where a unit of computing capacity corresponds to that of one active processor).

Let X_1, X_2, \dots, X_n be the times to failure of the n processors. Then, after a period of time $Y_1 = \min\{X_1, X_2, \dots, X_n\}$, only $n - 1$ processors will be active and the computing capacity of the system will have dropped to $n - 1$. The cumulative computing capacity that the system supplies until all processors have failed is then given by the random variable

$$C_n = nY_1 + (n - 1)(Y_2 - Y_1) + \dots + (n - j)(Y_{j+1} - Y_j) + \dots + (Y_n - Y_{n-1}).$$

From Figure 3.39, we note that C_n is the area under the curve. Beaudry [BEAU 1978] has coined the phrase “computation before failure” for C_n , while Meyer [MEYE 1980] prefers the term “performability.”

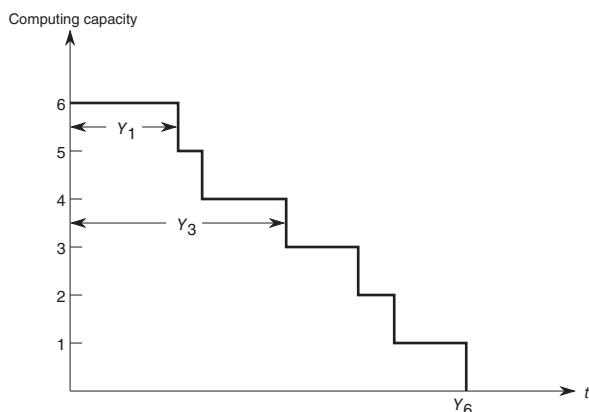


Figure 3.39. Computing capacity as a function of time

In order to obtain the distribution of C_n , we first obtain the distribution of $Y_{j+1} - Y_j$. If we assume processor lifetimes are mutually independent $EXP(\lambda)$ random variables, then we claim that the distribution of $Y_{j+1} - Y_j$ is $EXP[(n-j)\lambda]$. Define $Y_0 = 0$. Then since we know that

$$Y_1 = \min\{X_1, X_2, \dots, X_n\} \sim EXP(n\lambda),$$

our claim holds for $j = 0$. After j processors have failed, the residual lifetimes of the remaining $(n-j)$ processors, denoted by W_1, W_2, \dots, W_{n-j} , are each exponentially distributed with parameter λ because of the memoryless property of the exponential distribution. Note that $Y_{j+1} - Y_j$ is simply the time between the $(j+1)$ st and the j th failure:

$$Y_{j+1} - Y_j = \min\{W_1, W_2, \dots, W_{n-j}\}.$$

It follows that $Y_{j+1} - Y_j \sim EXP((n-j)\lambda)$. Hence, using Example 3.13 we get

$$(n-j)(Y_{j+1} - Y_j) \sim EXP(\lambda).$$

Therefore, C_n is the sum of n independent identically distributed exponential random variables. It follows from Theorem 3.2 that C_n is n -stage Erlang distributed with parameter λ . Now, since the standby redundant system of expression (3.74) has a unit processing capacity while functioning and the total duration of its lifetime is n -stage Erlang with parameter λ , we conclude that the distribution of computation before failure is the same in both modes of operation. (Remember the assumptions behind our model, however.)

‡

Example 3.27

Consider a computer system with job interarrival times that are exponentially distributed with parameter λ . Let X_i be the random variable denoting the time between the $(i-1)$ st and i th arrivals. Then $Z_r = X_1 + X_2 + \dots + X_r$ is the time until the r th arrival and has an r -stage Erlang distribution. Another way to obtain this result is to consider N_t , the number of arrivals in the interval $(0, t]$. As pointed out earlier, N_t has a Poisson distribution with parameter λt . Now the events $[Z_r > t]$ and $[N_t < r]$ are equivalent. Therefore

$$\begin{aligned} P(Z_r > t) &= P(N_t < r) \\ &= \sum_{j=0}^{r-1} P(N_t = j) \\ &= \sum_{j=0}^{r-1} e^{-\lambda t} \left[\frac{(\lambda t)^j}{j!} \right], \end{aligned}$$

which implies that

$$\begin{aligned} F_{Z_r}(t) &= P(Z_r \leq t) \\ &= 1 - \sum_{j=0}^{r-1} \frac{(\lambda t)^j}{j!} e^{-\lambda t}, \end{aligned}$$

which is the r -stage Erlang distribution function.

‡

In Example 3.25 of standby redundancy, we assumed that the failure rates of the two components were the same. Now let the failure rates be distinct; that is, let X and Y be exponentially distributed with parameters λ_1 and λ_2 , respectively.

THEOREM 3.3. If $X \sim EXP(\lambda_1)$, $Y \sim EXP(\lambda_2)$, X and Y are independent, and $\lambda_1 \neq \lambda_2$, then $Z = X + Y$ has a two-stage hypoexponential distribution with parameters λ_1 and λ_2 ; that is, $Z \sim HYPO(\lambda_1, \lambda_2)$.

Proof:

$$\begin{aligned} f_Z(z) &= \int_0^z f_X(x)f_Y(z-x)dx, z > 0 \quad [\text{by equation (3.72)}] \\ &= \int_0^z \lambda_1 e^{-\lambda_1 x} \lambda_2 e^{-\lambda_2(z-x)} dx \\ &= \lambda_1 \lambda_2 e^{-\lambda_2 z} \int_0^z e^{(\lambda_2 - \lambda_1)x} dx \\ &= \lambda_1 \lambda_2 e^{-\lambda_2 z} \left[\frac{e^{-(\lambda_1 - \lambda_2)x}}{-(\lambda_1 - \lambda_2)} \right]_{x=0}^{x=z} \\ &= \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} e^{-\lambda_2 z} + \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 z}. \end{aligned}$$

Comparing this density with equation (3.14), we conclude $X + Y$ has a two-stage hypoexponential distribution with parameters λ_1 and λ_2 .

A more general version of Theorem 3.3 is stated without proof.

THEOREM 3.4. Let $Z = \sum_{i=1}^r X_i$, where X_1, X_2, \dots, X_r are mutually independent and X_i is exponentially distributed with parameter λ_i ($\lambda_i \neq \lambda_j$ for $i \neq j$). Then the density of Z , which is an r -stage hypoexponentially distributed random variable, is given by

$$f_Z(z) = \sum_{i=1}^r a_i \lambda_i e^{-\lambda_i z}, \quad z > 0, \quad (3.75)$$

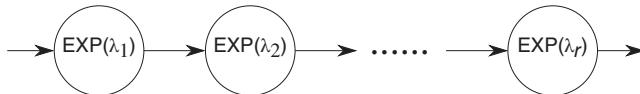


Figure 3.40. Hypoexponential as a series of exponential stages

where

$$a_i = \prod_{\substack{j=1 \\ j \neq i}}^r \frac{\lambda_j}{\lambda_j - \lambda_i}, \quad 1 \leq i \leq r. \quad (3.76)$$

Such a stage type distribution is often visualized as in Figure 3.40.

Another related result is stated by the following corollary.

COROLLARY 3.4. If $X_1 \sim \text{HYPO}(\lambda_1, \lambda_2, \dots, \lambda_k), X_2 \sim \text{HYPO}(\lambda_{k+1}, \dots, \lambda_r)$ and X_1 and X_2 are independent, then $(X_1 + X_2) \sim \text{HYPO}(\lambda_1, \lambda_2, \dots, \lambda_k, \lambda_{k+1}, \dots, \lambda_r)$.

Example 3.28

We have noted that a TMR system has higher reliability than simplex for short missions only. To improve on the reliability of TMR, we observe that after one of the three units has failed, both of the two remaining units have to function properly for the classical TMR configuration to function properly. Thus after one failure, the system reduces to a series system of two components, from the reliability point of view. An improvement over this simple scheme, known as **TMR/simplex**, detects a single component failure, discards the failed component, and reverts to one of the nonfailing simplex components. In other words, not only the failed component but also one of the good components is discarded.

Let X, Y, Z denote the times to failure of the three components. Also let W denote the residual time to failure of the selected surviving component. Let X, Y, Z be mutually independent and exponentially distributed with parameter λ . If L denotes the time to failure of TMR/simplex, then it is clear that

$$L = \min\{X, Y, Z\} + W.$$

Now, since the exponential distribution is memoryless, it follows that the lifetime W of the surviving component is exponentially distributed with parameter λ . Also, from our discussion of order statistics, it follows that $\min\{X, Y, Z\}$ is exponentially distributed with parameter 3λ . Then L has a two-stage hypoexponential distribution with parameters 3λ and λ (using Theorem 3.3). Therefore, using equation (3.15), we have

$$\begin{aligned} F_L(t) &= 1 - \frac{3\lambda}{2\lambda} e^{-\lambda t} + \frac{\lambda}{2\lambda} e^{-3\lambda t}, \quad t \geq 0 \\ &= 1 - \frac{3e^{-\lambda t}}{2} + \frac{e^{-3\lambda t}}{2}. \end{aligned}$$

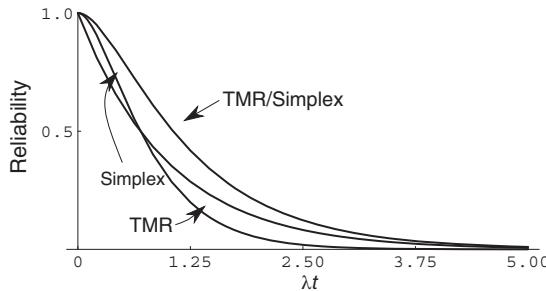


Figure 3.41. Comparison of simplex, TMR, and TMR/simplex reliabilities

Thus the reliability expression of TMR/simplex is given by

$$R(t) = \frac{3 e^{-\lambda t}}{2} - \frac{e^{-3\lambda t}}{2}. \quad (3.77)$$

It is not difficult to see that TMR/simplex has a higher reliability than either a simplex or an ordinary TMR system for all $t \geq 0$. Figure 3.41 compares the simplex reliability with that of TMR and that of TMR/simplex.

#

Example 3.29

Consider a module shown in Figure 3.42, consisting of a functional unit (e.g., an adder) together with an online fault detector (e.g., a modulo-3 checker). Let T and C , respectively, denote the times to failure of the unit and the detector. After the unit fails, it takes a finite time D (called the detection latency) to detect the failure. Failure of the detector, however, is detected instantaneously. Let X denote the time to failure indication and Y denote the time to failure occurrence (of either the detector or the unit). Clearly, $X = \min\{T + D, C\}$ and $Y = \min\{T, C\}$. If the

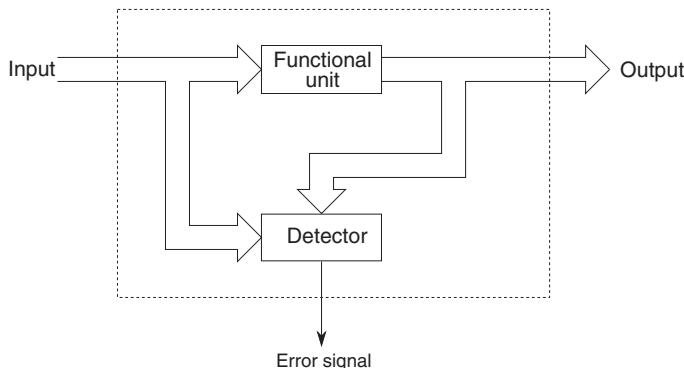


Figure 3.42. A module with an online fault detector

detector fails before the unit, then a false alarm is said to have occurred. If the unit fails before the detector, then the unit keeps producing erroneous output during the detection phase, and thus propagates the effect of the failure. The purpose of the detector is to reduce the detection time D .

We define the real reliability $R_r(t) = P(Y \geq t)$ and the apparent reliability $R_a(t) = P(X \geq t)$. A powerful detector will tend to narrow the gap between $R_r(t)$ and $R_a(t)$.

Assume that T , D , and C are mutually independent and exponentially distributed with parameters λ , δ , and α , respectively. Then, clearly, Y is exponentially distributed with parameter $\lambda + \alpha$ and

$$R_r(t) = e^{-(\lambda+\alpha)t}.$$

Also, $T + D$ is hypoexponentially distributed, so that

$$F_{T+D}(t) = 1 - \frac{\delta}{\delta - \lambda} e^{-\lambda t} + \frac{\lambda}{\delta - \lambda} e^{-\delta t}.$$

Next is the apparent reliability:

$$\begin{aligned} R_a(t) &= P(X \geq t) \\ &= P(\min\{T + D, C\} \geq t) \\ &= P(T + D \geq t \text{ and } C \geq t) \\ &= P(T + D \geq t)P(C \geq t), \text{ by independence} \\ &= [1 - F_{T+D}(t)]e^{-\alpha t} \\ &= \frac{\delta}{\delta - \lambda} e^{-(\lambda+\alpha)t} - \frac{\lambda}{\delta - \lambda} e^{-(\delta+\alpha)t}. \end{aligned}$$

#

Many of the examples in the previous sections can be interpreted as hypo-exponential random variables.

Example 3.30

Consider the TMR system and let X, Y, Z denote the lifetimes of the three components. Assume that these random variables are mutually independent and exponentially distributed with parameter λ . Let L denote the lifetime of the TMR system. Then

$$L = \min\{X, Y, Z\} + \min\{U, V\}$$

Here U and V denote the residual lifetimes of the two surviving components after the first failure. By the memoryless property of the exponential distribution, we conclude that U and V are exponentially distributed with parameter λ . Therefore $\min\{X, Y, Z\}$ has exponential distribution with parameter 3λ and $\min\{U, V\}$

has exponential distribution with parameter 2λ . Therefore, L has hypoexponential distribution with parameters 3λ and 2λ . Then the density of L is

$$\begin{aligned} f_L(t) &= \frac{6\lambda^2}{3\lambda - 2\lambda} e^{-2\lambda t} + \frac{6\lambda^2}{2\lambda - 3\lambda} e^{-3\lambda t} \\ &= 6\lambda e^{-2\lambda t} - 6\lambda e^{-3\lambda t}. \end{aligned}$$

The distribution function of L is

$$\begin{aligned} F_L(t) &= \frac{6}{2}(1 - e^{-2\lambda t}) - \frac{6}{3}(1 - e^{-3\lambda t}) \\ &= 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t}. \end{aligned}$$

Finally, the reliability of TMR is

$$\begin{aligned} R_{\text{TMR}}(t) &= 1 - F_L(t) \\ &= 3e^{-2\lambda t} - 2e^{-3\lambda t}. \end{aligned}$$

This agrees with expression (3.68) derived earlier.

‡

THEOREM 3.5. The order statistic Y_{n-k+1} (of X_1, X_2, \dots, X_n) is hypoexponentially distributed with $(n - k + 1)$ phases, that is

$$Y_{n-k+1} \sim \text{HYPO}[n\lambda, (n - 1)\lambda, \dots, k\lambda]$$

if $X_i \sim EXP(\lambda)$ for each i , and if X_1, X_2, \dots, X_n are mutually independent random variables.

Proof: We prove this theorem by induction. Let $n - k + 1 = 1$; then

$$Y_1 = \min\{X_1, X_2, \dots, X_n\}$$

and clearly $Y_1 \sim EXP(n\lambda)$, which can be interpreted as a one-stage hypoexponential, HYPO($n\lambda$). Next assume that Y_{n-j+1} is hypoexponentially distributed with parameters $n\lambda, (n - 1)\lambda, \dots, j\lambda$. It is clear that

$$Y_{n-j+2} = Y_{n-j+1} + \min\{W_{n-j+2}, \dots, W_n\},$$

where the W_i ($n - j + 2 \leq i \leq n$) denote the residual lifetimes of the surviving components. By the memoryless property of the exponential distribution, W_i has exponential distribution with parameter λ . Therefore

$$\min\{W_{n-j+2}, \dots, W_n\}$$

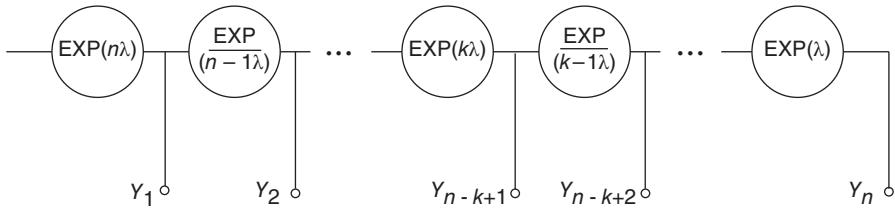


Figure 3.43. The order statistics of the exponential distribution

has an exponential distribution with the following parameter

$$[n - (n - j + 2) + 1]\lambda = (j - 1)\lambda.$$

The proof of the theorem then follows using Corollary 3.4. The result of the theorem may be visualized as in Figure 3.43.

Example 3.31

Consider a k -out-of- n system, each of whose components follows an exponential failure law with parameter λ . Then the lifetime of the system is given by $L(k|n) = Y_{n-k+1}$ and is hypoexponentially distributed with parameters $n\lambda, (n-1)\lambda, \dots, k\lambda$. The density of $L(k|n)$ is given by

$$f(t) = \sum_{i=k}^n a_i \lambda_i e^{-\lambda_i t},$$

where $\lambda_i = i\lambda$, and

$$\begin{aligned} a_i &= \prod_{\substack{j=k \\ j \neq i}}^n \frac{\lambda_j}{\lambda_j - \lambda_i} \\ &= \prod_{\substack{j=k \\ j \neq i}}^n \frac{j}{j - i} \\ &= \frac{k(k+1)\dots(i-1)(i+1)\dots(n-1)n}{(k-i)\dots(-1)(1)\dots(n-i)} \\ &= (-1)^{i-k} \frac{(i-1)!n!}{(k-1)!(i-k)!i!(n-i)!} \\ &= (-1)^{i-k} \binom{n}{i} \binom{i-1}{k-1}. \end{aligned}$$

Then

$$\begin{aligned} R(t) &= \sum_{i=k}^n a_i e^{-\lambda_i t} \\ &= \sum_{i=k}^n \binom{n}{i} \binom{i-1}{k-1} (-1)^{i-k} e^{-i\lambda t}. \end{aligned} \quad (3.78)$$

[Note that by substituting $k = 1$ in (3.78), we get (3.66).] It can be verified [see problem 4 at the end of this section], using a set of combinatorial identities, that expression (3.78) is equivalent to

$$\sum_{i=k}^n \binom{n}{i} e^{-i\lambda t} (1 - e^{-\lambda t})^{n-i}$$

as derived earlier.

#

Example 3.32

Consider a hybrid k -out-of- n system with $n + m$ components, n of which are initially put into operation with the remaining m components in a deenergized standby status. An active component has an exponential failure law with parameter λ . Unlike our earlier examples, we assume that a component can fail in a deenergized state with a constant failure rate μ (presumably $0 \leq \mu \leq \lambda$). This is sometimes known as a “warm spare”. Let X_i ($1 \leq i \leq n$) denote the lifetime of an energized component and let Y_j ($1 \leq j \leq m$) denote the lifetime of a deenergized component. Then the system lifetime $L(k|n, m)$ is given by

$$\begin{aligned} L(k|n, m) &= \min(X_1, X_2, \dots, X_n; Y_1, Y_2, \dots, Y_m) + L(k|n, m-1) \\ &= W(n, m) + L(k|n, m-1). \end{aligned}$$

This follows since $W(n, m)$ is the time to first failure among the $n + m$ components and after the removal of the failed component, the system has n energized and $m - 1$ deenergized components. Note that these $n + m - 1$ components have not aged by the exponential assumption. Therefore

$$L(k|n, m) = L(k|n, 0) + \sum_{i=1}^m W(n, i). \quad (3.79)$$

Here $L(k|n, 0) = L(k|n)$ is simply the lifetime of an k -out-of- n system and is therefore the $(n - k + 1)$ th-order statistic as shown in Example 3.31. The distribution of $L(k|n, 0)$ is therefore an $(n - k + 1)$ -phase hypoexponential with parameters $n\lambda, (n-1)\lambda, \dots, k\lambda$. Also, $W(n, i)$ has an exponential distribution with parameter $n\lambda + i\mu$. Then, using Corollary 3.4, we conclude that $L(k|n, m)$ has an $(n + m - k + 1)$ -stage hypoexponential distribution with parameters $n\lambda + m\mu, n\lambda + (m-1)\mu, \dots, n\lambda + \mu, n\lambda, (n-1)\lambda, \dots, k\lambda$. This can be visualized as in Figure 3.44.

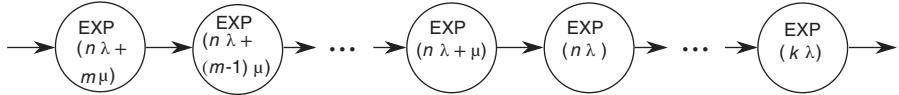


Figure 3.44. Lifetime distribution of a hybrid k -out-of- n system

Let $R_{[k|n,m]}(t)$ denote the reliability of such a system; then

$$R_{[k|n,m]}(t) = \sum_{i=1}^m a_i e^{-(n\lambda+i\mu)t} + \sum_{i=k}^n b_i e^{-i\lambda t}, \quad (3.80)$$

where

$$a_i = \prod_{\substack{j=1 \\ j \neq i}}^m \frac{n\lambda + j\mu}{j\mu - i\mu} \prod_{j=k}^n \frac{j\lambda}{j\lambda - n\lambda - i\mu}, \quad (3.81)$$

and

$$b_i = \prod_{j=1}^m \frac{n\lambda + j\mu}{(n-i)\lambda + j\mu} \prod_{\substack{j=k \\ j \neq i}}^n \frac{j\lambda}{j\lambda - i\lambda}. \quad (3.82)$$

Letting $\rho = \lambda/\mu$, we obtain

$$\begin{aligned} a_i &= \frac{(n\rho + m) \dots (n\rho + 1)}{(n\rho + i)(m - i) \dots (1)(-1) \dots (1 - i)} \\ &\quad \frac{(-1)^{n-k+1} n(n-1) \dots k}{\left(\frac{i}{\rho} + n - k\right) \dots \left(\frac{i}{\rho} + 1\right) \left(\frac{i}{\rho}\right)} \\ &= (-1)^{i-1} \frac{(n\rho + m)! m! i!}{(n\rho + i)(n\rho)! s!(s-i)! i!} \\ &\quad \frac{(-1)^{n-k+1} n(n-1)! \left(\frac{i}{\rho}\right)! (n-k)!}{\frac{i}{\rho} (k-1)! \left[\left(\frac{i}{\rho}\right) n - k\right]! (n-k)!} \\ &= (-1)^{n-k+i} \frac{\binom{n\rho+m}{m} \binom{m}{i} \binom{n-1}{k-1}}{\left[1 + \frac{i}{n\rho}\right] \left(\frac{i}{\rho} + n - k\right)}. \end{aligned} \quad (3.83)$$

[Note that if ρ is not an integer, we use the generalized definition of factorial above. Thus, for a real number α , $\alpha! = \Gamma(\alpha + 1)$.]

Similarly

$$\begin{aligned}
 b_i &= \frac{(n\rho + m) \dots (n\rho + 1)}{[(n-i)\rho + m] \dots [(n-i)\rho + 1]} \\
 &\quad \frac{n \dots k}{i[(n-i) \dots (1)(-1) \dots (k-i)]} \\
 &= \frac{(n\rho + m)![(n-i)\rho]!n!(-1)^{i-k}}{(n\rho)![(n-i)\rho + m]!i(k-1)!(n-i)!(i-k)!} \\
 &= (-1)^{i-k} \frac{(n\rho + m)!m!((n-i)\rho)!n!i!k}{m!(n\rho)![(n-i)\rho + m]!(n-i)!i!(i-k)!k!i} \\
 &= (-1)^{i-k} \frac{\binom{n\rho + m}{m} \binom{n}{i} \binom{i}{k}}{\frac{i}{k} \binom{(n-i)\rho + m}{m}}. \tag{3.84}
 \end{aligned}$$

Once again, using combinatorial identities, we can verify that our expression for hybrid k -out-of- n reliability matches with that given by Mathur and Avizienis [MATH 1970].

#

Problems

- Given n random numbers u_1, u_2, \dots, u_n , derive an expression for a random deviate of an n -stage hypoexponential distribution with parameters $\lambda_1, \lambda_2, \dots, \lambda_n$.
- Compare the TMR/simplex reliability with two-component and three-component redundant systems having standby redundancy. Graph the expressions on the same plot.
- Repeat problem 2 for two and three component parallel redundant systems.
- Show that the reliability expression (3.78) for k -out-of- n system reliability reduces to the expression

$$\sum_{i=k}^n \binom{n}{i} e^{-i\lambda t} (1 - e^{-\lambda t})^{n-i}.$$

- Using equation (3.80) obtain an explicit expression for the reliability of a hybrid TMR system with one spare. Compare the reliability of this system with those of a TMR system and a simplex system by plotting. Use $\lambda = 1/10,000 \text{ h}^{-1}$ and $\mu = 1/100,000 \text{ h}^{-1}$.
- Compare (by plotting) reliability expressions for the simplex system, the two-component parallel redundant system, and the two-component standby redundant system. Assume that the failure rate of an active component is constant at $1/10,000 \text{ h}^{-1}$, the failure rate of a spare is zero, and that the switching mechanism is fault-free.

3.9 FUNCTIONS OF NORMAL RANDOM VARIABLES

The normal distribution has great importance in mathematical statistics because of the central-limit theorem alluded to earlier. This distribution also plays an important role in communication and information theory. We will now study distributions derivable from the normal distribution. The use of most of these distributions will be deferred until Chapters 10 and 11.

THEOREM 3.6. Let X_1, X_2, \dots, X_n be mutually independent random variables such that $X_i \sim N(\mu_i, \sigma_i^2)$, $i = 1, 2, \dots, n$. Then $S_n = \sum_{i=1}^n X_i$ is normally distributed, that is, $S_n \sim N(\mu, \sigma^2)$, where

$$\mu = \sum_{i=1}^n \mu_i \quad \text{and} \quad \sigma^2 = \sum_{i=1}^n \sigma_i^2.$$

Owing to this theorem, we say that the normal distribution has the **reproductive** property. A proof of this theorem will be given in Chapter 4. The theorem can be further generalized as in problem 1 at the end of Section 4.4, so that if X_1, X_2, \dots, X_n are mutually independent random variables with $X_i \sim N(\mu_i, \sigma_i^2)$, $i = 1, 2, \dots, n$ and a_1, a_2, \dots, a_n are real constants, then $Y_n = \sum_{i=1}^n a_i X_i$ is normally distributed; that is, $Y_n \sim N(\mu, \sigma^2)$, where

$$\mu = \sum_{i=1}^n a_i \mu_i \quad \text{and} \quad \sigma^2 = \sum_{i=1}^n a_i^2 \sigma_i^2.$$

In particular, if we let $n = 2$, $a_1 = +1$, and $a_2 = -1$, then we conclude that the difference $Y = X_1 - X_2$ of two independent normal random variables $X_1 \sim N(\mu_1, \sigma_1^2)$ and $X_2 \sim N(\mu_2, \sigma_2^2)$ is normally distributed, that is, $Y \sim N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$.

Example 3.33

It has been empirically determined that the memory requirement of a program (called the **working-set size** of the program) is approximately normal. In a multiprogramming system, the number of programs sharing the main memory simultaneously (called the **degree of multiprogramming**) is found to be n . Now if X_i denotes the working-set size of the i th program with $X_i \sim N(\mu_i, \sigma_i^2)$, then it follows that the sum total memory demand, S_n , of the n programs is normally distributed with parameters $\mu = \sum_{i=1}^n \mu_i$ and $\sigma^2 = \sum_{i=1}^n \sigma_i^2$.

#

Example 3.34

A sequence of independent, identically distributed random variables, X_1, X_2, \dots, X_n , is known in mathematical statistics as a *random sample* of size n . In many problems of statistical sampling theory, it is reasonable to

assume that the underlying distribution is the normal distribution. Thus let $X_i \sim N(\mu, \sigma^2)$, $i = 1, 2, \dots, n$. Then from Theorem 3.6, we obtain

$$S_n = \sum_{i=1}^n X_i \sim N(n\mu, n\sigma^2)$$

One important function known as the sample mean is quite useful in problems of statistical inference. *Sample mean* \bar{X} is given by

$$\bar{X} = \frac{S_n}{n} = \sum_{i=1}^n \frac{X_i}{n}. \quad (3.85)$$

To obtain the pdf of the sample mean \bar{X} , we use equation (3.55) to obtain

$$f_{\bar{X}} = n f_{S_n}(nx).$$

But since $S_n \sim N(n\mu, n\sigma^2)$, we have

$$\begin{aligned} f_{\bar{X}}(x) &= n \frac{1}{\sqrt{2\pi}(\sqrt{n}\sigma)} e^{-\frac{(nx-n\mu)^2}{2n\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi}[\sigma(n)^{-1/2}]} e^{-\frac{(x-\mu)^2}{2(\sigma^2/n)}}, \quad -\infty < x < \infty. \end{aligned}$$

It follows that $\bar{X} \sim N(\mu, \sigma^2/n)$. Similarly, it can be shown that the random variable $(\bar{X} - \mu)\sqrt{n}/\sigma$ has the standard normal distribution, $N(0, 1)$.

‡

If X is $N(0, 1)$, we know from Example 3.9 that $Y = X^2$ is gamma-distributed with $Y \sim \text{GAM}(\frac{1}{2}, \frac{1}{2})$, which is the chi-square distribution with one degree of freedom. Now consider X_1, X_2 that are independent standard normal random variables and $Y = X_1^2 + X_2^2$.

Example 3.35

If $X_1 \sim N(0, 1)$, $X_2 \sim N(0, 1)$ and X_1 and X_2 are independent, then $Y = X_1^2 + X_2^2$ is exponentially distributed so that $Y \sim EXP(\frac{1}{2})$.

To see this, we obtain the distribution function of Y :

$$\begin{aligned} F_Y(y) &= P(X_1^2 + X_2^2 \leq y) \\ &= \int \int_{x_1^2 + x_2^2 \leq y} f_{X_1 X_2}(x_1, x_2) dx_1 dx_2. \end{aligned}$$

Note that the surface of integration is a circular area about the origin with the radius \sqrt{y} (see Figure 3.45). Using the fact that X_1 and X_2 are independent, and standard normal, we have

$$F_Y(y) = \int \int_{x_1^2 + x_2^2 \leq y} \frac{1}{2\pi} e^{-\frac{x_1^2+x_2^2}{2}} dx_1 dx_2.$$

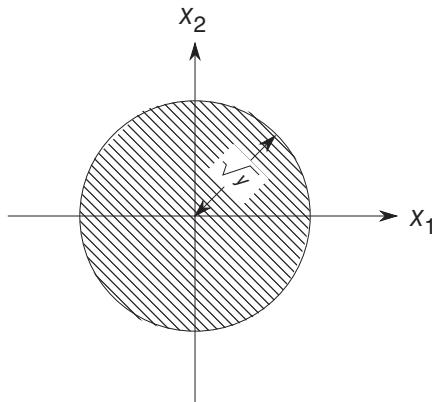


Figure 3.45. The area of integration for Example 3.35

Making a change of variables (to polar coordinates), $x_1 = r \cos \theta$, $x_2 = r \sin \theta$, so that $r^2 = x_1^2 + x_2^2$ and $\theta = \tan^{-1}(x_2/x_1)$, we have

$$\begin{aligned} F_Y(y) &= \int_{\theta=0}^{2\pi} \int_{r=0}^{\sqrt{y}} \frac{r}{2\pi} e^{-r^2/2} dr d\theta \\ &= \begin{cases} 1 - e^{-y/2}, & y > 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Therefore, Y is exponentially distributed with parameter $\frac{1}{2}$. ‡

This example is a special case of the following theorem.

THEOREM 3.7. If X_1, X_2, \dots, X_n is a sequence of mutually independent, standard normal random variables, then

$$Y = \sum_{i=1}^n X_i^2$$

has the gamma distribution, $\text{GAM}(\frac{1}{2}, n/2)$, or the chi-square distribution with n degrees of freedom, X_n^2 .

This theorem follows from the reproductive property of the gamma distribution (see Theorem 3.8).

THEOREM 3.8. Let X_1, X_2, \dots, X_n be a sequence of mutually independent gamma random variables such that $X_i \sim \text{GAM}(\lambda, \alpha_i)$ for $i = 1, 2, \dots, n$. Then $S_n = \sum_{i=1}^n X_i$ has the gamma distribution $\text{GAM}(\lambda, \alpha)$, where $\alpha = \alpha_1 + \alpha_2 + \dots + \alpha_n$.

This theorem will be proved in Chapter 4.

Since $X_n^2 \sim \text{GAM}(\frac{1}{2}, n/2)$, we have the following corollary.

COROLLARY 3.8. Let Y_1, Y_2, \dots, Y_n be mutually independent chi-square random variables such that $Y_i \sim X_{k_i}^2$. Then $Y_1 + Y_2 + \dots + Y_n$ has the X_k^2 distribution, where

$$k = \sum_{i=1}^n k_i.$$

Example 3.36

Assume that X_1, X_2, \dots, X_n are mutually independent, identically distributed normal random variables such that $X_i \sim N(\mu, \sigma^2)$, $i = 1, 2, \dots, n$. It follows that $Z_i = (X_i - \mu)/\sigma$ is standard normal. Thus Z_1, Z_2, \dots, Z_n are independent standard normal random variables. Hence, using Theorem 3.7, we have

$$Y = \sum_{i=1}^n Z_i^2 = \sum_{i=1}^n \frac{(X_i - \mu)^2}{\sigma^2}, \quad (3.86)$$

which has the chi-square distribution with n degrees of freedom. Note that the random variable $\sum_{i=1}^n (X_i - \mu)^2/n$ may be used as an estimator of the parameter σ^2 .

‡

Example 3.37

In the last example, we suggested that $\sum_{i=1}^n (X_i - \mu)^2/n$ may be used as an estimator of the parameter σ^2 assuming that X_1, X_2, \dots, X_n are independent observations from a normal distribution $N(\mu, \sigma^2)$. However, this expression assumes that the parameter μ of the distribution is already known. This is rarely the case in practice, and the sample mean $\bar{X} = \sum_{i=1}^n X_i/n$ is usually substituted in its place. Thus, the random variable

$$U = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1} \quad (3.87)$$

is usually used as an estimator of the parameter σ^2 and is often denoted by S^2 . (The reason for the value $n-1$ rather than n in the denominator will be seen in Chapter 10).

Rewriting, we have

$$S^2 = U = \frac{\sigma^2}{n-1} \sum_{i=1}^n \left(\frac{X_i - \bar{X}}{\sigma} \right)^2, \quad (3.88)$$

where we note that n random variables $\{(X_i - \bar{X})/\sigma | 1 \leq i \leq n\}$ satisfy the relation

$$\sum_{i=1}^n \frac{X_i - \bar{X}}{\sigma} = 0 \quad (3.89)$$

(from the definition of the sample mean, \bar{X}). Thus they are linearly dependent. It can be shown that the random variable

$$W = \sum_{i=1}^n \left(\frac{X_i - \bar{X}}{\sigma} \right)^2 \quad (3.90)$$

can be transformed to a sum of squares of $(n - 1)$ independent standard normal random variables, and hence $W = (n - 1)U/\sigma^2 = (n - 1)S^2/\sigma^2$ has a chi-square distribution with $n - 1$ degrees of freedom (rather than n degrees of freedom). \ddagger

Just as the sums of chi-square random variables are of interest, so is the ratio of two chi-square random variables. First assume that X and Y are independent, positive-valued random variables and let Z be their quotient:

$$Z = \frac{Y}{X}. \quad (3.91)$$

Then the distribution function of Z is obtained using the formula

$$F_Z(z) = \int \int_{A_z} f(x, y) \, dx \, dy,$$

where the set

$$A_z = \{(x, y) | y/x \leq z\}$$

is shown in Figure 3.46. Therefore

$$\begin{aligned} F_Z(z) &= \int_0^\infty \left[\int_0^{xz} f(x, y) \, dy \right] \, dx \\ &= \int_0^\infty \left[\int_0^z x \, f(x, xv) \, dv \right] \, dx, \end{aligned} \quad (3.92)$$

after a change of variables to $y = xv$.

It follows that the pdf of Z is given by

$$\begin{aligned} f_Z(z) &= \int_0^\infty x \, f(x, xz) \, dx \\ &= \int_0^\infty x f_X(x) f_Y(xz) \, dx, \quad 0 < z < \infty \end{aligned} \quad (3.93)$$

(by independence of X and Y).

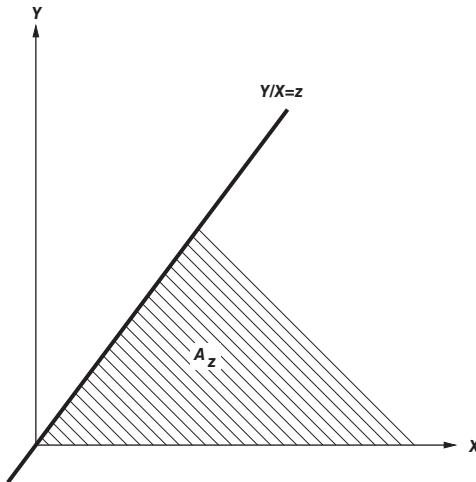


Figure 3.46. The area of integration for computing the CDF of $Y/X = Z$

THEOREM 3.9. Let Y_1 and Y_2 be independent random variables with $X_{n_1}^2$ and $X_{n_2}^2$ distributions, respectively. Then

$$Z = \frac{Y_1/n_1}{Y_2/n_2}$$

has the F distribution, which is characterized by two parameters, (n_1, n_2) , that is, $Z \sim F_{n_1, n_2}$. The pdf of Z is given by

$$f_Z(z) = \begin{cases} \frac{(n_1/n_2)\Gamma[(n_1+n_2)/2](n_1z/n_2)^{(n_1/2)-1}}{\Gamma(n_1/2)\Gamma(n_2/2)[1+(n_1z/n_2)]^{(n_1+n_2)/2}}, & z > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.94)$$

Proof: Recall that

$$f_{Y_1}(y_1) = \frac{y_1^{(n_1/2)-1} e^{-y_1/2}}{2^{n_1/2} \Gamma(n_1/2)}$$

and

$$f_{Y_2}(y_2) = \frac{y_2^{(n_2/2)-1} e^{-y_2/2}}{2^{n_2/2} \Gamma(n_2/2)}.$$

Let $Y = Y_1/n_1$ and $X = Y_2/n_2$. Using formula (3.55), it follows that

$$f_Y(y) = \frac{n_1(yn_1)^{n_1/2-1}e^{-(n_1y)/2}}{2^{n_1/2}\Gamma(n_1/2)}$$

and

$$f_X(x) = \frac{n_2(xn_2)^{n_2/2-1}e^{-(n_2x)/2}}{2^{n_2/2}\Gamma(n_2/2)}.$$

Now, applying equation (3.93), we get

$$\begin{aligned} f_Z(z) &= \int_0^\infty x \frac{n_1 n_2}{2^{(n_1+n_2)/2}\Gamma(n_1/2)\Gamma(n_2/2)} \\ &\quad \cdot (xz n_1)^{n_1/2-1} (xn_2)^{n_2/2-1} e^{-(n_1 xz + n_2 x)/2} dx \\ &= \frac{n_1 n_2 (n_1)^{n_1/2-1} (n_2)^{n_2/2-1} z^{n_1/2-1}}{2^{(n_1+n_2)/2}\Gamma(n_1/2)\Gamma(n_2/2)} \\ &\quad \cdot \int_0^\infty x^{n_1/2+n_2/2-1} e^{-x(n_1 z + n_2)/2} dx. \end{aligned} \quad (3.95)$$

Using equation (3.25), the last integral is evaluated as

$$\frac{\Gamma[(n_1 + n_2)/2]}{[(n_1 z + n_2)/2]^{(n_1 + n_2)/2}}.$$

Substituting this in (3.95), we get the required result as in (3.94).

Example 3.38

Suppose that $X_1, X_2, \dots, X_m, X_{m+1}, \dots, X_n$ are mutually independent normal random variables with the common distribution, $N(0, \sigma^2)$. Then by Theorem 3.7

$$Y = \sum_{i=1}^m \frac{X_i^2}{\sigma^2} \quad \text{and} \quad X = \sum_{i=m+1}^n \frac{X_i^2}{\sigma^2}$$

are chi-square distributed with m and $(n - m)$ degrees of freedom, respectively. Furthermore, X and Y are independent. It follows by Theorem 3.9 that

$$Z = \frac{\sum_{i=1}^m X_i^2/m}{\sum_{i=m+1}^n X_i^2/(n-m)} \quad (3.96)$$

has the $F_{m,n-m}$ distribution. ‡

The last distribution we introduce here is Student's t distribution.

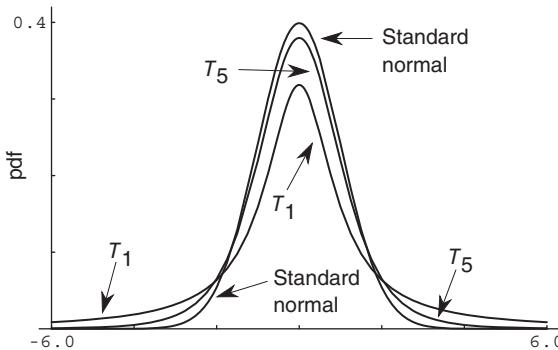


Figure 3.47. Student's t pdf and its comparison with standard normal pdf

THEOREM 3.10. If V and W are independent random variables such that $V \sim N(0, 1)$ and $W \sim X_n^2$, then the random variable

$$T = \frac{V}{\sqrt{W/n}} \quad (3.97)$$

has the t distribution with n degrees of freedom. The pdf of this random variable is given by

$$f_T(t) = \frac{\Gamma[(n+1)/2]}{\sqrt{n\pi}\Gamma[n/2]} \left[1 + \frac{t^2}{n}\right]^{-(n+1)/2}, \quad -\infty < t < \infty. \quad (3.98)$$

For $n = 1$, this pdf reduces to

$$f_T(t) = \frac{1}{\pi(1+t^2)}, \quad (3.99)$$

which is known as the **Cauchy pdf**.

The pdf in (3.98) is plotted for various degrees of freedom in Figure 3.47. It may be noted that as n approaches infinity, the t distribution approaches the normal distribution.

Example 3.39

Assume that X_1, X_2, \dots, X_n are mutually independent identically distributed normal random variables such that $X_i \sim N(\mu, \sigma^2)$. Then from Example 3.34, it follows that

$$V = \frac{(\bar{X} - \mu)\sqrt{n}}{\sigma} \quad (3.100)$$

has the standard normal distribution. Also, from Example 3.37

$$\frac{(n-1)S^2}{\sigma^2} = W = \sum_{i=1}^n \left[\frac{X_i - \bar{X}}{\sigma} \right]^2 \quad (3.101)$$

has the X_{n-1}^2 distribution. It follows that

$$\begin{aligned} T &= \frac{V}{\sqrt{\frac{W}{(n-1)}}} = \frac{(\bar{X} - \mu)\sqrt{n}/\sigma}{\left[S\frac{\sqrt{n-1}}{\sigma}\right]} \cdot \sqrt{n-1} \\ &= \frac{\bar{X} - \mu}{S/\sqrt{n}} \end{aligned} \quad (3.102)$$

has the t distribution with $(n-1)$ degrees of freedom.

‡

Problems

1. In communication theory, waveforms of the form

$$A(t) = x(t) \cos(\omega t) - y(t) \sin(\omega t)$$

appear quite frequently. At a fixed time instant, $t = t_1$, $X = X(t_1)$, and $Y = Y(t_1)$ are known to be independent Gaussian random variables, specifically, $N(0, \sigma^2)$. Show that the distribution function of the envelope $Z = \sqrt{X^2 + Y^2}$ is given by

$$F_Z(z) = \begin{cases} 1 - e^{-z^2/2\sigma^2}, & z > 0, \\ 0, & \text{otherwise.} \end{cases}$$

This distribution is called the **Rayleigh distribution**. Compute and plot its pdf.

2. The test effort during the software testing phase, such as human resources, the number of test cases, and CPU time, can be measured by the cumulative amount of testing effort during the time interval $(0, t]$. Yamada et al. [YAMA 1986] proposed a formula for $W(t)$: $dW(t)/dt = g(t)(1 - W(t))$ where $g(t)$ is the instantaneous consumption rate of the testing effort expenditures. $W(t)$ is defined as $W(t) = \int_0^t w(t)dt$ where $w(t)$ is the testing-effort consumption rate at time t . Find an explicit expression for $W(t)$ in terms of $g(t)$ and show that $W(t)$ is the Rayleigh distribution.
3. A calculator operates on two 1.5-V batteries (for a total of 3 V). The actual voltage of a battery is normally distributed with $\mu = 1.5$ and $\sigma^2 = 0.45$. The tolerances in the design of the calculator are such that it will not operate satisfactorily if the total voltage falls outside the range 2.70–3.30 V. What is the probability that the calculator will function correctly?

Review Problems

1. Show that the pdf of the product $Z = XY$ of two independent random variables with respective densities f_X and f_Y is given by

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{1}{|x|} f_X(x) f_Y\left(\frac{z}{x}\right) dx.$$

2. * Consider the problem of multiplying the mantissas X and Y of two floating-point numbers in base β [HAMM 1973]. Assume that X and Y are independent random variables with densities f_X and f_Y , respectively, and $1/\beta \leq X, Y < 1$. Note that the product XY satisfies $1/\beta^2 \leq XY < 1$. If $1/\beta^2 \leq XY < 1/\beta$, then a left shift is required in order to normalize the result. Let Z_N denote the normalized product (i.e., $Z_N = XY$ if $XY \geq 1/\beta$ and $Z_N = \beta^{-1}XY$ otherwise). Show that the pdf of Z_N is given by

$$f_{Z_N}(z) = \frac{1}{\beta} \int_{1/\beta}^z \frac{f_X(x)}{x} f_Y\left(\frac{z}{\beta x}\right) dx + \int_z^1 \frac{f_X(x)}{x} f_Y\left(\frac{z}{x}\right) dx, \frac{1}{\beta} \leq z < 1.$$

Assuming that $f_Y(y) = 1/(y \ln \beta)$, show that Z_N also has the same reciprocal density. Thus, in a long sequence of multiplications, if at least one factor has the reciprocal density, then the normalized product has the reciprocal density. Assuming that both X and Y have the reciprocal density, compute the probability that a left shift is required for normalization.

3. * Consider the quotient Y/X of two independent normalized floating-point mantissas in base β [HAMM 1973]. Since $1/\beta \leq Y/X < \beta$, a one-digit right shift may be required to obtain the normalized quotient Q_N . Show that the pdf of Q_N is given by

$$f_{Q_N}(z) = \frac{1}{z^2} \int_{1/\beta}^z x f_X(x) f_Y\left(\frac{x}{z}\right) dx + \frac{1}{\beta z^2} \int_z^1 x f_X(x) f_Y\left(\frac{x}{\beta z}\right) dx, \frac{1}{\beta} \leq z < 1.$$

Show that if the dividend Y has the reciprocal density, then the normalized quotient also has the same density. Also compute the probability that a shift is required assuming that both X and Y have the reciprocal density.

REFERENCES

- [ASH 1970] R. B. Ash, *Basic Probability Theory*, Wiley, New York, 1970.
- [BARL 1981] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models*, 2nd print, To Begin With, Silver Spring, MD, 1981.
- [BEAU 1978] M. D. Beaudry, “Performance related reliability for computing systems,” *IEEE Trans. Comput.*, 540–547 (June 1978).
- [BHAT 1984] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd ed., Wiley, New York, 1984.
- [BLAK 1989] J. T. Blake and K. S. Trivedi, “Multistage interconnection network reliability,” *IEEE Trans. Comput.*, C-38(11), 1600–1604 (Nov. 1989).
- [BREI 1968] L. Breiman, *Probability*, Addison-Wesley, Reading, MA, 1968.
- [CLAR 1978] J. Clary, A. Jai, S. Weikel, R. Saeks, and D. P. Siewiorek, *A Preliminary Study of Built-in-Test for the Military Computer Family*, Technical Report, Research Triangle Institute, Research Triangle Park, NC, 1978.

- [CROV 1997] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic evidence and possible causes," *IEEE/ACM Trans. Networking*, 835–846 (Dec. 1997).
- [DENG 1996] S. Deng, "Empirical model of WWW document arrivals at access link", *Proceedings of the 1996 IEEE International Conference on Communications*, June 1996, pp. 1797–1802.
- [FISH 1995] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York, 1995.
- [HAMM 1973] R. W. Hamming, *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1973.
- [IBM 1997] IBM OEM hard disk drive specifications for DTCA-23240/24090, *2.5-Inch Hard Disk Drive with ATA Interface*, revision 3.0.
- [KNUT 1997] D. E. Knuth, *The Art of Computer Programming*, Vol 2, *Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Reading, MA, 1997.
- [KOSL 2001] J. Koslov, Sun Microsystems Inc., personal communication, 2001.
- [LEE 1993] I. Lee, D. Tang, R. K. Iyer and M.-C. Hsueh, "Measurement-based evaluation of operating system fault tolerance", *IEEE Trans. Reliability*, **42**(2), 238–249 (June 1993).
- [LELA 1986] W. Leland and T. Ott, "Load-balancing heuristics and process behavior," *PERFORMANCE'86 and ACM SIGMETRICS Joint Conf. Computer Performance Modelling, Measurement and Evaluation*, Raleigh, NC, May 1986, pp. 54–69.
- [MATH 1970] F. P. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid redundant digital system: Generalized triple modular redundancy with self-repair," *AFIPS Conf. Proc. Spring Joint Computer Conf.*, Vol. **36**, 1970, pp. 375–383.
- [MEYE 1980] J. F. Meyer, D. G. Furchtgott, and L. T. Wu, "Performability evaluation of the SIFT computer," *IEEE Trans. Comput.*, 501–509 (June 1980).
- [NABE 1998] M. Nabe, M. Murata, and H. Miyahara, "Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines," *Performance Evaluation*, **34**(4), 249–271 (1998).
- [PAXS 1995] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, 226–244 (June, 1995).
- [RAO 1989] T. R. Rao and E. Fujiwara, *Error Control Coding for Computer Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [RUDI 1964] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, 1964.
- [SAHN 1996] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer System: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.
- [YAMA 1986] S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing effort," *IEEE Trans. Reliability*, **R-35**(1), 19–23 (1986).

Chapter 4

Expectation

4.1 INTRODUCTION

The distribution function $F(x)$ or the density $f(x)$ [pmf $p(x_i)$ for a discrete random variable] completely characterizes the behavior of a random variable X . Frequently, however, we need a more concise description such as a single number or a few numbers, rather than an entire function. One such number is the **expectation** or the **mean**, denoted by $E[X]$. Similarly, the **median**, which is defined as any number x such that $P(X < x) \leq \frac{1}{2}$ and $P(X > x) \leq \frac{1}{2}$, and the **mode**, defined as the number x , for which $f(x)$ or $p(x_i)$ attains its maximum, are two other quantities sometimes used to describe a random variable X . The mean, median, and mode are often called **measures of central tendency** of a random variable X .

Definition (Expectation). The expectation, $E[X]$, of a random variable X is defined by

$$E[X] = \begin{cases} \sum_i x_i p(x_i), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} xf(x) dx, & \text{if } X \text{ is continuous,} \end{cases} \quad (4.1)$$

provided the relevant sum or integral is absolutely convergent; that is, $\sum_i |x_i| p(x_i) < \infty$ and $\int_{-\infty}^{\infty} |x| f(x) dx < \infty$. If the right-hand side in (4.1) is not absolutely convergent, then $E[X]$ does not exist. Most common random variables have finite expectation; however, problem 1 at the end of this section provides an example of a random variable whose expectation does not

exist. Definition (4.1) can be extended to the case of mixed random variables through the use of Riemann–Stieltjes integral. Alternatively, the formula given in problem 2 at the end of this section can be used in the general case.

Example 4.1

Consider the problem of searching for a specific name in a table of names. A simple method is to scan the table sequentially, starting from one end, until we either find the name or reach the other end, indicating that the required name is missing from the table. The following is a C program fragment for sequential search:

```
#define n 100

Example() {
    NAME Table[n+1];
    NAME myName;
    int I;

    /* myName has been initialized elsewhere */
    Table[0] = myName; //Table[0] is used as a sentinel or marker.
    I = n;

    while (myName != Table[I])
        I = I - 1;
    if (I > 0) {
        printf("found!");
        myName = Table[I];
    }
    else
        printf("not found!");
}
```

In order to analyze the time required for sequential search, let X be the discrete random variable denoting the number of comparisons “ $myName \neq Table[I]$ ” made. Clearly, the set of all possible values of X is $\{1, 2, \dots, n+1\}$, and $X = n+1$ for unsuccessful searches. Since the value of X is fixed for unsuccessful searches, it is more interesting to consider a random variable Y that denotes the number of comparisons on a successful search. The set of all possible values of Y is $\{1, 2, \dots, n\}$. To compute the average search time for a successful search, we must specify the pmf of Y . In the absence of any specific information, it is natural to assume that Y is uniform over its range:

$$p_Y(i) = \frac{1}{n}, \quad 1 \leq i \leq n.$$

Then

$$E[Y] = \sum_{i=1}^n i p_Y(i) = \frac{1}{n} \frac{n(n+1)}{2} = \frac{(n+1)}{2}.$$

Thus, on the average, approximately half the table needs to be searched.

#

Example 4.2

The assumption of discrete uniform distribution, used in Example 4.1, rarely holds in practice. It is possible to collect statistics on access patterns and use empirical distributions to reorganize the table so as to reduce the average search time. Unlike Example 4.1, we now assume for convenience that table search starts from the front. If α_i denotes the access probability for name $Table[i]$, then the average successful search time is $E[Y] = \sum i\alpha_i$. Then $E[Y]$ is minimized when names in the table are in the order of nonincreasing access probabilities; that is, $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$. As an example, many tables in practice follow Zipf's law [ZIPF 1949]:

$$\alpha_i = \frac{c}{i}, \quad 1 \leq i \leq n,$$

where the constant c is determined from the normalization requirement, $\sum_{i=1}^n \alpha_i = 1$. Thus:

$$c = \frac{1}{\sum_{i=1}^n \frac{1}{i}} = \frac{1}{H_n} \simeq \frac{1}{\ln(n) + C}, \quad (4.2)$$

where H_n is the partial sum of a harmonic series; that is: $H_n = \sum_{i=1}^n (1/i)$ and $C (= 0.577)$ is the Euler constant.

Now, if the names in the table are ordered as above, then the average search time is

$$E[Y] = \sum_{i=1}^n i\alpha_i = \frac{1}{H_n} \sum_{i=1}^n i = \frac{n}{H_n} \simeq \frac{n}{\ln(n) + C},$$

which is considerably less than the previous value $(n + 1)/2$, for large n .

‡

Example 4.3

Zipf's law has been used to model the distribution of Web page requests [BRES 1999]. It has been found that $p_Y(i)$, the probability of a request for the i th most popular page is inversely proportional to i [ALME 1996, WILL 1996],

$$p_Y(i) = \frac{c}{i}, \quad 1 \leq i \leq n, \quad (4.3)$$

where n is the total number of Web pages in the universe.

We assume the Web page requests are independent and the cache can hold only m Web pages regardless of the size of each Web page. If we adopt a removal policy called “least frequently used”, which always keeps the m most popular pages, then the hit ratio $h(m)$ —the probability that a request can find its page in cache—is given by

$$h(m) = \sum_{i=1}^m p_Y(i) \simeq c H_m = \frac{H_m}{H_n} \simeq \frac{\ln(m) + C}{\ln(n) + C}, \quad (4.4)$$

which means the hit ratio increases logarithmically as a function of cache size. This result is consistent with previously observed behavior of Web cache [ALME 1996, WILL 1996].

‡

Example 4.4

Recall the example of a wireless cell with five channels (Examples 1.1, and 2.2), and let X be the number of available channels. Then

$$\begin{aligned} E[X] &= \sum_{i=0}^5 ip_X(i) \\ &= 0 \cdot \frac{1}{32} + 1 \cdot \frac{5}{32} + 2 \cdot \frac{10}{32} + 3 \cdot \frac{10}{32} + 4 \cdot \frac{5}{32} + 5 \cdot \frac{1}{32} \\ &= 2.5. \end{aligned}$$

The example above illustrates that $E[X]$ need not correspond to a possible value of the random variable X . The expected value denotes the “center” of a probability mass (or density) function in the sense of a weighted average, or better, in the sense of a center of gravity.

‡

Example 4.5

Let X be a continuous random variable with an exponential density given by

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0.$$

Then

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx = \int_0^{\infty} \lambda xe^{-\lambda x} dx.$$

Let $u = \lambda x$, then $du = \lambda dx$, and

$$E[X] = \frac{1}{\lambda} \int_0^{\infty} ue^{-u} du = \frac{1}{\lambda} \Gamma 2 = \frac{1}{\lambda}, \quad \text{using formula (3.25).}$$

Thus, if a component obeys an exponential failure law with parameter λ (known as the **failure rate**), then its expected life, or its mean time to failure (MTTF), is $1/\lambda$. Similarly, if the interarrival times of jobs to a computer center are exponentially distributed with parameter λ (known as the **arrival rate**), then the mean (average) interarrival time is $1/\lambda$. Finally, if the service time requirement of a job is an exponentially distributed random variable with parameter μ (known as the **service rate**), then the mean (average) service time is $1/\mu$.

‡

Problems

- Consider a discrete random variable X with the following pmf:

$$p_X(x) = \begin{cases} \frac{1}{x(x+1)}, & x = 1, 2, \dots, \\ 0, & \text{otherwise.} \end{cases}$$

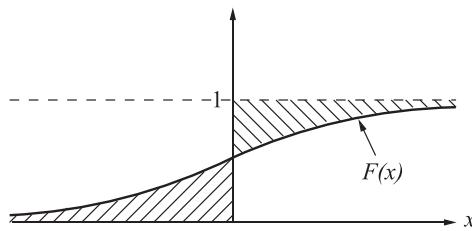


Figure 4.P.1. An alternative method of computing $E[X]$

Show that the function defined satisfies the properties of a pmf. Show that the formula (4.1) of expectation does not converge in this case and hence $E[X]$ is undefined. [Hint: Rewrite $1/x(x + 1)$ as $1/x - 1/(x + 1)$.]

2. * Using integration by parts, show (assuming that $E[X]$, $\int_0^\infty [1 - F(x)]dx$, and $\int_{-\infty}^0 F(x)dx$ are all finite) that for a continuous random variable X :

$$E[X] = \int_0^\infty [1 - F(x)]dx - \int_{-\infty}^0 F(x)dx.$$

This result states that the expectation of a random variable X equals the difference of the areas of the right-hand and left-hand shaded regions in Figure 4.P.1. (This formula applies to the case of discrete and mixed random variables as well.)

3. For a given event A show that the expectation of its indicator random variable I_A (refer to Section 2.5.9) is given by

$$E[I_A] = P(A).$$

4. For the modified exponential distribution with a mass at origin [formula (3.2)], compute its expected value.

4.2 MOMENTS

Let X be a random variable, and define another random variable Y as a function of X so that $Y = \phi(X)$. Suppose that we wish to compute $E[Y]$. In order to apply Definition (4.1), we must first compute the pmf (or pdf in the continuous case) of Y by methods of Chapter 2 (or Chapter 3 for the continuous case). An easier method of computing $E[Y]$ is to use the following result:

$$E[Y] = E[\phi(X)] = \begin{cases} \sum_i \phi(x_i) p_x(x_i), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^\infty \phi(x) f_X(x) dx, & \text{if } X \text{ is continuous,} \end{cases} \quad (4.5)$$

(provided the sum or the integral on the right-hand side is absolutely convergent).

A special case of interest is the power function $\phi(X) = X^k$. For $k = 1, 2, 3, \dots, E[X^k]$ is known as the k th moment of the random variable X . Note that the first moment, $E[X]$, is the ordinary expectation or the mean of X .

It is possible to show that if X and Y are random variables that have matching corresponding moments of *all* orders; that is, $E[X^k] = E[Y^k]$ for $k = 1, 2, \dots$, then X and Y have the same distribution.

To center the origin of measurement, it is convenient to work with powers of $X - E[X]$. We define the k th central moment, μ_k , of the random variable X by $\mu_k = E[(X - E[X])^k]$. Of special interest is the quantity

$$\mu_2 = E[(X - E[X])^2], \quad (4.6)$$

known as the variance of X , $\text{Var}[X]$, often denoted by σ^2 .

Definition (Variance). The variance of a random variable X is

$$\text{Var}[X] = \mu_2 = \sigma_X^2 = \begin{cases} \sum_i (x_i - E[X])^2 p(x_i) & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} (x - E[X])^2 f(x) dx & \text{if } X \text{ is continuous.} \end{cases} \quad (4.7)$$

It is clear that $\text{Var}[X]$ is always a nonnegative number. The square root, σ_X , of the variance is known as the **standard deviation**. Note that we will often omit subscript X . The variance and the standard deviation are measures of the “spread” or “dispersion” of a distribution. If X has a “concentrated” distribution so that X takes values near to $E[X]$ with a large probability, then the variance is small (see Figure 4.1). Figure 4.2 shows a diffuse distribution—one with a large value of σ^2 . Note that variance need not always exist (see problem 3 at the end of Section 4.7).

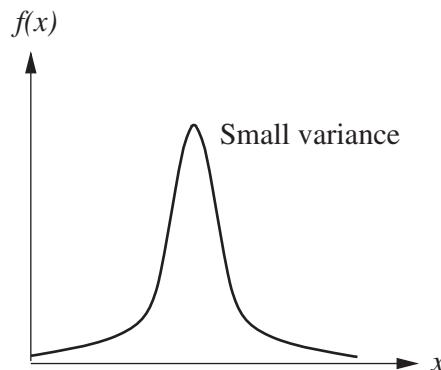


Figure 4.1. The pdf of a “concentrated” distribution

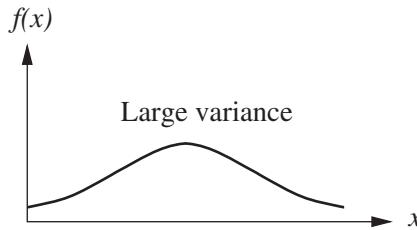


Figure 4.2. The pdf of a diffuse distribution

The third and the fourth central moments are called the *skewness* and *kurtosis*, respectively.

Example 4.6

Let X be an exponentially distributed random variable with parameter λ . Then, since $E[X] = 1/\lambda$, and $f(x) = \lambda e^{-\lambda x}$:

$$\begin{aligned}\sigma^2 &= \int_0^\infty (x - \frac{1}{\lambda})^2 \lambda e^{-\lambda x} dx \\ &= \int_0^\infty \lambda x^2 e^{-\lambda x} dx - 2 \int_0^\infty x e^{-\lambda x} dx + \frac{1}{\lambda} \int_0^\infty e^{-\lambda x} dx \\ &= \frac{1}{\lambda^2} \Gamma 3 - \frac{2}{\lambda^2} \Gamma 2 + \frac{1}{\lambda^2} \Gamma 1 = \frac{1}{\lambda^2}, \quad \text{using formula (3.25).}\end{aligned}$$

‡

The standard deviation is expressed in the same units as the individual value of the random variable. If we divide it by the mean, then we obtain a relative measure of the spread of the distribution of X . The coefficient of variation of a random variable X is denoted by C_X and defined by

$$C_X = \frac{\sigma_X}{E[X]}. \quad (4.8)$$

Note that the coefficient of variation of an exponential random variable is 1, so C_X is a measure of deviation from the exponential distribution.

Yet another function of X that is often of interest is $Y = aX + b$, where a and b are constants. It is not difficult to show that

$$E[Y] = E[aX + b] = aE[X] + b. \quad (4.9)$$

In particular, if a is zero, then $E[b] = b$; that is, the expectation of a constant random variable is that constant. If we take $a = 1$ and $b = -E[X]$, then we conclude that the first central moment, $\mu_1 = E[X - E[X]] = E[X] - E[X] = 0$.

Problems

1. * The problem of dynamic storage allocation in the main memory of a computer system [WOLM 1965] can be simplified by choosing a fixed node size, k , for allocation. Out of the k units (bytes, say) of storage allocated to a node, only $k - b$ bytes are available to the user, since b bytes are required for control information. Let the random variable L denote the length in bytes of a user request. Thus $\lceil L/(k - b) \rceil$ nodes must be allocated to satisfy the user request. Thus the total number of bytes allocated is $X = k\lceil L/(k - b) \rceil$. Find $E[X]$ as a function of k and $E[L]$. Then, by differentiating $E[X]$ with respect to k , show that the optimal value of k is approximately $b + \sqrt{2bE[L]}$.
2. Recall the problem of the mischievous student trying to open a password-protected file, and determine the expected number of trials $E[N_n]$ and the variance $\text{Var}[N_n]$ for both techniques described in problem 3, Section 2.5.
3. The number of failures of a computer system in a week of operation has the following pmf:

No. of Failures	0	1	2	3	4	5	6
Probability	.18	.28	.25	.18	.06	.04	.01

- (a) Find the expected number of failures in a week.
- (b) Find the variance of the number of failures in a week.
4. In a Bell System study made in 1961 regarding the dialing of calls between White Plains, New York, and Sacramento, California, the pmf of the number of trunks, X , required for a connection was found to be

i	1	2	3	4	5
$p_X(i)$.50	.30	.12	.07	.01

Determine the distribution function of X . Compute $E[X]$, $\text{Var}[X]$ and the mode of X . Let Y denote the number of telephone switching exchanges that this call has to pass through. Then $Y = X + 1$. Determine the pmf, the distribution function, the mean, and the variance of Y .

5. Let X , Y , and Z , respectively, denote EXP(1), two-stage hyperexponential with $\alpha_1 = .5 = \alpha_2$, $\lambda_1 = 2$, and $\lambda_2 = \frac{2}{3}$, and two-stage Erlang with parameter 2 random variables. Note that $E[X] = E[Y] = E[Z]$. Find the mode, the median, the variance, and the coefficient of variation of each random variable. Compare the densities of X , Y , and Z by plotting on the same graph. Similarly compare the three distribution functions.
6. Given a random variable X and two functions $h(x)$ and $g(x)$ satisfying the condition $h(x) \leq g(x)$ for all x , show that

$$E[h(X)] \leq E[g(X)],$$

whenever both expectations exist.

4.3 EXPECTATION BASED ON MULTIPLE RANDOM VARIABLES

Let X_1, X_2, \dots, X_n be n random variables defined on the same probability space, and let $Y = \phi(X_1, X_2, \dots, X_n)$. Then

$$\begin{aligned} E[Y] &= E[\phi(X_1, X_2, \dots, X_n)] \\ &= \begin{cases} \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} \phi(x_1, x_2, \dots, x_n) p(x_1, x_2, \dots, x_n) & (\text{discrete case}), \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \phi(x_1, x_2, \dots, x_n) f(x_1, x_2, \dots, x_n) dx_1 dx_2 \cdots dx_n & (\text{continuous case}). \end{cases} \quad (4.10) \end{aligned}$$

Example 4.7

Consider a moving head disk with the innermost cylinder of radius a and the outermost cylinder of radius b . We assume that the number of cylinders is very large and the cylinders are very close to each other, so that we may assume a continuum of cylinders. Let the random variables X and Y , respectively, denote the current and the desired position of the head. Further assume that X and Y are independent and uniformly distributed over the interval (a, b) . Therefore

$$f_X(x) = f_Y(y) = \frac{1}{b-a}, \quad a < x, y < b,$$

and

$$f(x, y) = \frac{1}{(b-a)^2}, \quad a < x, y < b.$$

Head movement for a seek operation traverses a distance that is a random variable given by $|X - Y|$. The expected seek distance is then given by (see Figure 4.3):

$$\begin{aligned} E[|X - Y|] &= \int_a^b \int_a^b |x - y| f(x, y) dx dy \\ &= \int_a^b \int_a^b |x - y| \frac{1}{(b-a)^2} dx dy \\ &= \iint_{a \leq y < x \leq b} \frac{(x-y)}{(b-a)^2} dy dx + \iint_{a \leq x < y \leq b} \frac{(y-x)}{(b-a)^2} dy dx \\ &= \frac{2}{(b-a)^2} \int_a^b \int_a^x (x-y) dy dx, \quad \text{by symmetry} \\ &= \frac{2}{(b-a)^2} \int_a^b \left(xy - \frac{y^2}{2} \right) \Big|_a^x dx \\ &= \frac{2}{(b-a)^2} \int_a^b \left(x^2 - ax - \frac{x^2}{2} + \frac{a^2}{2} \right) dx \end{aligned}$$

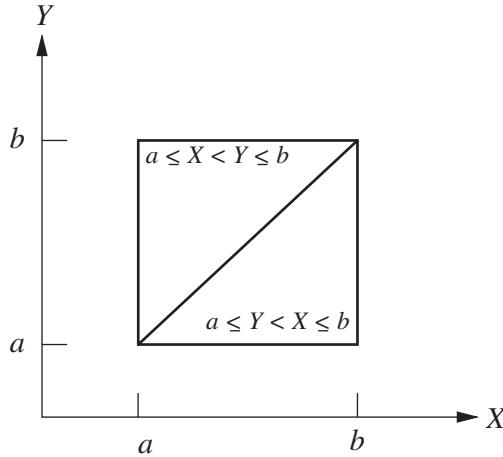


Figure 4.3. Two areas of integration for Example 4.7

$$\begin{aligned}
 &= \frac{2}{(b-a)^2} \left[\frac{b^3 - a^3}{6} - \frac{a}{2}(b^2 - a^2) + \frac{a^2(b-a)}{2} \right] \\
 &= \frac{b-a}{3}.
 \end{aligned}$$

Thus the expected seek distance is one third the maximum seek distance. Intuition may have led us to the incorrect conclusion that the expected seek distance is half of the maximum. (In practice, the expected seek distance is even smaller because of correlations between successive requests [HUNT 1980, IBM 1997].)

‡

Certain functions of random variables (e.g., sums), are of special interest and are of considerable use.

THEOREM 4.1 (The Linearity Property of Expectation).

Let X and Y be two random variables. Then the expectation of their sum is the sum of their expectations; that is, if $Z = X + Y$, then $E[Z] = E[X + Y] = E[X] + E[Y]$.

Proof: We will prove the theorem assuming that X , Y , and hence Z are continuous random variables. Proof for the discrete case is very similar.

$$\begin{aligned}
 E[X + Y] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + y)f(x, y)dx dy \\
 &= \int_{-\infty}^{\infty} x \int_{-\infty}^{\infty} f(x, y)dy dx + \int_{-\infty}^{\infty} y \int_{-\infty}^{\infty} f(x, y)dx dy
 \end{aligned}$$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} xf_X(x)dx + \int_{-\infty}^{\infty} yf_Y(y)dy \\
&\quad (\text{by definition of the marginal densities}) \\
&= E[X] + E[Y].
\end{aligned}$$

Note that Theorem 4.1 *does not* require that X and Y be independent. It can be generalized to the case of n variables:

$$E\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n E[X_i]$$

and to

$$E\left[a_0 + \sum_{i=1}^n a_i X_i\right] = a_0 + \sum_{i=1}^n a_i E[X_i], \quad (4.11)$$

where a_0, a_1, \dots, a_n are constants. For instance, let X_1, X_2, \dots, X_n , be random variables (not necessarily independent) with a common mean $\mu = E[X_i]$ ($i = 1, 2, \dots, n$). Then the expected value of their sample mean (defined in Section 3.9) is equal to μ :

$$E[\bar{X}] = E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n} \sum_{i=1}^n E[X_i] = \mu. \quad (4.12)$$

Example 4.8

We have noted that the variance:

$$\begin{aligned}
\sigma^2 &= E[(X - E[X])^2] \\
&= E[X^2 - 2XE[X] + (E[X])^2] \\
&= E[X^2] - E[2XE[X]] + (E[X])^2, \quad \text{by (4.11)} \\
&= E[X^2] - 2E[X]E[X] + (E[X])^2
\end{aligned}$$

(noting that $E[X]$ is a constant). Thus

$$\sigma^2 = E[X^2] - (E[X])^2. \quad (4.13)$$

This formula for $\text{Var}[X]$ is usually preferred over the original definition (4.7).

‡

Unlike the case of expectation of a sum, the expectation of a product of two random variables does not have a simple form, unless the two random variables are independent.

THEOREM 4.2. $E[XY] = E[X]E[Y]$, if X and Y are independent random variables.

Proof: We give a proof of the theorem under the assumption that X and Y are discrete random variables. The proof for the continuous case is similar:

$$\begin{aligned} E[XY] &= \sum_i \sum_j x_i y_j p(x_i, y_j) \\ &= \sum_i \sum_j x_i y_j p_X(x_i) p_Y(y_j) \quad \text{by independence} \\ &= \sum_i x_i p_X(x_i) \sum_j y_j p_Y(y_j) = E[X]E[Y]. \end{aligned}$$

Note that converse of Theorem 4.2 does not hold; that is, random variables X and Y may satisfy the relation $E[XY] = E[X]E[Y]$ without being independent.

Theorem 4.2 can be easily generalized to a mutually independent set of n random variables X_1, X_2, \dots, X_n :

$$E \left[\prod_{i=1}^n X_i \right] = \prod_{i=1}^n E[X_i] \quad (4.14)$$

and further to

$$E \left[\prod_{i=1}^n \phi_i(X_i) \right] = \prod_{i=1}^n E[\phi_i(X_i)].$$

Again with the assumption of independence, the variance of a sum takes a simpler form also, as follows.

THEOREM 4.3. $\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$, if X and Y are independent random variables.

Proof: From the definition of variance, we obtain

$$\begin{aligned} \text{Var}[X + Y] &= E[((X + Y) - E[X + Y])^2] \\ &= E[((X + Y) - E[X] - E[Y])^2] \\ &= E[(X - E[X])^2 + (Y - E[Y])^2 + 2(X - E[X])(Y - E[Y])] \\ &= E[(X - E[X])^2] + E[(Y - E[Y])^2] + 2E[(X - E[X])(Y - E[Y])] \\ &= \text{Var}[X] + \text{Var}[Y] + 2E[(X - E[X])(Y - E[Y])], \end{aligned}$$

by the linearity property of expectation.

The quantity $E[(X - E[X])(Y - E[Y])]$ is defined to be the **covariance** of X and Y and is denoted by $\text{Cov}(X, Y)$. It is easy to see that $\text{Cov}(X, Y)$ is zero when X and Y are independent:

$$\begin{aligned}
\text{Cov}(X, Y) &= E[(X - E[X])(Y - E[Y])] \\
&= E[XY - YE[X] - XE[Y] + E[X]E[Y]] \\
&= E[XY] - E[Y]E[X] - E[X]E[Y] + E[X]E[Y]
\end{aligned}$$

by the linearity of expectation,

$$\begin{aligned}
&= E[XY] - E[X]E[Y] \\
&= 0,
\end{aligned}$$

by Theorem 4.2, since X and Y are independent.

Therefore, $\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$ if X and Y are independent random variables.

In case X and Y are not independent, we obtain the following formula:

$$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y] + 2\text{Cov}(X, Y). \quad (4.15)$$

Theorem 4.3 can be generalized for a set of n mutually independent random variables X_1, X_2, \dots, X_n ; and constants a_1, a_2, \dots, a_n :

$$\text{Var}\left[\sum_{i=1}^n a_i X_i\right] = \sum_{i=1}^n a_i^2 \text{Var}[X_i]. \quad (4.16)$$

Thus, if X_1, X_2, \dots, X_n are mutually independent random variables with a common variance $\sigma^2 = \text{Var}[X_i]$ ($i = 1, 2, \dots, n$), then the variance of their sum is given by

$$\text{Var}\left[\sum_{i=1}^n X_i\right] = n\text{Var}[X_i] = n\sigma^2, \quad (4.17)$$

and the variance of their sample mean is

$$\begin{aligned}
\text{Var}[\bar{X}] &= \text{Var}\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n^2} \text{Var}\left[\sum_{i=1}^n X_i\right] \\
&= \frac{\sigma^2}{n}.
\end{aligned} \quad (4.18)$$

We have noted that $\text{Cov}(X, Y) = 0$, if X and Y are independent random variables. However, it is possible for two random variables to satisfy the condition $\text{Cov}(X, Y) = 0$ without being independent.

Definition (Uncorrelated Random Variables). Random variables X and Y are said to be uncorrelated provided $\text{Cov}(X, Y) = 0$.

Since $\text{Cov}(X, Y) = E[XY] - E[X]E[Y]$, an equivalent definition of uncorrelated random variables is the condition $E[XY] = E[X]E[Y]$. It follows that independent random variables are uncorrelated, but the converse need not hold.¹

Example 4.9

Let X be uniformly distributed over the interval $(-1, 1)$ and let $Y = X^2$, so that Y is completely dependent on X . Noting that for all odd values of $k > 0$, the k th moment $E[X^k] = 0$, we have

$$E[XY] = E[X^3] = 0 \quad \text{and} \quad E[X]E[Y] = 0 \cdot E[Y] = 0.$$

Therefore X and Y are uncorrelated!

‡

We have declared that $\text{Cov}(X, Y) = 0$ means X and Y are uncorrelated. On the other hand, if X and Y are linearly related—that is, $X = aY$ for some constant $a \neq 0$ —then, since $E[X] = aE[Y]$, we have

$$\text{Cov}(X, Y) = a\text{Var}[Y] = \frac{1}{a} \text{Var}[X]$$

or

$$\text{Cov}^2(X, Y) = \text{Var}[X]\text{Var}[Y].$$

In the general case, it can be shown that

$$0 \leq \text{Cov}^2(X, Y) \leq \text{Var}[X]\text{Var}[Y] \tag{4.19}$$

using the following Cauchy–Schwarz inequality:

$$(E[XY])^2 \leq E[X^2]E[Y^2]. \tag{4.20}$$

$\text{Cov}(X, Y)$ measures the degree of linear dependence (or the degree of correlation) between the two random variables. Recalling Example 4.9, we note that the notion of covariance completely misses the quadratic dependence. It is often useful to define a measure of this dependence in a *scale-independent* fashion. The correlation coefficient $\rho(X, Y)$ is defined by

$$\begin{aligned} \rho(X, Y) &= \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}[X]\text{Var}[Y]}} \\ &= \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \end{aligned} \tag{4.21}$$

whenever σ_X and σ_Y are defined.

¹If X and Y are both Gaussian distributed, then the converse is also true [PEEB 1980].

Using the relation (4.19), we conclude that

$$-1 \leq \rho(X, Y) \leq 1. \quad (4.22)$$

Also

$$\rho(X, Y) = \begin{cases} -1, & \text{if } X = -aY(a > 0), \\ 0, & \text{if } X \text{ and } Y \text{ are uncorrelated,} \\ +1, & \text{if } X = aY(a > 0). \end{cases} \quad (4.23)$$

Problems

1. Consider discrete random variables X and Y [BLAK 1979] with the joint pmf as shown below:

		Y		
		-1	0	1
X	-1	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
	0	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{8}$
1	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{8}$	
2	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	

Are X and Y independent? Are they uncorrelated?

2. Consider the discrete version of Example 4.7 and assume that the records of a file are evenly scattered over n tracks of a moving-head disk. Compute the expected number of gaps X between tracks that the head will pass over between two reads. Also compute the variance of X . [Hint: $\sum_{i=1}^N i^3 = (\sum_{i=1}^N i)^2$ is a useful identity.] Next assume that the seek time T is a function $\theta(X)$ of the number of gaps passed over. Compute $E[T]$ and $\text{Var}[T]$, assuming $T = 30.0 + 0.5X$.
3. Consider a directed graph G with n nodes. Let X_{ij} be a variable defined so that

$$X_{ij} = \begin{cases} 0 & \text{if there is no edge between node } i \text{ and node } j, \\ 1 & \text{otherwise.} \end{cases}$$

Assume that the $\{X_{ij}\}$ are mutually independent Bernoulli random variables with parameter p . The corresponding graph is called a **p-random-graph**. Find the pmf, the expected value, and the variance of the total number of edges X in the graph.

4. Let X_1, X_2, \dots, X_n be mutually independent and identically distributed random variables with means μ and variance σ^2 . Let $\bar{X} = (\sum_{i=1}^n X_i)/n$. Show that

$$\sum_{k=1}^n (X_k - \bar{X})^2 = \sum_{k=1}^n (X_k - \mu)^2 - n(\bar{X} - \mu)^2$$

and hence

$$E \left[\sum_{k=1}^n (X_k - \bar{X})^2 \right] = (n-1)\sigma^2.$$

5. A certain telephone company charges for calls in the following way: \$0.20 for the first 3 min or less; \$0.08 per min for any additional time. Thus, if X is the duration of a call, the cost Y is given by

$$Y = \begin{cases} 0.20, & 0 \leq X \leq 3, \\ 0.20 + 0.08(X-3), & X \geq 3. \end{cases}$$

Find the expected value of the cost of a call ($E[Y]$), assuming that the duration of a call is exponentially distributed with a mean of $1/\lambda$ minutes. Use $1/\lambda = 1, 2, 3, 4, 5$ min.

6. Show that $\text{Cov}^2(X, Y) \leq \text{Var}[X]\text{Var}[Y]$.
7. Random variables X and Y are said to be **orthogonal** if and only if $E[XY] = 0$.
- (a) Assuming that X and Y are orthogonal, determine the conditions under which they are uncorrelated.
 - (b) Assuming that X and Y are uncorrelated, determine the conditions under which they are orthogonal.
8. Consider random variables X and Y with the joint pdf (bivariate Gaussian):

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \cdot \exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X} \right)^2 - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right] \right\}$$

where $\rho \neq \pm 1$. Show that $\text{Cov}(X, Y) = \rho\sigma_X\sigma_Y$. Hence show that if X, Y are jointly Gaussian and uncorrelated (i.e., $\rho = 0$), then they are also independent. Note that this is *not* true in general.

4.4 TRANSFORM METHODS

In many probability problems, the form of the density function (or the pmf in the discrete case) may be so complex so as to render computations difficult, if not impossible. As an example, recall the analysis of the program MAX. A transform can provide a compact description of a distribution, and it is relatively easy to compute the mean, the variance, and other moments directly from a transform rather than resorting to a tedious sum (discrete case) or an equally tedious integral (continuous case). The transform methods are particularly useful in problems involving sums of independent random variables and in solving difference equations (discrete case) and differential

equations (continuous case) related to a stochastic process. We will revisit the z transform (also called the *probability generating function*) and introduce the Laplace–Stieltjes transform and the characteristic function (also called the *Fourier transform*). We will first define the moment generating function and derive the above three transforms as special cases.

For a random variable X , $e^{X\theta}$ is another random variable. The expectation $E[e^{X\theta}]$ will be a function of θ . Define the moment generating function (MGF) $M_X(\theta)$, abbreviated $M(\theta)$, of the random variable X by

$$M(\theta) = E[e^{X\theta}] \quad (4.24)$$

so that

$$M(\theta) = \begin{cases} \sum_j e^{x_j \theta} p(x_j), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} e^{x\theta} f(x) dx, & \text{if } X \text{ is continuous.} \end{cases} \quad (4.25)$$

The expectation defining $M(\theta)$ may not exist for all real numbers θ , but for most problems that we encounter, there will be an interval of θ values within which $M(\theta)$ does exist. Note that definition (4.24) allows us to define the moment generating function for a mixed random variable as well.

The closely related characteristic function of a random variable X is given by

$$N_X(\tau) = N(\tau) = M_X(i\tau). \quad (4.26)$$

Here i denotes $\sqrt{-1}$. $N(\tau)$ is known among electrical engineers as the Fourier transform. The advantage here is that for any X , its characteristic function, $N_X(\tau)$, is always defined for all τ . If X is a nonnegative continuous random variable, then we define the (one-sided) Laplace–Stieltjes transform (LST) of X :

$$L_X(s) = L(s) = M_X(-s) = \int_0^{\infty} e^{-sx} dF(x) = \int_0^{\infty} e^{-sx} f(x) dx. \quad (4.27)$$

Finally, if X is a nonnegative integer-valued discrete random variable, then, as we defined in Chapter 2, its z transform is

$$G_X(z) = G(z) = E[z^X] = M_X(\ln z) = \sum_{i=0}^{\infty} p_x(i) z^i. \quad (4.28)$$

The reasons for the usefulness of transform methods will be summarized by the following properties of the transforms. We will give the properties of the moment generating function, but by an appropriate substitution for θ , a similar property can be stated for all the other transforms as well.

THEOREM 4.4 (Linear Translation). Let $Y = aX + b$; then:

$$M_Y(\theta) = e^{b\theta} M_X(a\theta).$$

Proof:

$$\begin{aligned} E[e^{Y\theta}] &= E[e^{(aX+b)\theta}] \\ &= E[e^{b\theta} e^{aX\theta}] \\ &= e^{b\theta} E[e^{aX\theta}] \end{aligned}$$

by the linearity property of expectation.

THEOREM 4.5 (The Convolution Theorem). Let X_1, X_2, \dots, X_n be mutually independent random variables on a given probability space, and let $Y = \sum_{i=1}^n X_i$. If $M_{X_i}(\theta)$ exists for all i , then $M_Y(\theta)$ exists, and

$$M_Y(\theta) = M_{X_1}(\theta)M_{X_2}(\theta) \cdots M_{X_n}(\theta).$$

Thus the moment generating function of a sum of independent random variables is the product of the moment generating functions.

Proof:

$$\begin{aligned} M_Y(\theta) &= E[e^{Y\theta}] = E[e^{(X_1+X_2+\cdots+X_n)\theta}] \\ &= E\left[\prod_{i=1}^n e^{X_i\theta}\right] \\ &= \prod_{i=1}^n E[e^{X_i\theta}], \quad \text{by independence} \\ &= \prod_{i=1}^n M_{X_i}(\theta). \end{aligned}$$

Thus we may find the transform of a sum of independent random variables without any n -dimensional integration. But the technique will be of little value unless we can recover the distribution function from the transform. The following theorem, which we state without proof, is useful in this regard.

THEOREM 4.6 (Correspondence Theorem or Uniqueness Theorem). If $M_{X_1}(\theta) = M_{X_2}(\theta)$ for all θ , then $F_{X_1}(x) = F_{X_2}(x)$ for all x .

In other words, if two random variables have the same transform, then they have the same distribution function.

Next we study the **moment generating property** of the MGF. $e^{X\theta}$ can be expanded into a power series:

$$e^{X\theta} = 1 + X\theta + \frac{X^2\theta^2}{2!} + \cdots + \frac{X^k\theta^k}{k!} + \cdots.$$

Taking expectations on both sides (assuming all the expectations exist) yields

$$\begin{aligned} M(\theta) &= E[e^{X\theta}] \\ &= 1 + E[X]\theta + \cdots + \frac{E[X^k]\theta^k}{k!} + \cdots \\ &= \sum_{k=0}^{\infty} \frac{E[X^k]\theta^k}{k!}. \end{aligned}$$

Therefore the coefficient of $\theta^k/k!$ in the power-series expansion of its MGF yields the k th moment $E[X^k]$ of the random variable X . Alternatively

$$E[X^k] = \left. \frac{d^k M}{d\theta^k} \right|_{\theta=0}, \quad k = 1, 2, \dots \quad (4.29)$$

Note that $E[X^0] = M(0) = 1$.

The corresponding property for the Laplace–Stieltjes transform is

$$E[X^k] = (-1)^k \left. \frac{d^k L(s)}{ds^k} \right|_{s=0}, \quad k = 1, 2, \dots, \quad (4.30)$$

for the z transform:

$$E[X] = \left. \frac{dG}{dz} \right|_{z=1}, \quad (4.31)$$

$$E[X^2] = \left. \frac{d^2 G}{dz^2} \right|_{z=1} + \left. \left| \frac{dG}{dz} \right|_{z=1} \right., \quad (4.32)$$

and for the characteristic function:

$$E[X^k] = (i)^{-k} \left. \frac{d^k N}{d\tau^k} \right|_{\tau=0}, \quad k = 1, 2, \dots. \quad (4.33)$$

Example 4.10

Let X be exponentially distributed with parameter λ . Then

$$f_X(x) = \lambda e^{-\lambda x}, x > 0$$

and

$$\begin{aligned} L_X(s) &= \int_0^\infty e^{-sx} \lambda e^{-\lambda x} dx \\ &= \frac{\lambda}{s + \lambda} \int_0^\infty (\lambda + s) e^{-(\lambda+s)x} dx \\ &= \frac{\lambda}{s + \lambda}. \end{aligned} \quad (4.34)$$

Now, using (4.30), we have

$$E[X] = (-1) \frac{dL_X}{ds} \Big|_{s=0} = (-1) \frac{-\lambda}{(\lambda+s)^2} \Big|_{s=0} = \frac{1}{\lambda},$$

as derived earlier in Example 4.5. Also

$$E[X^2] = \frac{d^2L_X}{ds^2} \Big|_{s=0} = \frac{2\lambda}{(\lambda+s)^3} \Big|_{s=0} = \frac{2}{\lambda^2}$$

and

$$\text{Var}[X] = \frac{2}{\lambda^2} - \left(\frac{1}{\lambda} \right)^2 = \frac{1}{\lambda^2}.$$

#

Example 4.11

We are now in a position to complete the analysis of program MAX (Section 2.6). Recall that the PGF (probability generating function) of the random variable X_n , was shown to have the recurrence

$$G_{X_n}(z) = \frac{(z+n-1)}{n} G_{X_{n-1}}(z), \quad n \geq 2$$

with

$$G_{X_1}(z) = 1.$$

Here X_n denotes the number of executions of the **if** statement in program MAX. Then the expected number of executions of the **if** statement is derived using the property (4.31):

$$\begin{aligned} E[X_n] &= \frac{dG_{X_n}}{dz} \Big|_{z=1} \\ &= \frac{1}{n} G_{X_{n-1}}(1) + \frac{z+n-1}{n} \Big|_{z=1} \cdot \frac{dG_{X_{n-1}}(z)}{dz} \Big|_{z=1} \\ &= \frac{1}{n} + E[X_{n-1}] \end{aligned}$$

(since $G_X(1) = 1$ for any PGF). With $E[X_1] = 0$, we have

$$E[X_n] = \sum_{i=2}^n \frac{1}{i} = H_n - 1 \simeq \ln n - 0.423.$$

To compute the variance of X_n , first observe that if Y_k is a Bernoulli random variable with parameter $p = 1/k$, then

$$G_{Y_k}(z) = (1 - \frac{1}{k}) + \frac{z}{k} = \frac{z + k - 1}{k}.$$

If Y_2, Y_3, \dots, Y_n are mutually independent, then, using the convolution theorem, $W = \sum_{k=2}^n Y_k$ has the PGF:

$$\begin{aligned} G_W(z) &= \prod_{k=2}^n G_{Y_k}(z) \\ &= \prod_{k=2}^n \frac{z + k - 1}{k} \\ &= G_{X_n}(z). \end{aligned}$$

So we conclude, by the correspondence theorem, that X_n has the same distribution as W :

$$X_n = \sum_{k=2}^n Y_k.$$

(Note that although X_n is the sum of $n - 1$ mutually independent Bernoulli random variables, it is not a binomial random variable; why?) Now, since $\{Y_k\}$ are mutually independent, we use formula (4.16) to obtain

$$\begin{aligned} \text{Var}[X_n] &= \sum_{k=2}^n \text{Var}[Y_k] \\ &= \sum_{k=2}^n \frac{1}{k} \left(1 - \frac{1}{k}\right) \\ &= H_n - H_n^{(2)}, \end{aligned}$$

where $H_n^{(2)}$ is defined to be $\sum_{k=1}^n \frac{1}{k^2}$.

The power of the notion of transforms should now be clear, since we could compute the mean and the variance without the explicit knowledge of pmf, which in this case is quite complex (it involves Stirling numbers!).

#

Example 4.12 (Analysis of Straight Selection Sort)

We are given an array **a** of type **item** as declared below:

```
#define n 100

struct itemstr
```

```

{
int key;
τ info;
};

typedef itemstr item;
item a[n];

```

We are required to sort the array so that keys are in nondecreasing order. We can use the following procedure [WIRT 1976]:

```

for (i = n; n > 1; n--) {
    1: ‘‘Assign the index of the item with the largest
       key among the items a[1], a[2], ..., a[i] to k’’
    2: ‘‘Exchange a[i] and a[k]’’;
}

```

Assume that each element of the array is a large record and, therefore, exchanging (or moving) items is expensive. The total number of moves due to the second statement is easily computed and seen to be a fixed number. But the number of moves in the first statement is variable. Assume that the program MAX is used to perform this operation. Then the number of moves for a fixed value of i will be given by X_i , which was studied in Chapter 2 and in Example 4.11. Now the total number of moves, W_n , contributed by the first statement is given by

$$W_n = \sum_{i=2}^n X_i.$$

Then

$$E[W_n] = \sum_{i=2}^n E[X_i] = \sum_{i=2}^n (H_i - 1) \simeq O(n(\ln n)).$$

Where, $O(n(\ln n))$ notation is used to denote computational complexity [KNUT 1997].

#

Example 4.13

Let X_i ($i = 1, 2$) be independent exponentially distributed random variables with parameters λ_i . If $\lambda_1 = \lambda_2 = \lambda$, then $X = X_1 + X_2$ will be a two-stage Erlang random variable. Assume $\lambda_1 \neq \lambda_2$, implying that X is a hypoexponentially distributed random variable. Using formula (4.34), we have

$$L_{X_1}(s) = \frac{\lambda_1}{\lambda_1 + s} \quad \text{and} \quad L_{X_2}(s) = \frac{\lambda_2}{\lambda_2 + s}.$$

By the convolution theorem

$$L_X(s) = \frac{\lambda_1 \lambda_2}{(\lambda_1 + s)(\lambda_2 + s)}. \quad (4.35)$$

We expand this expression into a partial fraction:

$$L_X(s) = \frac{a_1\lambda_1}{\lambda_1 + s} + \frac{a_2\lambda_2}{\lambda_2 + s},$$

where

$$a_1 = \frac{\lambda_2}{\lambda_2 - \lambda_1} \text{ and } a_2 = \frac{\lambda_1}{\lambda_1 - \lambda_2}.$$

Recalling that if Y is EXP (λ), then $L_Y(s) = \lambda/(\lambda + s)$, we conclude (using the uniqueness theorem of Laplace–Stieltjes transforms) that

$$\begin{aligned} f_X(x) &= a_1\lambda_1 e^{-\lambda_1 x} + a_2\lambda_2 e^{-\lambda_2 x} \\ &= \frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 x} + \frac{\lambda_1\lambda_2}{\lambda_1 - \lambda_2} e^{-\lambda_2 x}, \end{aligned}$$

which is the hypoexponential density.

‡

More generally, if $\{X_i | i = 1, 2, \dots, n\}$ are mutually independent and exponentially distributed with parameters λ_i ($\lambda_i \neq \lambda_j, i \neq j$), then $X = \sum_{i=1}^n X_i$ is an n -stage hypoexponential random variable and

$$L_X(s) = \prod_{i=1}^n \frac{\lambda_i}{\lambda_i + s}.$$

Using the technique of partial fraction expansion [KOBA 1978], the Laplace–Stieltjes transform of X can be rewritten as

$$L_X(s) = \sum_{i=1}^n \frac{a_i\lambda_i}{\lambda_i + s}, \quad (4.36)$$

where

$$a_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{\lambda_j}{\lambda_j - \lambda_i}. \quad (4.37)$$

Again, from the uniqueness theorem of Laplace–Stieltjes transforms, it follows that

$$f_X(x) = \sum_{i=1}^n a_i \lambda_i e^{-\lambda_i x}. \quad (4.38)$$

(Although this form of f_X resembles a hyperexponential density function, it is quite a different hypoexponential density; why?)

‡

Example 4.14

Let X be normally distributed with parameters μ and σ^2 . Then

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right], \quad -\infty < x < \infty.$$

The characteristic function of X is given by

$$N_X(\tau) = \int_{-\infty}^{\infty} e^{i\tau x} f_X(x) dx.$$

Making the change of variables $y = (x - \mu)/\sigma$, we obtain

$$\begin{aligned} N_X(\tau) &= \int_{-\infty}^{\infty} e^{i(\sigma y + \mu)\tau} \frac{e^{-(1/2)y^2}}{\sqrt{2\pi}} dy \\ &= e^{i\tau\mu} \int_{-\infty}^{\infty} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} e^{i\tau\sigma y} dy \\ &= e^{i\tau\mu + (i\tau\sigma)^2/2} \int_{-\infty}^{\infty} e^{-1/2(y - i\tau\sigma)^2} \frac{dy}{\sqrt{2\pi}} \end{aligned}$$

(noting that $i^2 = -1$). Thus, the characteristic function of a normal random variable is given by

$$N_X(\tau) = e^{i\tau\mu - \tau^2\sigma^2/2}, \quad (4.39)$$

since it can be shown that

$$\int_{-\infty}^{\infty} e^{-1/2(y - i\tau\sigma)^2} \frac{dy}{\sqrt{2\pi}} = 1.$$

[it is the area under the normal density $N(i\tau\sigma, 1)$]. Check that

$$N_X(0) = e^0 = 1.$$

To compute the expected value, we use equation (4.33):

$$\begin{aligned} E[X] &= \frac{1}{i} \left. \frac{dN_X}{d\tau} \right|_{\tau=0} \\ &= \frac{1}{i} \left. \left[(i\mu - \tau\sigma^2 e^{i\tau\mu - \frac{\tau^2\sigma^2}{2}}) \right] \right|_{\tau=0} \\ &= \frac{1}{i} [i\mu e^0] = \mu. \end{aligned}$$

Similarly, it can be shown that

$$\begin{aligned} E[X^2] &= \frac{1}{i^2} \left. \frac{d^2N_X}{d\tau^2} \right|_{\tau=0} \\ &= \sigma^2 + \mu^2 \end{aligned}$$

(after the computations are worked out).

#

Thus the normal distribution $N(\mu, \sigma^2)$ has mean μ and variance σ^2 . This distribution is completely specified by the two parameters.

Example 4.15 (Proof of Theorem 3.6)

Let X_1, X_2, \dots, X_n be mutually independent Gaussian random variables so that X_j is $N(\mu_j, \sigma_j^2)$, $j = 1, 2, \dots, n$. Then from formula (4.39) we have

$$N_{X_j}(\tau) = e^{i\tau\mu_j - \tau^2\sigma_j^2/2}, \quad j = 1, 2, \dots, n.$$

Let $Y = \sum_{i=1}^n X_i$; then, using the convolution theorem, we have

$$\begin{aligned} N_Y(\tau) &= \prod_{j=1}^n N_{X_j}(\tau) \\ &= e^{i\tau\mu - \tau^2\sigma^2/2}. \end{aligned}$$

where

$$\mu = \sum_{j=1}^n \mu_j \text{ and } \sigma^2 = \sum_{j=1}^n \sigma_j^2.$$

Comparing the characteristic function given above with that in (4.39), we conclude that Y is $N(\mu, \sigma^2)$.

Characteristic functions are somewhat more complex than the MGF, but they have two advantages: (1) $N_X(\tau)$ is finite for all random variables X and for all real numbers τ ; and (2) the characteristic function possesses the inversion property, so that the density $f_X(x)$ may be derived from $N_X(\tau)$ by the inversion formula:

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ix\tau} N_X(\tau) d\tau. \quad (4.40)$$

Inversion of a Laplace–Stieltjes transform is usually performed using a table lookup. It is helpful first to perform a partial fraction expansion of the transform. See Appendix D for further details.

#

Problems

1. Show that if $X_1 \sim N(\mu_1, \sigma_1^2)$ and $X_2 \sim N(\mu_2, \sigma_2^2)$ are independent random variables, then the random variable $Y = X_1 - X_2$ is also normally distributed with

$$E[Y] = \mu_1 - \mu_2 \quad \text{and} \quad \text{Var}[Y] = \sigma_1^2 + \sigma_2^2.$$

Generalize to the case of n mutually independent random variables with

$$X_i \sim N(\mu_i, \sigma_i^2) \quad \text{and} \quad Y = \sum_{i=1}^n a_i X_i.$$

2. Take the program to find the maximal element of a given (one-dimensional) array B of size n (discussed in Chapter 2). Call this subroutine MAX. Write a driver

for this subroutine that generates all $n!$ permutations of the set $\{1, 2, \dots, n\}$ and, for each such permutation, loads it into the array B and calls subroutine MAX. Count the number of exchanges made in subroutine MAX. Add the number of exchanges over all permutations and divide the sum by $(n!)$. Check whether the result equals $H_n - 1$. Similarly compute the variance and check it against the expression $H_n - H_n^{(2)}$. Use $n = 1, 3, 5$, and 10. To generate $n!$ permutations systematically, you may refer to Sedgewick [SEDG 1977].

4.5 MOMENTS AND TRANSFORMS OF SOME DISTRIBUTIONS

4.5.1 Discrete Uniform Distribution

The pmf is given by

$$p_x(i) = \frac{1}{n}, \quad 1 \leq i \leq n.$$

Therefore

$$E[X^k] = \sum_{i=1}^n \frac{i^k}{n}.$$

Then, the mean is

$$E[X] = \frac{n+1}{2},$$

and the variance is

$$\begin{aligned} \text{Var}[X] &= E[X^2] - (E[X])^2 \\ &= \frac{(n+1)(2n+1)}{6} - \frac{(n+1)^2}{4} \\ &= \frac{n+1}{12}[2(2n+1) - 3(n+1)] \\ &= \frac{(n+1)(n-1)}{12} = \frac{n^2-1}{12}. \end{aligned}$$

The coefficient of variation is

$$C_X = \sqrt{\frac{n^2-1}{3(n+1)^2}} = \sqrt{\frac{1}{3} \left(1 - \frac{2}{n+1}\right)},$$

so

$$0 \leq C_X < \frac{1}{\sqrt{3}}.$$

The generating function in this case is

$$G_X(z) = \sum_{i=1}^n \frac{1}{n} z^i = \frac{1}{n} \sum_{i=1}^n z^i.$$

4.5.2 Bernoulli pmf

$$p_X(0) = q, \quad p_X(1) = p, \quad p + q = 1.$$

$$E[X^k] = 0^k \cdot q + 1^k \cdot p = p, \quad k = 1, 2, \dots$$

Therefore, the mean is

$$E[X] = p,$$

and the variance is

$$\text{Var}[X] = E[X^2] - (E[X])^2 = p - p^2 = p(1 - p) = pq.$$

The coefficient of variation is

$$C_X = \sqrt{\frac{q}{p}},$$

and the generating function is

$$G_X(z) = (1 - p) + pz = q + pz.$$

4.5.3 Binomial Distribution

Note that a binomial random variable X is the sum of n mutually independent Bernoulli random variables X_1, X_2, \dots, X_n . Thus

$$X = \sum_{i=1}^n X_i,$$

and the linearity property of the expectation yields the following result:

$$E[X] = \sum_{i=1}^n E[X_i] = np.$$

Similarly, using formula (4.17), we get the variance:

$$\text{Var}[X] = \sum_{i=1}^n \text{Var}[X_i] = npq,$$

The coefficient of variation is as follows:

$$C_X = \sqrt{\frac{npq}{n^2 p^2}} = \sqrt{\frac{q}{np}}.$$

Thus, the expected number of successes in a sequence of n Bernoulli trials is np . Also note that the coefficient of variation reduces as n increases, and

it approaches zero in the limit as $n \rightarrow \infty$. This observation is related to the weak law of large numbers, as will be seen later. We can easily obtain the generating function, using the convolution theorem:

$$G_X(z) = \prod_{i=1}^n G_{X_i}(z) = (q + pz)^n.$$

4.5.4 Geometric Distribution

The pmf is given by

$$p_x(i) = pq^{i-1}, i = 1, 2, \dots$$

The mean is computed by

$$\begin{aligned} E[X] &= \sum_{i=1}^{\infty} ipq^{i-1} \\ &= p \sum_{i=1}^{\infty} iq^{i-1} \\ &= p \sum_{i=0}^{\infty} \frac{d}{dq}(q^i) \\ &= p \frac{d}{dq} \left(\sum_{i=0}^{\infty} q^i \right) \\ &= p \frac{d}{dq} \left(\frac{1}{1-q} \right) \\ &= \frac{p}{(1-q)^2} \\ &= \frac{1}{p}. \end{aligned}$$

Therefore, if we assume that, at the end of a CPU burst, a program requests an I/O operation with probability q and it finishes execution with probability p , then the average number of CPU bursts per program is given by $1/p$. Similarly, if a communication channel transmits a message correctly, on each trial, with probability p , then the average number of trials required for a successful transmission is $1/p$.

The generating function of X is given by

$$G_X(z) = \sum_{i=1}^{\infty} pq^{i-1} z^i$$

$$\begin{aligned}
&= pz \sum_{i=1}^{\infty} (qz)^{i-1} \\
&= pz \sum_{j=0}^{\infty} (qz)^j \\
&= \frac{pz}{1 - qz}.
\end{aligned}$$

From this, $E[X]$ can be derived in an easier fashion:

$$\begin{aligned}
E[X] &= \left. \frac{dG}{dz} \right|_{z=1} \\
&= \left. \frac{p(1 - qz) - pz(-q)}{(1 - qz)^2} \right|_{z=1} \\
&= \frac{p(1 - q) + pq}{(1 - q)^2} \\
&= \frac{p}{p^2} \\
&= \frac{1}{p}.
\end{aligned}$$

The variance is computed in a fashion similar to that used for the mean; we get

$$\text{Var}[X] = \frac{q}{p^2} \quad \text{and} \quad C_X = \sqrt{\frac{qp^2}{p^2}} = \sqrt{q} = \sqrt{1-p}.$$

For the modified geometric distribution, with the pmf $p_Y(i) = pq^i$, $i = 0, 1, 2, \dots$, we obtain

$$E[Y] = \frac{q}{p}, \quad \text{Var}[Y] = \frac{q}{p^2}, \quad C_Y = \sqrt{\frac{qp^2}{p^2q^2}} = \frac{1}{\sqrt{q}},$$

and the generating function is

$$G_Y(z) = \frac{p}{1 - qz}.$$

4.5.5 Poisson pmf

$$p_X(i) = \frac{\alpha^i e^{-\alpha}}{i!}, \quad 0 \leq i < \infty, \quad \alpha > 0.$$

Then

$$\begin{aligned} E[X] &= \sum_{i=0}^{\infty} \frac{i\alpha^i}{i!} e^{-\alpha} \\ &= \alpha e^{-\alpha} \sum_{i=1}^{\infty} \frac{\alpha^{i-1}}{(i-1)!} \\ &= \alpha e^{-\alpha} e^{\alpha} = \alpha. \end{aligned}$$

If the number of job arrivals to a file server in interval $(0, t]$ is Poisson distributed with parameter $\alpha = \lambda t$, then the average number of arrivals in that interval is λt . Thus, the average arrival rate of jobs is λ .

The $\text{Var}[X]$ is easily computed to be α . Therefore

$$C_X = \frac{1}{\sqrt{\alpha}}.$$

The generating function is given by

$$G_X(z) = \sum_{k=0}^{\infty} e^{-\alpha} \frac{\alpha^k}{k!} z^k = e^{-\alpha} \sum_{k=0}^n \frac{(\alpha z)^k}{k!} = e^{-\alpha} e^{\alpha z} = e^{-\alpha(1-z)}.$$

4.5.6 Continuous Uniform Distribution

The density function is given by

$$f_X(x) = \frac{1}{b-a}, \quad a < x < b.$$

Then

$$E[X] = \int_a^b \frac{x}{b-a} dx = \frac{b^2 - a^2}{2(b-a)} = \frac{b+a}{2},$$

the midpoint of the interval (a, b) . The k th moment is computed as follows:

$$E[X^k] = \frac{1}{b-a} \int_a^b x^k dx = \frac{b^{k+1} - a^{k+1}}{(k+1)(b-a)}.$$

Therefore

$$\begin{aligned} \text{Var}[X] &= E[X^2] - (E[X])^2 \\ &= \frac{b^3 - a^3}{3(b-a)} - \frac{(b+a)^2}{4} \\ &= \frac{(b-a)^2}{12} \end{aligned}$$

and

$$C_X = \frac{b-a}{b+a} \sqrt{\frac{1}{3}}.$$

Assuming $0 \leq a < b$, the Laplace–Stieltjes transform of X is

$$\begin{aligned} L_X(s) &= \int_a^b e^{-sx} \frac{1}{b-a} dx \\ &= \frac{e^{-as} - e^{-bs}}{s(b-a)}. \end{aligned}$$

4.5.7 Exponential Distribution

We have already determined that if the density is given by

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

then the mean is

$$E[X] = \frac{1}{\lambda},$$

the variance is

$$\text{Var}[X] = \frac{1}{\lambda^2},$$

the coefficient of variation is

$$C_X = 1,$$

and the Laplace–Stieltjes transform is

$$L_X(s) = \frac{\lambda}{\lambda + s}.$$

4.5.8 Gamma Distribution

The density function of the random variable X is given by

$$f_X(x) = \frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}, \quad x > 0.$$

Then, making the substitution $u = \lambda x$, we compute the mean

$$E[X] = \int_0^\infty \frac{x^\alpha \lambda^\alpha e^{-\lambda x}}{\Gamma(\alpha)} dx = \frac{1}{\lambda \Gamma(\alpha)} \int_0^\infty u^\alpha e^{-u} du,$$

and hence, using formula (3.25), we obtain

$$E[X] = \frac{\Gamma(\alpha+1)}{\lambda \Gamma(\alpha)} = \frac{\alpha}{\lambda}.$$

Similarly, the variance is computed to be

$$\text{Var}[X] = \frac{\alpha}{\lambda^2},$$

and thus:

$$C_X = \frac{1}{\sqrt{\alpha}}.$$

Note that if α is an integer, then these results could be shown by the properties of sums, since X will be the sum of α exponential random variables. Note also that the coefficient of variation of a gamma random variable is less than 1 if $\alpha > 1$; it is equal to 1 if $\alpha = 1$; and otherwise the coefficient of variation is greater than 1. Thus the gamma family is capable of modeling a very powerful class of random variables exhibiting from almost none to a very high degree of variability.

The Laplace–Stieltjes transform is given by

$$\begin{aligned} L_X(s) &= \int_0^\infty e^{-sx} \frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)} dx \\ &= \frac{\lambda^\alpha}{(\lambda+s)^\alpha} \int_0^\infty \frac{(\lambda+s)^\alpha x^{\alpha-1} e^{-(\lambda+s)x}}{\Gamma(\alpha)} dx \\ &= \frac{\lambda^\alpha}{(\lambda+s)^\alpha} \end{aligned}$$

since the last integral is the area under a gamma density with parameter $\lambda + s$ and α —that is, 1. If α were an integer, this result could be derived using the convolution property of the Laplace–Stieltjes transforms.

4.5.9 Hypoexponential Distribution

We have seen that if X_1, X_2, \dots, X_n are mutually independent exponentially distributed random variables with parameters $\lambda_1, \lambda_2, \dots, \lambda_n$ ($\lambda_i \neq \lambda_j$, $i \neq j$), respectively, then

$$X = \sum_{i=1}^n X_i$$

is hypoexponentially distributed with parameters $\lambda_1, \lambda_2, \dots, \lambda_n$; that is, X is HYPO $(\lambda_1, \lambda_2, \dots, \lambda_n)$. The mean of X can then be obtained using the linearity property of expectation, so that

$$E[X] = \sum_{i=1}^n E[X_i] = \sum_{i=1}^n \frac{1}{\lambda_i}.$$

Also, because of independence of $\{X_i\}$, we get the variance of X as

$$\text{Var}[X] = \sum_{i=1}^n \text{Var}[X_i] = \sum_{i=1}^n \frac{1}{\lambda_i^2},$$

$$C_X = \sqrt{\frac{\sum_{i=1}^n 1/\lambda_i^2}{(\sum_{i=1}^n 1/\lambda_i)^2}} \quad \text{and} \quad L_X(s) = \prod_{i=1}^n \frac{\lambda_i}{\lambda_i + s}.$$

Note that $C_X \leq 1$, and thus this distribution can model random variables with variability less than or equal to that of the exponential distribution.

It has been observed that service times at I/O devices are generally hypo-exponentially distributed. Also, programs are often organized into a set of sequential phases (or job steps). If the execution time of the i th step is exponentially distributed with parameter λ_i , then the total program execution time is hypoexponentially distributed and its parameters are specified by the formulas above.

4.5.10 Hyperexponential Distribution

The density in this case is given by

$$f(x) = \sum_{i=1}^n \alpha_i \lambda_i e^{-\lambda_i x}, \quad \sum_{i=1}^n \alpha_i = 1, \quad \alpha_i \geq 0, \quad x > 0.$$

Then the mean is

$$\begin{aligned} E[X] &= \int_0^\infty \left(\sum_{i=1}^n x \alpha_i \lambda_i e^{-\lambda_i x} \right) dx \\ &= \sum_{i=1}^n \alpha_i \int_0^\infty x \lambda_i e^{-\lambda_i x} dx \\ &= \sum_{i=1}^n \frac{\alpha_i}{\lambda_i}. \end{aligned}$$

(since the last integral represents the expected value of an exponentially distributed random variable with parameter λ_i). Similarly

$$E[X^2] = \sum_{i=1}^n \frac{2\alpha_i}{\lambda_i^2}$$

and

$$\begin{aligned}\text{Var}[X] &= E[X^2] - (E[X])^2 \\ &= 2 \sum_{i=1}^n \frac{\alpha_i}{\lambda_i^2} - \left[\sum_{i=1}^n \frac{\alpha_i}{\lambda_i} \right]^2.\end{aligned}$$

Finally:

$$\begin{aligned}C_X^2 &= \frac{\text{Var}[X]}{(E[X])^2} = \frac{2 \sum_{i=1}^n (\alpha_i / \lambda_i^2) - (\sum_{i=1}^n \alpha_i / \lambda_i)^2}{(\sum_{i=1}^n \alpha_i / \lambda_i)^2} \\ &= 2 \frac{\sum_{i=1}^n (\alpha_i / \lambda_i^2)}{(\sum_{i=1}^n \alpha_i / \lambda_i)^2} - 1.\end{aligned}$$

Using the well-known Cauchy–Schwartz inequality [an alternative form of (4.20)], we can show that $C_X > 1$ for $n > 1$. The inequality states that

$$\left(\sum_{i=1}^n a_i b_i \right)^2 \leq \left(\sum_{i=1}^n a_i^2 \right) \left(\sum_{i=1}^n b_i^2 \right). \quad (4.41)$$

Substitute $a_i = \sqrt{\alpha_i}$ and $b_i = (\sqrt{\alpha_i})/\lambda_i$; then:

$$\begin{aligned}\left(\sum_{i=1}^n \frac{\alpha_i}{\lambda_i} \right)^2 &\leq \left(\sum_{i=1}^n \alpha_i \right) \left(\sum_{i=1}^n \frac{\alpha_i}{\lambda_i^2} \right) \\ &= \sum_{i=1}^n \frac{\alpha_i}{\lambda_i^2},\end{aligned}$$

which implies that:

$$C_X^2 = 2 \frac{\sum_{i=1}^n (\alpha_i / \lambda_i^2)}{(\sum_{i=1}^n \alpha_i / \lambda_i)^2} - 1 \geq 1.$$

Thus the hyperexponential distribution models random variables with more variability than does the exponential distribution. As has been pointed out, CPU service times usually follow this distribution.

The Laplace–Stieltjes transform is

$$\begin{aligned}L_X(s) &= \int_0^\infty e^{-sx} \sum_{i=1}^n \alpha_i \lambda_i e^{-\lambda_i x} dx \\ &= \sum_{i=1}^n \alpha_i \int_0^\infty \lambda_i e^{-\lambda_i x} e^{-sx} dx \\ &= \sum_{i=1}^n \frac{\alpha_i \lambda_i}{\lambda_i + s}.\end{aligned}$$

4.5.11 Weibull Distribution

Recall that this is the random variable with the pdf $f(x) = H'(x)e^{-H(x)}$ where $H(x) = \lambda x^\alpha$:

$$f(x) = \lambda \alpha x^{\alpha-1} e^{-\lambda x^\alpha}, \quad \lambda > 0, \quad \alpha > 0, \quad x > 0.$$

The mean is

$$E[X] = \int_0^\infty \lambda x \alpha x^{\alpha-1} e^{-\lambda x^\alpha} dx.$$

Now, making the substitution $u = \lambda x^\alpha$, we obtain

$$\begin{aligned} E[X] &= \int_0^\infty \left(\frac{u}{\lambda}\right)^{1/\alpha} e^{-u} du \\ &= \left(\frac{1}{\lambda}\right)^{1/\alpha} \int_0^\infty u^{1/\alpha} e^{-u} du = \left(\frac{1}{\lambda}\right)^{1/\alpha} \Gamma\left(1 + \frac{1}{\alpha}\right). \end{aligned}$$

This reduces to the value $1/\lambda$ when $\alpha = 1$, since then the Weibull distribution becomes the exponential distribution. Similarly

$$\begin{aligned} E[X^2] &= \int_0^\infty \lambda x^2 \alpha x^{\alpha-1} e^{-\lambda x^\alpha} dx \\ &= \left(\frac{1}{\lambda}\right)^{2/\alpha} \int_0^\infty u^{2/\alpha} e^{-u} du \\ &= \left(\frac{1}{\lambda}\right)^{2/\alpha} \Gamma\left(1 + \frac{2}{\alpha}\right), \end{aligned}$$

so

$$\begin{aligned} \text{Var}[X] &= \left(\frac{1}{\lambda}\right)^{2/\alpha} \Gamma\left(1 + \frac{2}{\alpha}\right) - \left(\frac{1}{\lambda}\right)^{2/\alpha} \left[\Gamma\left(1 + \frac{1}{\alpha}\right) \right]^2 \\ &= \left(\frac{1}{\lambda}\right)^{2/\alpha} \left[\Gamma\left(1 + \frac{2}{\alpha}\right) - \left[\Gamma\left(1 + \frac{1}{\alpha}\right) \right]^2 \right] \end{aligned}$$

and

$$C_X = \sqrt{\frac{\Gamma(1 + 2/\alpha) - [\Gamma(1 + 1/\alpha)]^2}{[\Gamma(1 + 1/\alpha)]^2}} = \sqrt{\frac{\Gamma(1 + 2/\alpha)}{[\Gamma(1 + 1/\alpha)]^2} - 1}.$$

4.5.12 Log-logistic Distribution

The density is given by

$$f(t) = \frac{\lambda \kappa (\lambda t)'^{\kappa-1}}{[1 + (\lambda t)^\kappa]^2}, \quad t \geq 0;$$

the mean, by

$$E[X] = \frac{\pi}{\lambda \kappa \sin(\frac{\pi}{\kappa})}, \kappa > 1, \tan^{-1}(\lambda^\kappa) \neq \pi;$$

the second moment,

$$E[X^2] = \frac{2\pi}{\lambda^2 \kappa \sin(\frac{2\pi}{\kappa})}, \kappa > 2, \tan^{-1}(\lambda^\kappa) \neq \pi;$$

and the i th moment, by

$$E[X^i] = \frac{\pi i}{\lambda^i \kappa \sin(\frac{\pi i}{\kappa})}, \kappa > i, \tan^{-1}(\lambda^\kappa) \neq \pi.$$

4.5.13 Pareto Distribution

The density is given by

$$f(x) = \alpha k^\alpha x^{-\alpha-1}, x \geq k ; \alpha, k > 0; \quad (4.42)$$

the mean, by

$$\begin{aligned} E[X] &= \int_k^\infty \alpha k^\alpha x^{-\alpha} dx \\ &= \begin{cases} \frac{k^\alpha}{\alpha-1}, & \alpha > 1, \\ \infty, & \alpha \leq 1; \end{cases} \end{aligned}$$

and the second moment, by

$$\begin{aligned} E[X^2] &= \int_k^\infty \alpha k^\alpha x^{-\alpha+1} dx \\ &= \begin{cases} \frac{k^{2\alpha}}{\alpha-2}, & \alpha > 2, \\ \infty, & \alpha \leq 2. \end{cases} \end{aligned}$$

Therefore the Pareto distribution has infinite mean if the shape parameter $\alpha \leq 1$, and has infinite variance if the shape parameter $\alpha \leq 2$. Thus, as α decreases, an arbitrarily large portion of the probability mass may be present in the tail of its pdf. In practical cases, a random variable that follows a Pareto distribution with $\alpha \leq 2$ can give rise to extremely large values with a nonzero probability. This is the phenomenon commonly observed on the Internet, for example, the Web file size and the think time of Web browser.

4.5.14 The Normal Distribution

The density

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}, \quad -\infty < x < \infty.$$

The mean, the variance, and the characteristic function have been derived earlier:

$$E[X] = \mu, \quad \text{Var}[X] = \sigma^2, \quad N_X(\tau) = e^{i\tau\mu - \tau^2\sigma^2/2}.$$

Problems

- Consider an database server designed to handle a maximum of 15 transactions per second. During the peak hour of its activity, transactions arrive at the average rate of 10 per second (10 s^{-1}). Assuming that the number of transactions arriving per second follows a Poisson distribution, compute the probability that the server will be overloaded during a peak hour.
- The CPU time requirement X of a typical job can be modeled by the following hyperexponential distribution:

$$P(X \leq t) = \alpha(1 - e^{-\lambda_1 t}) + (1 - \alpha)(1 - e^{-\lambda_2 t}),$$

where $\alpha = 0.6$, $\lambda_1 = 10$, and $\lambda_2 = 1$. Compute (a) the probability density function of X , (b) the mean service time $E[X]$, (c) the variance of service time $\text{Var}[X]$, and (d) the coefficient of variation. Plot the distribution and the density function of X .

- The CPU time requirement, T , for jobs has a gamma distribution with mean of 40 s and variance of 400 s^2 .
 - Find the shape parameter α and the scale parameter λ .
 - A short job ($T < 20 \text{ s}$) gets priority. Compute the probability that a randomly chosen job is a short job.
- A telephone exchange can handle at most 20 simultaneous conversations. It has been observed that an incoming call finds an “all busy” signal 1% of the time. Assuming that the number of incoming calls, X , per unit time has a Poisson distribution, find the parameter α (or the average call arrival rate) of the distribution.
- For the three parameter Weibull distribution [formula (3.33)], find the $E[X]$.
- The time to failure distribution of Tandem software was found to be captured well by a two-phase hyperexponential distribution with the following pdf:

$$f(t) = \alpha_1 \lambda_1 e^{-\lambda_1 t} + \alpha_2 \lambda_2 e^{-\lambda_2 t},$$

with $\alpha_1 = 0.87, \alpha_2 = 0.13, \lambda_1 = 0.10, \lambda_2 = 2.78$ [LEE 1993]. Find the mean and variance of the time to failure.

4.6 COMPUTATION OF MEAN TIME TO FAILURE

Let X denote the lifetime of a component so that its reliability $R(t) = P(X > t)$ and $R'(t) = -f(t)$. Then the **expected life** or the **mean time to failure** (MTTF) of the component is given by

$$E[X] = \int_0^\infty tf(t)dt = - \int_0^\infty tR'(t)dt.$$

Integrating by parts we obtain

$$E[X] = -tR(t)\Big|_0^\infty + \int_0^\infty R(t)dt.$$

Now, since $R(t)$ approaches zero faster than t approaches ∞ , we have

$$E[X] = \int_0^\infty R(t)dt. \quad (4.43)$$

(This formula is a special case of problem 2 at the end of Section 4.1.) This latter expression for MTTF is in more common use in reliability theory. More generally:

$$\begin{aligned} E[X^k] &= \int_0^\infty t^k f(t)dt \\ &= - \int_0^\infty t^k R'(t)dt \\ &= -t^k R(t)\Big|_0^\infty + \int_0^\infty kt^{k-1} R(t)dt. \end{aligned}$$

Thus

$$E[X^k] = \int_0^\infty kt^{k-1} R(t)dt. \quad (4.44)$$

In particular

$$\text{Var}[X] = \int_0^\infty 2tR(t)dt - \left[\int_0^\infty R(t)dt \right]^2. \quad (4.45)$$

If the component lifetime is exponentially distributed, then $R(t) = e^{-\lambda t}$ and

$$E[X] = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda},$$

$$\begin{aligned}\text{Var}[X] &= \int_0^\infty 2te^{-\lambda t} dt - \frac{1}{\lambda^2} \\ &= \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}\end{aligned}$$

as derived earlier. Next we consider a system consisting of n components connected in several different ways (we continue to make the usual assumptions of independence).

4.6.1 Series System

Assume that the lifetime of the i th component for a series system is exponentially distributed with parameter λ_i . Then system reliability [using equation (3.63)] is given by

$$R(t) = \prod_{i=1}^n R_i(t) = \prod_{i=1}^n e^{-\lambda_i t} = \exp \left[- \left(\sum_{i=1}^n \lambda_i \right) t \right].$$

Thus, the lifetime of the system is also exponentially distributed with parameter $\lambda = \sum_{i=1}^n \lambda_i$. Therefore the series system MTTF is

$$\frac{1}{\sum_{i=1}^n \lambda_i}. \quad (4.46)$$

The MTTF of a series system is much smaller than the MTTF of its components.

If X_i denotes the lifetime of component i (not necessarily exponentially distributed), and X denotes the series system lifetime, then we can show that

$$0 \leq E[X] \leq \min\{E[X_i]\} \quad (4.47)$$

which gives rise to the common remark that a system is weaker than its weakest link. To prove inequality (4.47), note that

$$R_X(t) = \prod_{i=1}^n R_{X_i}(t) \leq \min_i \{R_{X_i}(t)\},$$

since $0 \leq R_{X_i}(t) \leq 1$. Then

$$\begin{aligned}E[X] &= \int_0^\infty R_X(t) dt \leq \min_i \left\{ \int_0^\infty R_{X_i}(t) dt \right\} \\ &= \min_i \{E[X_i]\}.\end{aligned}$$

4.6.2 Parallel System

Consider a parallel system of n independent components, where X_i denotes the lifetime of component i and X denoting the lifetime of the system. Then

$$X = \max\{X_1, X_2, \dots, X_n\}$$

and, using formula (3.64), we get

$$R_X(t) = 1 - \prod_{i=1}^n [1 - R_{X_i}(t)] \geq 1 - [1 - R_{X_i}(t)], \quad \text{for all } i, \quad (4.48)$$

which implies that the reliability of a parallel redundant system is larger than that of any of its components. Therefore

$$\begin{aligned} E[X] &= \int_0^\infty R_X(t) dt \geq \max_i \left\{ \int_0^\infty R_{X_i}(t) dt \right\} \\ &= \max_i \{E[X_i]\}. \end{aligned} \quad (4.49)$$

Now assume that X_i is exponentially distributed with parameter λ (all components have the same parameter). Then

$$R_X(t) = 1 - (1 - e^{-\lambda t})^n$$

and

$$E[X] = \int_0^\infty [1 - (1 - e^{-\lambda t})^n] dt.$$

Let $u = 1 - e^{-\lambda t}$; then $dt = 1/\lambda(1-u)du$. Thus:

$$E[X] = \frac{1}{\lambda} \int_0^1 \frac{1-u^n}{1-u} du.$$

Now since the integrand above is the sum of a finite geometric series:

$$\begin{aligned} E[X] &= \frac{1}{\lambda} \int_0^1 \left(\sum_{i=1}^n u^{i-1} \right) du \\ &= \frac{1}{\lambda} \sum_{i=1}^n \int_0^1 u^{i-1} du. \end{aligned}$$

Note that

$$\int_0^1 u^{i-1} du = \frac{u^i}{i} \Big|_0^1 = \frac{1}{i}.$$

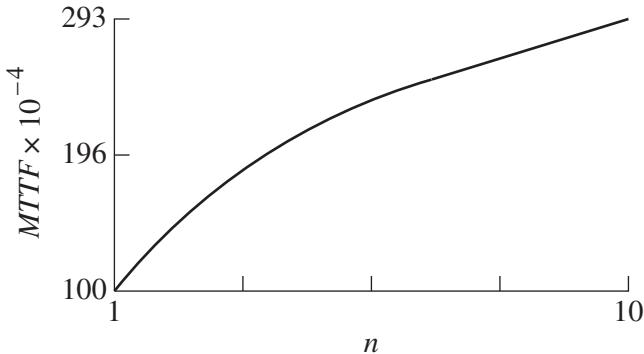


Figure 4.4. The variation in the expected life with the degree of (parallel) redundancy (simplex failure rate $\lambda = 10^{-6}$)

Thus, with the usual exponential assumptions, the MTTF of a parallel redundant system is given by

$$E[X] = \frac{1}{\lambda} \sum_{i=1}^n \frac{1}{i} = \frac{H_n}{\lambda} \simeq \frac{\ln(n) + C}{\lambda}. \quad (4.50)$$

Figure 4.4 shows the expected life of a parallel system as a function of n . It should be noted that beyond $n = 2$ or 3 , the gain in expected life (due to adding one additional component), is not very significant. Note that the rate of increase in the MTTF is $1/(n\lambda)$.

Alternatively, formula (4.50) for $E[X]$, can be derived by noting that X is hypoexponentially distributed with parameters $n\lambda, (n-1)\lambda, \dots, \lambda$. (See Theorem 3.5.) In other words, $X = \sum_{i=1}^n Y_i$, where Y_i is exponentially distributed with parameter $i\lambda$. Then, using the linearity property of expectation, we have

$$E[X] = \sum_{i=1}^n E[Y_i] = \sum_{i=1}^n \frac{1}{i\lambda} = \frac{H_n}{\lambda}.$$

Also, since the $\{Y_i\}$ are mutually independent:

$$\text{Var}[X] = \sum_{i=1}^n \text{Var}[Y_i] = \sum_{i=1}^n \frac{1}{i^2 \lambda^2} = \frac{1}{\lambda^2} H_n^{(2)}. \quad (4.51)$$

Note that $C_X < 1$; hence, not only does the parallel configuration increase the MTTF but it also reduces the variability of the system lifetime.

4.6.3 Standby Redundancy

Assume that the system has one component operating and $(n - 1)$ cold (unpowered) spares. The failure rate of an operating component is λ , and a

cold spare does not fail. Furthermore, the switching equipment is failure free. Let X_i be the lifetime of the i th component from the point it is put into operation until its failure. Then the system lifetime X is given by

$$X = \sum_{i=1}^n X_i.$$

Thus X has an n -stage Erlang distribution, and therefore

$$E[X] = \frac{n}{\lambda} \quad \text{and} \quad \text{Var}[X] = \frac{n}{\lambda^2}. \quad (4.52)$$

Note that the gain in expected life is linear as a function of the number of components, unlike the case of parallel redundancy. Of course, the price paid is the added complexity of the detection and switching mechanism. Furthermore, if we allow the detection and switching equipment to fail, the gain will be much less. Note that the computation of $E[X]$ does not require the independence assumption (but the computation of variance does).

4.6.4 TMR and TMR/Simplex Systems

We have noted that the reliability, $R(t)$, of a TMR system consisting of components with independent exponentially distributed lifetimes (with parameter λ) is given by [formula (3.68)]

$$R(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}.$$

Then the expected life is given by

$$E[X] = \int_0^\infty 3e^{-2\lambda t} dt - \int_0^\infty 2e^{-3\lambda t} dt.$$

Thus, the TMR MTTF is

$$E[X] = \frac{3}{2\lambda} - \frac{2}{3\lambda} = \frac{5}{6\lambda}. \quad (4.53)$$

Compare this with the expected life of a single component ($1/\lambda$). Thus, TMR actually *reduces* (by about 16%) the MTTF over the simplex system. This fact points out that in certain cases MTTF can be a misleading measure. Although TMR has a lower MTTF than does simplex, we know that TMR has higher reliability than simplex for “short” missions, defined by mission time $t < (\ln 2)/\lambda$.

Next consider the case when the voter used in TMR is not perfect and it has reliability $r \leq 1$. Then

$$R(t) = r(3e^{-2\lambda t} - 2e^{-3\lambda t})$$

and the MTTF of a TMR system with imperfect voter is

$$E[X] = \frac{5r}{6\lambda}. \quad (4.54)$$

Thus TMR MTTF is degraded even further.

Next consider the improvement of TMR known as TMR/simplex. In Example 3.28 the lifetime X of this system was shown to be the sum of two exponential random variables, one with parameter 3λ and the other with parameter λ . Then the MTTF of TMR/simplex is given by

$$E[X] = \frac{1}{3\lambda} + \frac{1}{\lambda} = \frac{4}{3\lambda}. \quad (4.55)$$

Thus the TMR/simplex has 33% longer expected life than the simplex.

4.6.5 The k -out-of- n System

We showed in Example 3.31 that the lifetime $L(k|n)$ of an k -out-of- n system with components having independent exponentially distributed lifetimes (with parameter λ) is the sum of $(n - k + 1)$ exponentially distributed random variables with parameters $k\lambda, (k + 1)\lambda, \dots, n\lambda$. Therefore, the MTTF of a k -out-of- n system is given by

$$E[L(k|n)] = \sum_{i=k}^n \frac{1}{i\lambda} = \frac{H_n - H_{k-1}}{\lambda}. \quad (4.56)$$

Also, the variance of the lifetime of a k -out-of- n system is

$$\text{Var}[L(k|n)] = \sum_{i=k}^n \frac{1}{i^2\lambda^2} = \frac{H_n^{(2)} - H_{k-1}^{(2)}}{\lambda^2}. \quad (4.57)$$

It may be verified that TMR is a special case of a k -out-of- n system with $n = 3$ and $k = 2$.

4.6.6 The Hybrid k -out-of- n System

Consider a system of n operating components and m warm spares. Of the n active components k are required for the system to function correctly. An active component has an exponential lifetime distribution with parameter λ , and now we will let the spare also fail with failure rate $\mu < \lambda$. It is for this reason that the spare is called a “warm spare”. The lifetime $L(k|n, m)$ was shown in Example 3.32 to be the sum of $n - k + 1 + m$ exponentially distributed random variables with parameters $n\lambda + m\mu, n\lambda + (m - 1)\mu, \dots, n\lambda + \mu$,

$n\lambda, \dots, k\lambda$. Then the MTTF of a hybrid k -out-of- n system is given by

$$E[L(k|n, m)] = \sum_{i=1}^m \frac{1}{n\lambda + i\mu} + \sum_{i=k}^n \frac{1}{i\lambda}. \quad (4.58)$$

Also, the variance of the lifetime of such a system is given by

$$\text{Var}[L(k|n, m)] = \sum_{i=1}^m \frac{1}{(n\lambda + i\mu)^2} + \sum_{i=k}^n \frac{1}{i^2\lambda^2}. \quad (4.59)$$

All the previous cases we have considered are special cases of hybrid k -out-of- n . For example, the series system corresponds to $m = 0$, $k = n$. The parallel system corresponds to $m = 0$, $k = 1$. The standby system corresponds to $m = n - 1$, $n = 1$, $k = 1$, $\mu = 0$. The k -out-of- n system corresponds to $m = 0$.

Example 4.16

Consider the workstation–file server (WFS) example in Chapter 3 (Example 3.21). Given that the times to failure for the workstations and the file servers are exponentially distributed, calculate the MTTF for the system when $n = 2$ for the workstation, $m = 1$ for the file server, and $k = l = 1$, that is, the system is up so long as a workstation and the file server are up.

Let λ_w and λ_f represent the failure rates of each workstation and the file server, respectively. The system reliability is

$$\begin{aligned} R(t) &= [1 - (1 - R_w(t))^2]R_f(t) \\ &= [1 - (1 - e^{-\lambda_w t})^2]e^{-\lambda_f t} \\ &= (2e^{-\lambda_w t} - e^{-2\lambda_w t})e^{-\lambda_f t} \\ &= 2e^{-(\lambda_w + \lambda_f)t} - e^{-(2\lambda_w + \lambda_f)t}. \end{aligned}$$

Hence the mean time to failure for the system is given by

$$\begin{aligned} \text{MTTF} &= \int_0^\infty R(t)dt \\ &= \int_0^\infty [2e^{-(\lambda_w + \lambda_f)t} - e^{-(2\lambda_w + \lambda_f)t}] dt \\ &= \frac{2}{\lambda_w + \lambda_f} - \frac{1}{2\lambda_w + \lambda_f}. \end{aligned}$$
‡

Example 4.17

Recall the SEN (shuffle exchange network) discussed in Example 3.22. Given that the switching element of SEN follows the exponential failure law with parameter λ , we calculate the MTTF for the $N \times N$ SEN.

The reliability of each switching element is $r_{SE}(t) = e^{-\lambda t}$. We have

$$\begin{aligned} \text{MTTF}_{\text{SEN}} &= \int_0^{\infty} R_{\text{SEN}}(t) dt \\ &= \int_0^{\infty} [r_{SE}(t)]^{(N/2)\log_2 N} dt \\ &= \int_0^{\infty} [e^{-\lambda t}]^{(N/2)\log_2 N} dt. \end{aligned}$$

Thus

$$\text{MTTF}_{\text{SEN}} = \frac{2}{N\lambda\log_2 N}.$$

We can generalize these computations to the case of Weibull time-to-failure distribution for each element.

#

Problems

1. The time to failure T of a device is known to follow a normal distribution with mean μ and $\sigma = 10000$ h. Determine the value of μ if the device is to have a reliability equal to 0.90 for a mission time of 50000 h.
 2. * Consider a series system of two independent components. The lifetime of the first component is exponentially distributed with parameter λ , and the lifetime of the second component is normally distributed with parameters μ and σ^2 . Determine the reliability $R(t)$ of the system and show that the expected life of the system is
- $$\frac{1}{\lambda} \left[1 - \exp \left(-\lambda\mu + \frac{\lambda^2\sigma^2}{2} \right) \right].$$
3. The failure rate for a certain type of component is $\lambda(t) = at$ ($t \geq 0$), where $a > 0$ and is constant. Find the component's reliability, and its expected life (or MTTF).
 4. Two alternative workstations are being considered for acquisition. Workstation A consists of 10 chips, each with a constant failure rate of 10^{-5} h⁻¹ and all 10 chips must function properly for the system to function. Workstation B consists of five chips, each having a time-dependent failure rate given by at per hour, for some constant a and all five chips must function properly for the workstation to function. If both workstations have the same mean time to failure, which one should be recommended? Assume that the reliability for a mission time of 1000 h is the criterion for selection.
 5. The data obtained from testing a device indicate that the expected life is 5 h and the variance is approximately 1 h². Compare the reliability functions obtained by assuming (a) a Weibull failure law with $\alpha = 5$, (b) a normal failure pdf with $\mu = 5$ and $\sigma^2 = 1$, and (c) a gamma pdf with appropriate values of α and λ . Plot the results.

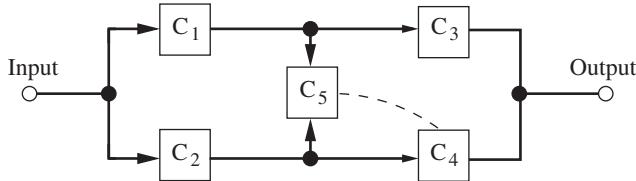


Figure 4.P.2. A system with dependent failures

6. The failure rate of a device is given by

$$h(t) = \begin{cases} at, & 0 < t < 1000 \text{ h}, \\ b, & t \geq 1000 \text{ h}. \end{cases}$$

Choose b so that $h(t)$ is continuous, and find an expression for device reliability.

7. Consider the system shown in Figure 4.P.2. Each component has an exponential failure law with parameter λ . All components behave independently, except that whenever C_4 fails it triggers an immediate failure of C_5 and vice versa. Find the reliability and the expected life of the system.
8. Carnegie-Mellon multiprocessor C.mmp consists of processors, switches, and memory units. Consider a configuration with 16 processors, 16 64K (64-kilobyte) memories, and one switch. At least four processors and four memories are required for a given task. Assume that we have a (constant) failure rate for each processor of $68.9 \text{ failures}/10^6 \text{ h}$, $224 \text{ failures}/10^6 \text{ h}$ for each memory, and a failure rate for the switch of $202 \text{ failures}/10^6 \text{ h}$. Compute the reliability function for the system. Also compute the MTTF of the system.
9. Consider a parallel redundant system of two independent components with the lifetime of i th component $X_i \sim \text{EXP}(\lambda_i)$. Show that system MTTF is given by

$$\text{MTTF} = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2}.$$

Generalize to the case of n components. Next consider a standby redundant system consisting these two components. Assuming that the component in the spare status does not fail, obtain the reliability and the MTTF of the system.

10. Compare the reliability of the workstation and file server example (see Example 3.21 and 4.16) under the exponential and Weibull failure distribution assumptions. To make a fair comparison, choose Weibull scale parameters in such way that the MTTFs of individual workstation and file server under the two assumptions are the same. Assume $l = m = k = 2$, $n = 3$, $\text{MTTF}_f = 200 \text{ h}$, $\text{MTTF}_w = 1000 \text{ h}$. Choose shape parameter $\alpha = 0.5$.
11. Assuming constant failure rates λ_c and λ_v for a control channel and a voice channel, respectively, write down system reliability and system MTTF expressions for problem 6 of Section 1.10. Repeat for problem 7 in the same section.

12. Recall Example 1.21. Assuming constant failure rates λ_i , ($i \in \{x, p, d, c\}$) for an XCVR, a pass-thru, a duplexer, and a combiner, respectively, write down the system reliability and system MTTF expressions.
13. An airplane has four propellers, two on each side [LEEM 1995]. The airplane will fly (or function correctly) if at least one propeller on each wing functions. Assuming that each propeller has a constant failure rate λ , find explicit expressions for system reliability and system MTTF.

4.7 INEQUALITIES AND LIMIT THEOREMS

We have mentioned that the distribution function (equivalently the density function or the pmf) provides a complete characterization of a random variable, and that the probability of any event concerning the random variable can be determined from it. Numbers such as the mean or the variance provide a limited amount of information about the random variable. We have discussed methods to compute various moments (including mean and variance), given the distribution function. Conversely, if all the moments, $E[X^k]$, $k = 1, 2, \dots$, are given, then we can reconstruct the distribution function via the transform. In case all moments are not available, we are not able to recover the distribution function in general. However, it may still be possible to obtain bounds on the probabilities of various events based on the limited information.

First assume that we are given the mean $E[X] = \mu$ of a nonnegative random variable X , where μ is assumed to be finite. Then the Markov inequality states that for $t > 0$

$$P(X \geq t) \leq \frac{\mu}{t}. \quad (4.60)$$

To prove this inequality, fix $t > 0$ and define the random variable Y by

$$Y = \begin{cases} 0, & \text{if } X < t, \\ t, & \text{if } X \geq t. \end{cases}$$

Then Y is a discrete random variable with the pmf:

$$\begin{aligned} p_Y(0) &= P(X < t), \\ p_Y(t) &= P(X \geq t). \end{aligned}$$

Thus

$$E[Y] = 0 \cdot p_Y(0) + t \cdot p_Y(t) = tP(X \geq t).$$

Now, since $X \geq Y$, we have $E[X] \geq E[Y]$ and hence

$$E[X] \geq E[Y] = tP(X \geq t)$$

which gives the desired inequality.

Example 4.18

Consider a system with MTTF = 100 h. We can get a bound on the reliability of the system for a mission time t using the Markov inequality:

$$R(t) = P(X \geq t) \leq \frac{100}{t}.$$

Thus, if the mission time exceeds $100/0.9 \simeq 111$ h, we know that the reliability will be less than or equal to 0.9. This suggests that if the required level of reliability is 0.9, then mission time can be no more than 111 hours (it may have to be restricted further). ‡

It is not difficult to see that the inequality (4.60) is quite crude, since only the mean is assumed to be known. For example, let the lifetime, X , of a system be exponentially distributed with mean $1/\lambda$. Then the reliability $R(t) = e^{-\lambda t}$, and the Markov inequality asserts that

$$R(t) \leq \frac{1}{\lambda t} \quad \text{or} \quad \frac{1}{R(t)} \geq \lambda t$$

that is

$$e^{\lambda t} \geq \lambda t,$$

which is quite poor in this case. On the other hand, using our knowledge of the distribution of X , let us reconsider Example 4.18. Now the mission time at which the required level of reliability is certainly lost is computed from $e^{-t/100} \leq 0.9$, or $t \geq 10.5$ h, which allows for missions much shorter than that suggested by the Markov inequality.

Next assume that both the mean μ and the variance σ^2 are given. We can now get a better estimate of the probability of events of interest by using the Chebyshev inequality:

$$P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}, \quad t > 0. \quad (4.61)$$

This inequality formalizes the intuitive meaning of the variance given earlier; if σ is small, there is a high probability for getting a value close to the mean, and if σ is large, there is a high probability for getting values farther away from the mean.

To prove the Chebyshev inequality (4.61), we apply the Markov inequality (4.60) to the nonnegative random variable $(X - \mu)^2$ and number t^2 to obtain

$$P[(X - \mu)^2 \geq t^2] \leq \frac{E[(X - \mu)^2]}{t^2} = \frac{\sigma^2}{t^2}, \quad (4.62)$$

noting that the event $[(X - \mu)^2 \geq t^2] = [|X - \mu| \geq t]$ yields the Chebyshev inequality (4.61).

The importance of Chebyshev's inequality lies in its generality. No assumption on the nature of random variable X is made other than that it has a finite variance. For most distributions, there are bounds for $P(|X - \mu| \geq t)$ sharper than that given by Chebyshev's inequality; however, examples show that in general the bound given by this inequality cannot be improved (see problem 4 at the end of this section).

Example 4.19

Let X be the execution time of a job on a server, and assume that X is exponentially distributed with mean $1/\lambda$ and variance $1/\lambda^2$. Then, using the Chebyshev inequality, we have

$$P\left(\left|X - \frac{1}{\lambda}\right| \geq t\right) \leq \frac{1}{\lambda^2 t^2}.$$

In particular, if we let $t = 1/\lambda$, this inequality does not give us any information, since it yields

$$P\left(\left|X - \frac{1}{\lambda}\right| \geq \frac{1}{\lambda}\right) \leq 1.$$

But if we compute this probability from the distribution function $F_X(x) = 1 - e^{-\lambda x}$, we get

$$\begin{aligned} P\left(\left|X - \frac{1}{\lambda}\right| \geq t\right) &= P\left(0 \leq X \leq \frac{1}{\lambda} - t \quad \text{or} \quad \frac{1}{\lambda} + t \leq X < \infty\right) \\ &= F\left(\frac{1}{\lambda} - t\right) + 1 - F\left(\frac{1}{\lambda} + t\right) \\ &= 1 - e^{\lambda t - 1} + e^{-\lambda t - 1} \end{aligned}$$

and thus

$$P\left(\left|X - \frac{1}{\lambda}\right| \geq \frac{1}{\lambda}\right) = e^{-2} < 1.$$

#

Two alternate forms of Chebyshev's inequality are easily derived from (4.61):

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}, \quad (4.63)$$

$$P(|X - \mu| \leq k\sigma) \geq 1 - \frac{1}{k^2}. \quad (4.64)$$

Another important result can be obtained by applying Chebyshev's inequality to the binomial distribution. Substituting $\mu = np$ and $\sigma = \sqrt{np(1-p)}$ into (4.64), we get

$$P\left(|X - np| < k\sqrt{np(1-p)}\right) \geq 1 - \frac{1}{k^2}$$

and

$$P\left(\left|\frac{X}{n} - p\right| < k\sqrt{\frac{p(1-p)}{n}}\right) \geq 1 - \frac{1}{k^2}.$$

Substitute ϵ for $k\sqrt{p(1-p)/n}$ to obtain

$$P\left(\left|\frac{X}{n} - p\right| < \epsilon\right) \geq 1 - \frac{p(1-p)}{n\epsilon^2} \quad (4.65)$$

which implies that

$$\lim_{n \rightarrow \infty} P\left(\left|\frac{X}{n} - p\right| < \epsilon\right) = 1 \quad (4.66)$$

for any given value of $\epsilon > 0$. Recalling that X denotes the observed number of successes in a sequence of Bernoulli trials, we conclude that as the number of trials, n , increases, the probability that the observed proportion of successes differs from p by less than any positive number ϵ (however small) approaches unity. Formula (4.66), known as *Bernoulli's theorem*, is a special case of the weak law of large numbers, which is discussed next.

Let X_1, X_2, \dots, X_n be n mutually independent identically distributed random variables. An n -tuple of values (x_1, x_2, \dots, x_n) , where x_i is a specific value of X_i , may be thought of as n independent measurements of some quantity that is distributed according to their common distribution. In this sense, we sometimes speak of the n -tuple (x_1, x_2, \dots, x_n) as a random sample of size n from this distribution.

Assume that the common distribution of these random variables has a finite mean μ . Then, for a sufficiently large value of n , we would expect that their arithmetic mean (or sample mean)

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

will be close to μ . Let $S_n = \sum_{i=1}^n X_i$ and the sample mean $\bar{X} = S_n/n$. If X_i has a finite variance σ^2 , then

$$\text{Var}[\bar{X}] = \text{Var}\left[\frac{S_n}{n}\right] = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}. \quad (4.67)$$

Thus $\text{Var}[\bar{X}]$ approaches 0 as n approaches infinity, implying that the distribution of \bar{X} becomes more concentrated about its mean μ . In fact, by applying Chebyshev's inequality to \bar{X} we obtain

$$P(|\bar{X} - \mu| \geq \delta) \leq \frac{\text{Var}[\bar{X}]}{\delta^2} = \frac{\sigma^2}{n\delta^2} \quad (4.68)$$

from which we get

$$\lim_{n \rightarrow \infty} P(|\bar{X} - \mu| \geq \delta) = 0. \quad (4.69)$$

Here number δ can be thought of as the desired accuracy in the approximation of μ by \bar{X} . Equation (4.69) assures us that no matter how small δ is, the probability that \bar{X} approximates μ to within δ converges to 1.

Equation (4.69) is known as the **weak law of large numbers**. Although our derivation required that the $\{X_i\}$ have finite variance, the law holds just under the assumption that the $\{X_i\}$ have finite mean.

For the final limit theorem, recall from Chapter 3 that sums of independent normal random variables are themselves normally distributed. The following **central-limit theorem** tells us that sums of independent random variables tend to be normally distributed even though the summands are not.

THEOREM 4.7 (The Central-Limit Theorem). Let X_1, X_2, \dots, X_n be independent random variables with a finite mean $E[X_i] = \mu_i$ and a finite variance $\text{Var}[X_i] = \sigma_i^2$ ($i = 1, 2, \dots, n$). We form the normalized random variable:

$$Z_n = \frac{\sum_{i=1}^n X_i - \sum_{i=1}^n \mu_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}} \quad (4.70)$$

so that $E[Z_n] = 0$ and $\text{Var}[Z_n] = 1$. Then, under certain regularity conditions, the limiting distribution of Z_n is standard normal, denoted $Z_n \rightarrow N(0, 1)$:

$$\lim_{n \rightarrow \infty} F_{Z_n}(t) = \lim_{n \rightarrow \infty} P(Z_n \leq t) = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy. \quad (4.71)$$

Example 4.20

As a special case of the central-limit theorem, assume that X_1, X_2, \dots, X_n are mutually independent and identically distributed with the common mean $\mu = E[X_i]$ and common variance $\sigma^2 = \text{Var}[X_i]$. Then equation (4.70) reduces to

$$Z_n = \frac{(\bar{X} - \mu)\sqrt{n}}{\sigma}, \quad (4.72)$$

where \bar{X} is the sample mean. Therefore, the sample mean from random samples (after standardization) tends toward normality as the sample size n increases.

#

The central-limit theorem should not be used indiscriminately, since there are distributions that do not obey it. For example, the Cauchy random variable X with pdf:

$$f(x) = \frac{1}{\pi(1+x^2)} \quad (4.73)$$

does not have a finite variance; hence the standard form of Z_n [of equation (4.70)] cannot be written.

It is difficult to give the value of n (the sample size) beyond which the normal approximation is accurate, since it depends on the form of the underlying distributions (F_{X_i}). Moderate sample sizes, such as 10, commonly are considered adequate.

Problems

1. The average CPU time per request is known to be 4.39 s for a compute server. We classify a request as a trivial request if it takes less than 1 s of CPU time, a moderate request if it takes between 1 and 5 s of CPU time, and a number-crunching request otherwise.

- (a) Obtain a bound on the probability that a given request is a number-crunching request.
- (b) Obtain a bound on the probability that a given request is not a trivial request.

Now, assume that the CPU time per request is exponentially distributed with mean 4.39 s. Recompute the two bounds.

- 2. Using the normal tables, plot $P(|X| \geq \delta)$ for $0 < \delta < 3$, where $X \sim N(0, 1)$. On the same graph, plot the upper bound on the abovementioned probability given by Chebyshev's inequality, and compare the two plots.
- 3. Consider a random variable X with the Cauchy pdf:

$$f(x) = \frac{1}{\pi(1+x^2)}, \quad -\infty < x < \infty.$$

- (a) Show that neither $E[X]$ nor $\text{Var}[X]$ exists in this case.
- (b) Show that the characteristic function is given by $N_X(\tau) = e^{-|\tau|}$.
- (c) Now consider $Z = \sum_{i=1}^n X_i$ where the $\{X_i\}$ are Cauchy and mutually independent. Thus, $N_Z(\tau) = e^{-n|\tau|}$, hence show that Z/n has the Cauchy distribution.

(Comment: Z/n is not Gaussian in the limit, since $\text{Var}[X_i]$ is not finite, and hence the central-limit theorem does not apply.)

- 4. Construct an example of a discrete random variable X that takes on each of the values $-b, 0, b$ with nonzero probability, so that the Chebyshev inequality becomes an equality when applied to the following expression:

$$P(|X - E[X]| \geq b).$$

In particular, determine $p_x(-b)$, $p_x(0)$, and $p_x(b)$.

5. * In order to represent a nonnegative real number X in a computer with finite precision, the number is either rounded to obtain X_r , or chopped (or truncated) to obtain X_c [STER 1974]. The representation errors in the two cases are bounded by

$$-\frac{1}{2} \leq Y_r = X - X_r \leq \frac{1}{2}$$

and

$$0 \leq Y_c = X - X_c < 1$$

(measured in the units of the last digit). It is common to assume that Y_r and Y_c are uniformly distributed over their respective ranges. Now assume two independent numbers X_1 and X_2 are being added. Compute the pdf, the mean, and the variance of the cumulative error in the sum $X_1 + X_2$ in cases of both rounding and chopping. Next assume that n mutually independent real numbers are to be added, each subject to rounding or chopping. What are the mean and the variance of the cumulative error in the two cases? Compare the mean with the worst-case errors. For $n = 4, 9, 16, 25, 36, 49, 100$, estimate the probability that the computed sum will differ from the sum of the original numbers by more than 0.5. [Hint: Use the central-limit theorem.]

Review Problems

1. A program has potentially N distinct input data sets indexed from 1 to N . Suppose the program is run on n randomly chosen data sets with repetition allowed. Let X be the largest index out of the n data sets used. Derive the pmf and the expected value of X . Assume that each of the N data sets is equally likely.

2. Given a **for** statement:

```
for (i = 1; i <= n; i++) { \(\ S \ \); }
```

Derive expressions for the distribution, the expected value, and the variance of the execution time T of the **for** loop, assuming that the distribution of the execution time of a single execution of the statement group S is known and that successive executions of S are independent.

3. Let the execution time X of a fixed instance of a problem using some randomized algorithm [WEID 1978] have the distribution function:

$$F(x) = x^\delta, 0 \leq x \leq 1 \quad \text{for some } \delta > 0.$$

If we ran the algorithm on that problem instance on a multiprocessor with two processors and ran the same algorithm on each one, the expected solution time would be equal to the expected value of the minimum of two independent random variables (denoted by Y), each having the distribution function F . Determine the conditions under which the speedup (defined by the ratio $E[X]/E[Y]$) exceeds the number of required processors; that is, under what conditions is $2E[Y] < E[X]$?

4. * Returning to the problem of adder design (Chapter 2, review problem 3), show that the expected length of the longest carry sequence is given by

$$\begin{aligned} E[V_n] &= \sum_{v=0}^n v[R_n(v) - R_n(v+1)] \\ &\leq \log_2 n. \end{aligned}$$

Thus, although in the worst case the length of a carry sequence can be as large as n , it is much smaller on the average. This fact can be used in speeding up the average addition time.

5. * Consider the representation error in storing a real number in a machine with m -digit base β normalized floating-point arithmetic [TSAO 1974]. Let the original mantissa X have the reciprocal pdf: $f_X(x) = 1/(x \ln \beta)$, $1/\beta \leq x < 1$. Let X_c and X_r denote the machine representations of X assuming chopping and rounding, respectively. Then the respective (relative) representation errors Δ_c and Δ_r are given by

$$\Delta_c = \frac{X - X_c}{X} \text{ and } \Delta_r = \frac{X - X_r}{X}.$$

Assuming that the absolute error $Y_c = X - X_c$ is a continuous random variable, uniformly distributed over the interval $(0, \beta^{-m})$, and that Y_c is independent of X (a questionable assumption), show that the pdf of the relative representation error Δ_c is given by

$$f_{\Delta_c}(\delta) = \begin{cases} \frac{\beta^{m-1}(\beta-1)}{\ln \beta}, & \beta^{-m} > \delta \geq 0, \\ \frac{1/\delta - \beta^{m-1}}{\ln \beta}, & \beta^{1-m} > \delta \geq \beta^{-m}. \end{cases}$$

Similarly, assuming that the random variable $Y_r = X - X_r$ is uniformly distributed over $(-\beta^{-m}/2, \beta^{-m}/2)$ and that Y_r and X are independent, show that

$$f_{\Delta_r}(\delta) = \begin{cases} \frac{\beta^{m-1}(\beta-1)}{\ln \beta}, & |\delta| \leq \frac{\beta^{-m}}{2}, \\ \frac{\frac{1}{2|\delta|} - \beta^{m-1}}{\ln \beta}, & \frac{\beta^{-m}}{2} < |\delta| < \frac{\beta^{1-m}}{2}. \end{cases}$$

Plot the two densities and compute the average representation errors $E[\Delta_c]$ and $E[\Delta_r]$. Compare these with the respective maximum representation errors β^{-m+1} and $\frac{1}{2}\beta^{-m+1}$. Also compute the variances of Δ_c and Δ_r .

REFERENCES

- [ALME 1996] V. A. F. Almeida, A. Bestavros, M. E. Crovella, and A. de Oliveira, "Characterizing reference locality in the WWW," *IEEE Int. Conf. Parallel and Distributed Information Systems*, Dec. 1996.

- [BLAK 1979] I. F. Blake, *An Introduction to Applied Probability*, Wiley, New York, 1979.
- [BRES 1999] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and Zipf-like distributions: Evidence and implications,” *Proc. Infocom’99*, March 1999.
- [HUNT 1980] D. Hunter, *Modeling Real DASD Configurations*, IBM T. J. Watson Center Report, RC-8606, Yorktown Heights, NY, 1980.
- [IBM 1997] IBM OEM hard disk drive specifications for DTCA-23240/24090, *2.5-Inch Hard Disk Drive with ATA Interface*, revision 3.0, 1997.
- [KNUT 1997] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*, 3rd ed., Addison-Wesley, Reading, MA, 1997.
- [KOBA 1978] H. Kobayashi, *Modeling and Analysis*, Addison-Wesley, Reading, MA, 1978.
- [LEE 1993] I. Lee, D. Tang, R. K. Iyer, and M.-C. Hsueh, “Measurement-based evaluation of operating system fault tolerance,” *IEEE Trans. Reliability*, **42**(2), 238–249 (June 1993).
- [LEEM 1995] L. M. Leemis, *Reliability: Probabilistic Models and Statistical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [PEEB 1980] P. Z. Peebles, *Probability, Random Variables, and Random Signal Principles*, McGraw-Hill, New York, 1980.
- [SEDG 1977] R. Sedgewick, “Permutation generation methods,” *ACM Comput. Surv.*, **9**(2), 137–164 (June 1977).
- [STER 1974] P. H. Sterbenz, *Floating-Point Computation*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [TSAO 1974] N.-K. Tsao, “On the distributions of significant digits and roundoff errors,” *Commun. ACM*, **17**(5), 269–271 (1974).
- [WEID 1978] B. Weide, *Statistical Methods in Algorithm Design and Analysis*, Ph.D. thesis, Department of Computer Science, Carnegie-Mellon Univ. Pittsburgh, PA, 1978.
- [WILL 1996] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. Fox, “Removal policies in network caches for WWW documents,” *ACM SIGCOMM’96*, Aug. 1996, pp. 293–305.
- [WIRT 1976] N. Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [WOLM 1965] E. Wolman, “A fixed optimum cell-size for records of variable length,” *J. ACM* **12**, 53–70 (1965).
- [ZIPF 1949] G. K. Zipf, *Human Behavior and the Principle of Least Effort, An Introduction to Human Ecology*, Addison-Wesley, Reading, MA, 1949.

Chapter 5

Conditional Distribution and Expectation

5.1 INTRODUCTION

We have seen that if two random variables are independent, then their joint distribution can be determined from their marginal distribution functions. In the case of dependent random variables, however, the joint distribution can not be determined in this simple fashion. This leads us to the notions of conditional pmf, conditional pdf, and conditional distribution.

Recalling the definition of conditional probability, $P(A|B)$, for two events A and B , we can define the **conditional probability** $P(A|X = x)$ of event A , given that the event $[X = x]$ has occurred, as

$$P(A|X = x) = \frac{P(A \text{ occurs and } X = x)}{P(X = x)} \quad (5.1)$$

whenever $P(X = x) \neq 0$. In Chapter 3 we noted that if X is a continuous random variable, then $P(X = x) = 0$ for all x . In this case, definition (5.1) is not satisfactory. On the other hand, if X is a discrete random variable, then definition (5.1) is adequate, as shown in the following definition.

Definition (Conditional pmf). Let X and Y be discrete random variables having a joint pmf $p(x, y)$. The conditional pmf of Y given X is

$$\begin{aligned} p_{Y|X}(y|x) &= P(Y = y | X = x) \\ &= \frac{P(Y = y, X = x)}{P(X = x)} \end{aligned} \quad (5.2)$$

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

$$= \frac{p(x, y)}{p_x(x)},$$

if $p_x(x) \neq 0$.

Note that the conditional pmf, as defined above, satisfies properties (p1)–(p3) of a pmf, discussed in Chapter 2. Rewriting the above definition another way, we have

$$p(x, y) = p_x(x)p_{Y|X}(y|x) = p_Y(y)p_{X|Y}(x|y). \quad (5.3)$$

This is simply another form of the multiplication rule (of Chapter 1), and it gives us a way to compute the joint pmf regardless of whether X and Y are independent. If X and Y are independent, then from (5.3) and the definition of independence (in Chapter 2) we conclude that

$$p_{Y|X}(y|x) = p_Y(y). \quad (5.4)$$

From (5.3) we also have the following marginal probability:

$$p_Y(y) = \sum_{\text{all } x} p(x, y) = \sum_{\text{all } x} p_{Y|X}(y|x)p_X(x). \quad (5.5)$$

This is another form of the theorem of total probability of Chapter 1.

We can also define the conditional distribution function $F_{Y|X}(y|x)$ of a random variable Y , given a discrete random variable X by

$$F_{Y|X}(y|x) = P(Y \leq y | X = x) = \frac{P(Y \leq y \text{ and } X = x)}{P(X = x)} \quad (5.6)$$

for all values of y and for all values of x such that $P(X = x) > 0$. Definition (5.6) applies even for the case when Y is not discrete.

Note that the conditional distribution function can be obtained from the conditional pmf (assuming that both X and Y are discrete):

$$F_{Y|X}(y|x) = \frac{\sum_{t \leq y} p(x, t)}{p_x(x)} = \sum_{t \leq y} p_{Y|X}(t|x). \quad (5.7)$$

Example 5.1

A server cluster has two servers labeled A and B. Incoming jobs are independently routed by the front end equipment (called server switch) to server A with probability p and to server B with probability $(1 - p)$. The number of jobs, X , arriving per unit time is Poisson distributed with parameter λ . Determine the distribution function of the number of jobs, Y , received by server A, per unit time.

Let us determine the conditional probability of the event $[Y = k]$ given that event $[X = n]$ has occurred. Note that routing of the n jobs can be thought of as a sequence of n independent Bernoulli trials. Hence, the conditional probability that

$[Y = k]$ given $[X = n]$ is binomial with parameters n and p :

$$p_{Y|X}(k|n) = \begin{cases} P(Y = k|X = n) = \binom{n}{k} p^k (1-p)^{n-k}, & 0 \leq k \leq n \\ 0, & \text{otherwise.} \end{cases}$$

Recalling that $P(X = n) = e^{-\lambda} \lambda^n / n!$ and using formula (5.5), we get

$$\begin{aligned} p_Y(k) &= \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} \frac{\lambda^n e^{-\lambda}}{n!} \\ &= \frac{(\lambda p)^k e^{-\lambda}}{k!} \sum_{n=k}^{\infty} \frac{(\lambda(1-p))^{n-k}}{(n-k)!} \\ &= \frac{(\lambda p)^k e^{-\lambda}}{k!} e^{\lambda(1-p)} \end{aligned}$$

(since the last sum is the Taylor series expansion of $e^{\lambda(1-p)}$) and

$$p_Y(k) = \frac{(\lambda p)^k e^{-\lambda p}}{k!}.$$

Thus, Y is Poisson distributed with parameter λp . For this reason we often say that the Poisson distribution is preserved under random selection.

‡

Example 5.2

As a related application of Example 5.1, we consider the testing process for a software product. Let $N(t)$ be the number of faults detected up to time t and let the initial number of faults M be Poisson distributed with parameter λ . Let us label these M faults $1, 2, \dots, M$. Further let X_i be the time to detection of the fault labeled i . Assume that X_1, X_2, \dots are mutually independent and identically distributed random variables with the common distribution function $F(t)$. Since $P(X_i \leq t) = F(t)$ for all i , it follows that the probability for a specific fault i to have been detected by time t is $p = F(t)$. Hence

$$P[N(t) = j | M = m] = \binom{m}{j} [F(t)]^j [1 - F(t)]^{m-j}.$$

By our assumption, M is Poisson distributed with the probability mass function:

$$P[M = m] = \frac{(\lambda)^m e^{-\lambda}}{m!},$$

hence, we have

$$P[N(t) = j] = \frac{[\lambda F(t)]^j e^{-\lambda F(t)}}{j!}.$$

‡

If X and Y are jointly continuous, then we define the conditional pdf of Y given X in a way analogous to the definition of the conditional pmf.

Definition (Conditional pdf). Let X and Y be continuous random variables with joint pdf $f(x, y)$. The conditional density $f_{Y|X}$ is defined by

$$f_{Y|X}(y|x) = \frac{f(x, y)}{f_X(x)}, \quad \text{if } 0 < f_X(x) < \infty. \quad (5.8)$$

It can be easily verified that the function defined in (5.8) satisfies properties (f1) and (f2) of a pdf.

It follows from the definition of conditional density that

$$f(x, y) = f_X(x)f_{Y|X}(y|x) = f_Y(y)f_{X|Y}(x|y). \quad (5.9)$$

This is the continuous analog of the multiplication rule (MR) of Chapter 1. If X and Y are independent, then

$$f(x, y) = f_X(x)f_Y(y),$$

which implies that

$$f_{Y|X}(y|x) = f_Y(y). \quad (5.10)$$

Conversely, if equation (5.10) holds, then it follows that X and Y are independent random variables. Thus (5.10) is a necessary and sufficient condition for two random variables X and Y , which have a joint density, to be independent.

From the expression of joint density (5.9), we can obtain an expression for the marginal density of Y in terms of conditional density by integration:

$$\begin{aligned} f_Y(y) &= \int_{-\infty}^{\infty} f(x, y) dx \\ &= \int_{-\infty}^{\infty} f_X(x)f_{Y|X}(y|x) dx. \end{aligned} \quad (5.11)$$

This is the continuous analog of the theorem of total probability.

Further, in the definition of conditional density, we can reverse the role of X and Y to define (whenever $f_Y(y) > 0$):

$$f_{X|Y}(x|y) = \frac{f(x, y)}{f_Y(y)}.$$

Using the expression (5.11) for $f_Y(y)$ and noting that $f(x, y) = f_X(x)f_{Y|X}(y|x)$, we obtain

$$f_{X|Y}(x|y) = \frac{f_X(x)f_{Y|X}(y|x)}{\int_{-\infty}^{\infty} f_X(x)f_{Y|X}(y|x) dx}. \quad (5.12)$$

This is the continuous analog of Bayes' rule discussed in Chapter 1.

The conditional pdf can be used to obtain the conditional probability:

$$P(a \leq Y \leq b | X = x) = \int_a^b f_{Y|X}(y|x) dy, \quad a \leq b. \quad (5.13)$$

In particular, the conditional distribution function $F_{Y|X}(y|x)$ is *defined*, analogous to (5.6), as

$$\begin{aligned} F_{Y|X}(y|x) &= P(Y \leq y | X = x) = \frac{\int_{-\infty}^y f(x,t) dt}{f_X(x)} \\ &= \int_{-\infty}^y f_{Y|X}(t|x) dt. \end{aligned} \quad (5.14)$$

As motivation for definition (5.14) we observe that

$$\begin{aligned} F_{Y|X}(y|x) &= \lim_{h \rightarrow 0} P(Y \leq y | x \leq X \leq x + h) \\ &= \lim_{h \rightarrow 0} \frac{P(x \leq X \leq x + h \text{ and } Y \leq y)}{P(x \leq X \leq x + h)} \\ &= \lim_{h \rightarrow 0} \frac{\int_x^{x+h} \int_{-\infty}^y f(s,t) dt ds}{\int_x^{x+h} f_X(s) ds}. \end{aligned}$$

For some x_1^*, x_2^* with $x \leq x_1^*, x_2^* \leq x + h$, we obtain

$$F_{Y|X}(y|x) = \lim_{h \rightarrow 0} \frac{h \int_{-\infty}^y f(x_1^*, t) dt}{h f_X(x_2^*)}$$

(by the mean value theorem of integrals)

$$= \lim_{h \rightarrow 0} \frac{\int_{-\infty}^y f(x_1^*, t) dt}{f_X(x_2^*)}$$

(since both x_1^* and x_2^* approach x as h approaches 0)

$$\begin{aligned} &= \int_{-\infty}^y \frac{f(x,t)}{f_X(x)} dt \\ &= \int_{-\infty}^y f_{Y|X}(t|x) dt. \end{aligned}$$

Example 5.3

In modeling software reliability during the testing phase we are interested in deriving the conditional reliability defined as a conditional survivor function associated with i th failure, given that the software has experienced $(i-1)$ failures

$$R_i(t) = P(T_i > t),$$

where T_i , known as interfailure time, defines the time between the occurrence of $(i-1)$ st and i th failure.

A number of software reliability growth models are based on the assumption that interfailure times T_1, T_2, \dots, T_i are independent exponentially distributed random variables with failure rate λ_i which changes at each failure occurrence due to the fault removal attempts.

The Littlewood–Verrall model [LITT 1973] assumes that interfailure times are independent random variables with conditional density

$$f_{T_i|\Lambda_i}(t|\lambda_i) = \lambda_i e^{-\lambda_i t},$$

that is

$$R_i(t|\lambda_i) = P(T_i > t | \Lambda_i = \lambda_i) = e^{-\lambda_i t}.$$

The failure rate Λ_i has gamma pdf with shape parameter α and scale parameter $\psi(i)$:

$$f_{\Lambda_i}(\lambda_i) = \frac{(\psi(i))^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\psi(i)\lambda_i}.$$

By the continuous version of the theorem of total probability [equation (5.11)], it follows that

$$\begin{aligned} R_i(t) &= \int_0^\infty P(T_i > t | \Lambda_i = \lambda_i) f_{\Lambda_i}(\lambda_i) d\lambda_i \\ &= \int_0^\infty e^{-\lambda_i t} \frac{(\psi(i))^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\psi(i)\lambda_i} d\lambda_i \\ &= \frac{(\psi(i))^\alpha}{\Gamma(\alpha)} \int_0^\infty \lambda_i^{\alpha-1} e^{-(\psi(i)+t)\lambda_i} d\lambda_i, \end{aligned}$$

and hence, using formula (3.25), we obtain

$$R_i(t) = \frac{(\psi(i))^\alpha}{\Gamma(\alpha)} \cdot \frac{\Gamma(\alpha)}{[\psi(i) + t]^\alpha} = \left(\frac{\psi(i)}{\psi(i) + t} \right)^\alpha$$

which is the survivor function of the Pareto distribution.

The function $\psi(i)$ is supposed to reflect the quality of the testing efforts. For example, $\psi(i) = \beta_0 + \beta_1 i$ ensures that the expected value of failure rate (see Section 4.5.8) $E[\Lambda_i] = \alpha/\psi(i)$ decreases with i . Consequently, for $i \geq 2$ the sequence of failure rates Λ_i form a stochastically decreasing sequence, that is, for any $\lambda \geq 0$, $P(\Lambda_i \leq \lambda) \geq P(\Lambda_{i-1} \leq \lambda)$. This reflects the likelihood, but not a guarantee, that a fault removal will improve reliability and if an improvement does take place it would be of uncertain magnitude.

‡

Example 5.4

Consider a series system of two independent components with respective lifetime distributions $X \sim \text{EXP}(\lambda_1)$ and $Y \sim \text{EXP}(\lambda_2)$. We wish to determine the probability that component 2 is the cause of system failure. Let A denote the event that component 2 is the cause of system failure; then

$$P(A) = P(X \geq Y).$$

To compute this probability, first consider the conditional distribution function:

$$F_{X|Y}(t|t) = P(X \leq t | Y = t) = F_X(t)$$

(by the independence of X and Y). Now by the continuous version of the theorem of total probability, we obtain

$$\begin{aligned} P(A) &= \int_0^\infty P(X \geq t | Y = t) f_Y(t) dt \\ &= \int_0^\infty [1 - F_X(t)] f_Y(t) dt \\ &= \int_0^\infty e^{-\lambda_1 t} \lambda_2 e^{-\lambda_2 t} dt \\ &= \frac{\lambda_2}{\lambda_1 + \lambda_2}. \end{aligned}$$

This result generalizes to a series system of n independent components, each with a respective constant failure rate λ_j ($j = 1, 2, \dots, n$). The probability that the j th component is the cause of system failure is given by

$$\frac{\lambda_j}{\sum_{i=1}^n \lambda_i}. \quad (5.15)$$

‡

Example 5.5 [BARD 1975]

Thus far in our reliability computations, we have considered failure mechanisms of components to be independent. We have derived the exponential lifetime distribution from a Poisson shock model. We now model the behavior of a system of two non-independent components using a bivariate exponential distribution. Assume three independent Poisson shock sources. A shock from source 1 destroys component 1, and the time to the occurrence U_1 of such a shock is exponentially distributed with parameter λ_1 , so that $P(U_1 > t) = e^{-\lambda_1 t}$. A shock from source 2 destroys component 2, and $P(U_2 > t) = e^{-\lambda_2 t}$. Finally, a shock from source 3 destroys both components and it occurs at random time U_{12} , so that $P(U_{12} > t) = e^{-\lambda_{12} t}$. Thus the lifetime X of component 1 satisfies

$$X = \min\{U_1, U_{12}\}$$

and is exponentially distributed with parameter $\lambda_1 + \lambda_{12}$. The lifetime Y of component 2 is given by

$$Y = \min\{U_2, U_{12}\}$$

and is exponentially distributed with parameter $\lambda_2 + \lambda_{12}$. Therefore

$$f_X(x) = (\lambda_1 + \lambda_{12})e^{-(\lambda_1 + \lambda_{12})x}, \quad x > 0,$$

and

$$f_Y(y) = (\lambda_2 + \lambda_{12})e^{-(\lambda_2 + \lambda_{12})y}, \quad y > 0.$$

To compute the joint distribution function $F(x, y) = P(X \leq x, Y \leq y)$, we first compute the following:

$$\begin{aligned} R(x, y) &= P(X > x, Y > y) \\ &= P(\min\{U_1, U_{12}\} > x, \min\{U_2, U_{12}\} > y) \\ &= P(U_1 > x, U_{12} > \max\{x, y\}, U_2 > y) \\ &= P(U_1 > x)P(U_{12} > \max\{x, y\})P(U_2 > y) \\ &= e^{-\lambda_1 x - \lambda_2 y - \lambda_{12} \max\{x, y\}}, \quad x \geq 0, y \geq 0. \end{aligned}$$

This is true since U_1 , U_2 , and U_{12} are mutually independent. It is interesting to note that $R(x, y) \geq R_X(x)R_Y(y)$. Now $F(x, y)$ can be obtained using the following relation (see Figure 5.1):

$$\begin{aligned} F(x, y) &= R(x, y) + F_X(x) + F_Y(y) - 1 \\ &= 1 + e^{-\lambda_1 x - \lambda_2 y - \lambda_{12} \max\{x, y\}} - e^{-(\lambda_1 + \lambda_{12})x} - e^{-(\lambda_2 + \lambda_{12})y}. \end{aligned}$$

In particular

$$F(x, y) \neq F_X(x)F_Y(y)$$

since

$$F_X(x)F_Y(y) = 1 - e^{-(\lambda_1 + \lambda_{12})x} - e^{-(\lambda_2 + \lambda_{12})y} + e^{-(\lambda_1 + \lambda_{12})x - (\lambda_2 + \lambda_{12})y}.$$

Thus X and Y are indeed *dependent* random variables.

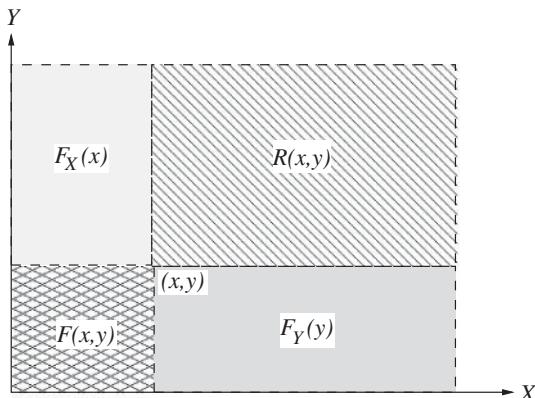


Figure 5.1. Illustration for $R(x, y) + F_X(x) + F_Y(y) = 1 + F(x, y)$

The joint density $f(x, y)$ may be obtained by taking partial derivatives:

$$\begin{aligned} f(x, y) &= \frac{\partial^2 F(x, y)}{\partial x \partial y} \\ &= \begin{cases} \lambda_1(\lambda_2 + \lambda_{12})e^{-\lambda_1 x - \lambda_2 y - \lambda_{12} y}, & x \leq y, \\ \lambda_2(\lambda_1 + \lambda_{12})e^{-\lambda_1 x - \lambda_2 y - \lambda_{12} x}, & x > y, \end{cases} \end{aligned}$$

and the conditional density by

$$f_{Y|X}(y|x) = \begin{cases} \frac{\lambda_1(\lambda_2 + \lambda_{12})}{\lambda_1 + \lambda_{12}} e^{-(\lambda_2 + \lambda_{12})y + \lambda_{12}x}, & x \leq y, \\ \lambda_2 e^{-\lambda_2 y}, & x > y. \end{cases}$$

Once again, this confirms that X and Y are not independent.

‡

For further discussion of such dependencies see the paper by Muppala et al. [MUPP 1996].

Problems

- Consider a fault-tolerant multiprocessor computer system with two processors, each with its own private memory module and a third memory module that is shared between the processors. Supposing that the processors and the memory modules have constant failure rates λ_p and λ_m , respectively, and the computer system is operational as long as there is at least one operational processor with access to either a private or shared memory module, determine the failure-time distribution for the computer system. Recompute the failure-time distribution of the computer system, this time, assuming a Weibull distribution for time to failure of the processors and memory modules with the different λ parameters and but the same shape parameter α . [Hint: Use conditioning (or factoring) on the event that the shared memory is operational or not.]
- Consider again the problem of 1M (1-megabyte) RAM chips supplied by two semiconductor houses (problem 1, Section 3.6). Determine the conditional probability density of the lifetime X , given that the lifetime Y does not exceed 10^6 h.
- Consider the operation of an online file updating system [MEND 1979]. Let p_i be the probability that a transaction inserts a record into file i ($i = 1, 2, \dots, n$), so that $\sum_{i=1}^n p_i = 1$. The record size (in bytes) of file i is a random variable denoted by Y_i . Determine
 - The average number of bytes added to file i per transaction.
 - The variance of the number of bytes added to file i per transaction.

[Hint: You may define the Bernoulli random variable:

$$A_i = \begin{cases} 1, & \text{transaction updates file } i, \\ 0, & \text{otherwise,} \end{cases}$$

and let the random variable $V_i = A_i Y_i$ be the number of bytes added to file i in a transaction.]

4. X_1 and X_2 are independent random variables with Poisson distributions, having respective parameters α_1 and α_2 . Show that the conditional pmf of X_1 , given $X_1 + X_2, p_{X_1|X_1+X_2}(X_1 = x_1|X_1 + X_2 = y)$, is binomial. Determine its parameters.
5. Let the execution times X and Y of two independent parallel processes be uniformly distributed over $(0, t_X)$ and $(0, t_Y)$, respectively, with $t_X \leq t_Y$. Find the probability that the former process finishes execution before the latter.

5.2 MIXTURE DISTRIBUTIONS

The definition of conditional density (and conditional pmf) can be naturally extended to the case where X is a discrete random variable and Y is a continuous random variable (or vice versa).

Example 5.6

Consider a file server whose workload may be divided into r distinct classes. For job class i ($1 \leq i \leq r$), the CPU service time is exponentially distributed with parameter λ_i . Let Y denote the service time of a job and let X be the job class. Then

$$f_{Y|X}(y|i) = \lambda_i e^{-\lambda_i y}, \quad y > 0.$$

Now let α_i (≥ 0) be the probability that a randomly chosen job belongs to class i :

$$p_X(i) = \alpha_i, \quad \sum_{i=1}^r \alpha_i = 1.$$

Then the joint density is

$$\begin{aligned} f(i, y) &= f_{Y|X}(y|i)p_X(i) \\ &= \alpha_i \lambda_i e^{-\lambda_i y}, \quad y > 0, \end{aligned}$$

and the marginal density is

$$\begin{aligned} f_Y(y) &= \sum_{i=1}^r f(i, y) \\ &= \sum_{i=1}^r \alpha_i f_{Y|X}(y|i) \\ &= \sum_{i=1}^r \alpha_i \lambda_i e^{-\lambda_i y}, \quad y > 0. \end{aligned}$$

Thus Y has an r -stage hyperexponential distribution, denoted by a set of parallel exponential stages (or phases) as in Figure 5.2.

#

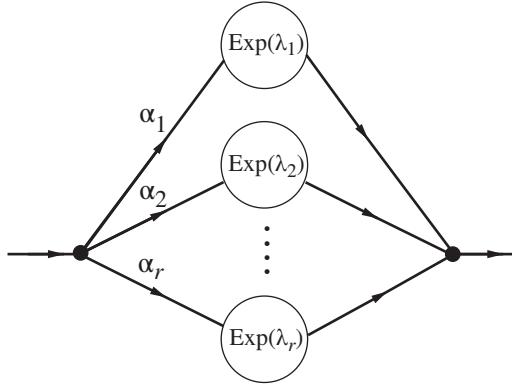


Figure 5.2. The hyperexponential distribution as a set of parallel exponential stages

Of course, the conditional distribution of Y does not have to be exponential. In general, if we let

$$f_{Y|X}(y|i) = f_i(y) = f_{Y_i}(y)$$

and

$$F_{Y|X}(y|i) = F_i(y),$$

then we have the unconditional pdf of Y

$$f_Y(y) = \sum_{i=1}^r \alpha_i f_i(y), \quad (5.16)$$

and the unconditional CDF of Y

$$F_Y(y) = \sum_{i=1}^r \alpha_i F_i(y). \quad (5.17)$$

Taking Laplace–Stieltjes transforms on both sides of (5.16), we also have

$$L_Y(s) = \sum_{i=1}^r \alpha_i L_{Y_i}(s). \quad (5.18)$$

Finally, applying the definitions of the mean and higher moments to (5.16), we have

$$E[Y] = \sum_{i=1}^r \alpha_i E[Y_i], \quad (5.19)$$

$$E[Y^k] = \sum_{i=1}^r \alpha_i E[Y_i^k]. \quad (5.20)$$

Such mixture distributions often arise in a number of reliability situations. For example, suppose that a manufacturer produces α_i fraction of a certain product in assembly line i , and the lifetime of a unit produced in assembly line i has a distribution F_i . Now if the outputs of the assembly lines are merged, then a randomly chosen unit from the merged stream will possess the lifetime distribution given by equation (5.17).

Example 5.7

Assume that in a mixture of two groups, one group consists of components in the chance-failure period (with constant hazard rate λ_1) and the other of aging items (modeled by an r -stage Erlang lifetime distribution with parameter λ_2). If α is the fraction of group 1 components, then the distribution of the lifetime Y of a component from the merged stream is given by

$$F_Y(y) = \alpha(1 - e^{-\lambda_1 y}) + (1 - \alpha) \left(1 - \sum_{k=0}^{r-1} \frac{(\lambda_2 y)^k}{k!} e^{-\lambda_2 y} \right)$$

and

$$f_Y(y) = \alpha \lambda_1 e^{-\lambda_1 y} + (1 - \alpha) \frac{\lambda_2^r y^{r-1}}{(r-1)!} e^{-\lambda_2 y}.$$

This density and the corresponding hazard rate are shown in Figures 5.3 and 5.4. Note that this distribution has a nonmonotonic hazard function.

#

More generally, the distributions being mixed may be uncountably infinite in number; that is, X may be a continuous random variable. For instance, the lifetime of a product may depend on the amount X of impurity present in the raw material. Let the conditional distribution of the lifetime Y be given by

$$F_{Y|X}(y|x) = G_X(y) = \int_{-\infty}^y \frac{f(x,t)}{f_X(x)} dt$$

where the impurity X has a density function $f_X(x)$. Then the resultant lifetime distribution F_Y is given by

$$F_Y(y) = \int_{-\infty}^{\infty} f_X(x) G_X(y) dx = \int_{-\infty}^{\infty} \int_{-\infty}^y f(x,t) dt dx.$$

In the next example we let Y be discrete and X continuous.

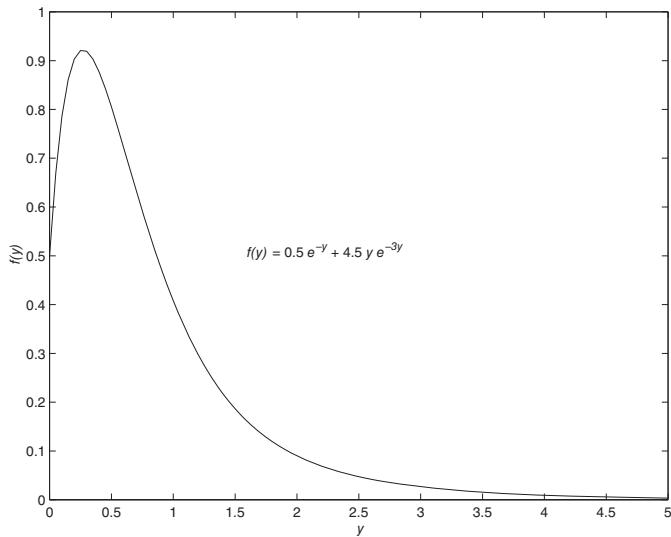


Figure 5.3. The pdf of a mixture of exponential and Erlang

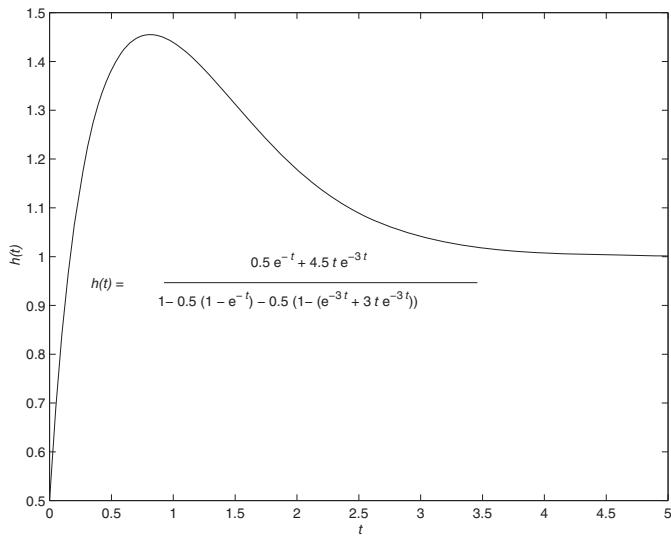


Figure 5.4. Hazard rate of a mixture of exponential and Erlang

Example 5.8 [CLAR 1970]

Let X be the service time of a request to a Web server and let it be exponentially distributed with parameter μ , so that

$$f_X(x) = \mu e^{-\mu x}, \quad x > 0.$$

Let the number of requests arriving in the interval $(0, t]$ be Poisson distributed with parameter λt . Finally, let Y be the number of requests arriving while one is being served.

If we fix the value of X to be x , the Poisson arrival assumption can be used to obtain the conditional pmf of Y given $[X = x]$:

$$\begin{aligned} p_{Y|X}(y|x) &= P(Y = y | X = x) \\ &= e^{-\lambda x} \frac{(\lambda x)^y}{y!}, \quad y = 0, 1, 2, \dots \end{aligned}$$

The joint probability density function of X and Y is then given by

$$\begin{aligned} f(x, y) &= f_X(x)p_{Y|X}(y|x) \\ &= \frac{\mu e^{-(\lambda+\mu)x} (\lambda x)^y}{y!}, \quad y = 0, 1, 2, \dots; \quad x > 0. \end{aligned}$$

The unconditional (or marginal) pmf of Y can now be obtained by integration:

$$\begin{aligned} p_Y(y) &= P(Y = y) \\ &= \int_0^\infty f(x, y) dx \\ &= \frac{\mu \lambda^y}{y!} \int_0^\infty e^{-(\lambda+\mu)x} (x)^y dx. \end{aligned}$$

Substituting $(\lambda + \mu)x = w$, we get

$$\begin{aligned} p_Y(y) &= \frac{\mu \lambda^y}{y!(\lambda + \mu)^{y+1}} \int_0^\infty e^{-w} w^y dw \\ &= \frac{\mu \lambda^y y!}{y!(\lambda + \mu)^{y+1}} \end{aligned}$$

[since the last integral is equal to $\Gamma(y + 1) = y!$ by formulas (3.21) and (3.23)]. Thus

$$\begin{aligned} p_Y(y) &= \frac{\rho^y}{(1 + \rho)^{y+1}}, \quad \text{where } \rho = \frac{\lambda}{\mu} \\ &= \left(\frac{\rho}{1 + \rho} \right)^y \frac{1}{1 + \rho}, \quad y = 0, 1, 2, \dots \end{aligned}$$

Thus Y has a modified geometric pmf with parameter $1/(1 + \rho)$; hence the expected value is

$$E[Y] = \frac{\rho/(1 + \rho)}{1/(1 + \rho)} = \rho = \frac{\lambda}{\mu}.$$

This is an example of the so-called $M/M/1$ queuing system to be discussed in a later chapter. We may argue that an undesirable backlog of customers will not occur provided the average number of customers arriving in the interval representing the service time of a typical customer is less than 1. In other words, the queuing system will remain stable provided

$$E[Y] = \rho < 1 \quad \text{or} \quad \lambda < \mu.$$

This last condition says that the rate at which requests arrive is less than the rate at which work can be completed.

‡

Example 5.9 [GAVE 1973]

Consider a series system with n independent components, each with a lifetime distribution function $G(t)$ and density $g(t)$. Because of the options offered, the number of components, Y , in a specific system is a random variable. Let X denote the lifetime of the series system. Then, clearly

$$\begin{aligned} F_{X|Y}(t|n) &= 1 - [1 - G(t)]^n, \quad n = 0, 1, 2, \dots, \quad t > 0, \\ f_{X|Y}(t|n) &= n[1 - G(t)]^{n-1}g(t), \quad n = 0, 1, 2, \dots, \quad t > 0. \end{aligned}$$

Assume that the number of components, Y , has a Poisson pmf with parameter α . Then

$$p_Y(n) = e^{-\alpha} \frac{\alpha^n}{n!}, \quad \alpha > 0, \quad n = 0, 1, 2, \dots,$$

and the joint density is

$$\begin{aligned} f(t, n) &= f_{X|Y}(t|n)p_Y(n) \\ &= \begin{cases} e^{-\alpha} \frac{\alpha^n}{(n-1)!} [1 - G(t)]^{n-1}g(t), & t > 0, \quad n = 0, 1, 2, \dots, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

We can now determine the marginal density:

$$f_X(t) = \sum_{n=1}^{\infty} [1 - G(t)]^{n-1} g(t) e^{-\alpha} \frac{\alpha^n}{(n-1)!}.$$

The system reliability is then given by

$$\begin{aligned} R_X(t) &= P(X > t) \\ &= \sum_{n=0}^{\infty} [1 - F_{X|Y}(t|n)] p_Y(n) \end{aligned}$$

(by the theorem of total probability)

$$\begin{aligned}
 &= \sum_{n=0}^{\infty} [1 - G(t)]^n e^{-\alpha} \frac{\alpha^n}{n!} \\
 &= e^{-\alpha} \sum_{n=0}^{\infty} \frac{\{\alpha[1 - G(t)]\}^n}{n!} \\
 &= e^{-\alpha} e^{\alpha[1 - G(t)]} \\
 &= e^{-\alpha G(t)}.
 \end{aligned}$$

Now suppose that the system has survived until time t . We are interested in computing the conditional pmf of the number of components Y that it has

$$\begin{aligned}
 P(Y = n \mid X > t) &= \frac{P(X > t, Y = n)}{P(X > t)} \\
 &= \frac{[1 - F_{X|Y}(t|n)] p_Y(n)}{R_X(t)} \\
 &= e^{-\alpha[1 - G(t)]} \frac{[\alpha(1 - G(t))]^n}{n!}.
 \end{aligned}$$

Thus the conditional pmf of Y , given that no failure has occurred until time t , is Poisson with parameter $\alpha[1 - G(t)]$. Since $G(t)$ is a monotonically increasing function of t , the parameter of the Poisson pmf decreases with t . In other words, the longer the system survives, the greater is the evidence that it has a small number of components.

#

Example 5.10

In the previous example, we note that

$$\lim_{t \rightarrow \infty} R_X(t) = e^{-\alpha} \neq 0.$$

In other words X is a defective random variable and hence, $E[X]$ does not exist. The primary reason is that $p_Y(0) = e^{-\alpha} \neq 0$. In other words, there is a nonzero probability that the number of components n in the series system can be equal to zero. We can remove this possibility by considering a modified Poisson pmf for the number of components Y so that

$$p_Y(n) = \frac{e^{-\alpha}}{1 - e^{-\alpha}} \frac{\alpha^n}{n!}, \quad \alpha > 0, \quad n = 1, 2, \dots$$

#

Yet another case of a mixture distribution occurs when we mix two distributions, one discrete and the other continuous. The mixture distribution then represents a mixed random variable [see the distribution (3.2) in Chapter 3].

Problems

1. Consider the **if** statement:

```
if B {S1; } else {S2; }
```

Let the random variables X_1 and X_2 respectively, denote the execution times of the statement groups S_1 and S_2 . Assuming the probability that the Boolean expression $B = \text{true}$ is p , derive an expression for the distribution of the total execution time X of the **if** statement. Compute $E[X]$ and $\text{Var}[X]$ as functions of the means and variances of X_1 and X_2 . Generalize your results to a case statement with k clauses.

2. Describe a method of generating a random deviate of a two-stage hyperexponential distribution.
3. One of the inputs to a certain program is a random variable whose value is a nonnegative real number; call it Λ . The probability density function of Λ is given by

$$f_{\Lambda}(\lambda) = \lambda e^{-\lambda}, \quad \lambda > 0.$$

Conditioned on $\Lambda = \lambda$, the execution time of the program is an exponentially distributed random variable with parameter λ . Compute the distribution function of the program execution time X .

5.3 CONDITIONAL EXPECTATION

If X and Y are continuous random variables, then the conditional density $f_{Y|X}$ is given by formula (5.8). Since $f_{Y|X}$ is a density of a continuous random variable, we can talk about its various moments. Its mean (if it exists) is called the **conditional expectation** of Y given $[X = x]$ and will be denoted by $E[Y|X = x]$ or $E[Y|x]$. Thus

$$\begin{aligned} E[Y|x] &= \int_{-\infty}^{\infty} yf(y|x)dy \\ &= \frac{\int_{-\infty}^{\infty} yf(x,y)dy}{f_X(x)}, \quad 0 < f_X(x) < \infty. \end{aligned} \tag{5.21}$$

We will define $E[Y|x] = 0$ elsewhere. The quantity $m(x) = E[Y|x]$, considered as a function of x , is known as the **regression function** of Y on X .

In case the random variables X and Y are discrete, the conditional expectation $E[Y|x]$ is defined as

$$\begin{aligned} E[Y|X = x] &= \sum_y yP(Y = y | X = x) \\ &= \sum_y y p_{Y|X}(y|x). \end{aligned} \tag{5.22}$$

Similar definitions can be given in mixed situations. These definitions can be easily generalized to define the conditional expectation of a function $\phi(Y)$:

$$E[\phi(Y)|X = x] = \begin{cases} \int_{-\infty}^{\infty} \phi(y) f_{Y|X}(y|x) dy, & \text{if } Y \text{ is continuous,} \\ \sum_i \phi(y_i) p_{Y|X}(y_i|x), & \text{if } Y \text{ is discrete.} \end{cases} \quad (5.23)$$

As a special case of definition (5.23), we have the conditional k th moment of Y , $E[Y^k|X = x]$, and the conditional moment generating function of Y , $M_{Y|X}(\theta|x) = E[e^{\theta Y}|X = x]$. From the conditional moment generating function we also obtain the definition of the conditional Laplace–Stieltjes transform, $L_{Y|X}(s|x) = E[e^{-s Y}|X = x]$, and the conditional PGF, $G_{Y|X}(z|x) = E[z^Y|X = x]$.

We may take the expectation of the regression function $m(X)$ to obtain the unconditional expectation of Y

$$E[m(X)] = E[E[Y|X]] = E[Y];$$

that is to say

$$E[Y] = \begin{cases} \sum_x E[Y|X = x] p_X(x), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} E[Y|X = x] f_X(x) dx, & \text{if } X \text{ is continuous.} \end{cases} \quad (5.24)$$

This last formula, known as the **theorem of total expectation**, is found to be quite useful in practice. A similar result called the **theorem of total moments** is given by

$$E[Y^k] = \begin{cases} \sum_x E[Y^k|X = x] p_X(x), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} E[Y^k|X = x] f_X(x) dx, & \text{if } X \text{ is continuous.} \end{cases} \quad (5.25)$$

Similarly, we have theorems of total transforms. For example, the theorem of total Laplace–Stieltjes transform is (assuming that Y is a nonnegative continuous random variable)

$$L_Y(s) = \begin{cases} \sum_x L_{Y|X}(s|x) p_X(x), & \text{if } X \text{ is discrete,} \\ \int_{-\infty}^{\infty} L_{Y|X}(s|x) f_X(x) dx, & \text{if } X \text{ is continuous.} \end{cases} \quad (5.26)$$

Example 5.11

Consider the Example 5.6 of r job classes in a file server. Since

$$f_{Y|X}(y|i) = \lambda_i e^{-\lambda_i y},$$

then

$$E[Y|X = i] = \frac{1}{\lambda_i}$$

and

$$E[Y^2|X = i] = \frac{2}{\lambda_i^2}.$$

Then, by the theorem of total expectation, we obtain

$$E[Y] = \sum_{i=1}^r \frac{\alpha_i}{\lambda_i}$$

and

$$E[Y^2] = \sum_{i=1}^r \frac{2\alpha_i}{\lambda_i^2}.$$

Then

$$\text{Var}[Y] = \sum_{i=1}^r \frac{2\alpha_i}{\lambda_i^2} - \left(\sum_{i=1}^r \frac{\alpha_i}{\lambda_i} \right)^2.$$

‡

Example 5.12

Refer to Example 5.10 of a series system with a random number of components, where

$$f_{X|Y}(t|n) = n[1 - G(t)]^{n-1}g(t), \quad t > 0.$$

Let

$$G(t) = 1 - e^{-\lambda t}, \quad x > 0, t \geq 0.$$

Then

$$\begin{aligned} f_{X|Y}(t|n) &= ne^{-\lambda(n-1)t} \lambda e^{-\lambda t} \\ &= n\lambda e^{-n\lambda t}, \end{aligned}$$

which is the exponential pdf with parameter $n\lambda$. It follows that

$$E[X|Y = n] = \frac{1}{n\lambda}$$

and since Y has a modified Poisson pmf, we have

$$E[X] = \sum_{n=1}^{\infty} \frac{1}{n\lambda} \frac{e^{-\alpha}}{1 - e^{-\alpha}} \frac{\alpha^n}{n!}.$$

‡

Example 5.13 [HESS 2000]

Let Y denote the time to failure of a system. System MTTF varies with the temperature of its environment. The conditional MTTF given temperature T is assumed

to have the form: $\text{MTTF}(t) = E[Y|T = t] = e^{a+bt+ct^2}$. The parameters in this formula have been evaluated from measured data as follows: $a = 0.973$, $b = 0.00442$, $c = -0.00036$. It is reasonable to assume that T has a normal distribution with mean $\bar{T} = 40^\circ\text{C}$ and $\sigma = 20^\circ\text{C}$. So the density of T is

$$f_T(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\frac{t-\bar{T}}{\sqrt{2\sigma}})^2}$$

Then the unconditional MTTF can be evaluated by the theorem of total expectation [equation (5.24)]

$$\begin{aligned}\text{MTTF} &= E[Y] = \int_{-\infty}^{\infty} \text{MTTF}(t)f_T(t)dt \\ &= \int_{-\infty}^{\infty} e^{a+bt+ct^2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\frac{t-\bar{T}}{\sqrt{2\sigma}})^2} dt \\ &= \frac{1}{\sqrt{1-2c\sigma^2}} \exp \left[\left(a + \frac{(b\sigma)^2}{2} \right) + b\bar{T} + c\bar{T}^2 \right] \\ &= 1.58\end{aligned}$$

Hence the ratio of the unconditional MTTF and the conditional MTTF at the mean temperature is

$$\frac{\text{MTTF}}{\text{MTTF}(\bar{T})} = \frac{1}{\sqrt{1-2c\sigma^2}} e^{(b\sigma)^2/2} = 0.89$$

Hence, if we use the conditional MTTF at the ambient temperature of 40°C in our calculation, we will have made an error on the optimistic side by about 11%. ‡

Example 5.14 (Analysis of Uniform Hashing) [KNUT 1998]

A popular method of storing tables for fast searching is known as *hashing*. The table has M entries indexed from 0 to $M - 1$. Given a search key k , an application of the hash function h produces an index, $h(k)$, into the table, where we generally expect to find the required entry. Since there are distinct keys $k_i \neq k_j$ that hash to the same value $h(k_i) = h(k_j)$, a situation known as “collision”, we have to derive some method for producing secondary indices for search.

Assume that k entries out of M in the table are currently occupied. As a consequence of the assumption that h distributes values uniformly over the table, all $\binom{M}{k}$ possible configurations are equally likely. Let the random variable X denote the number of probes necessary to insert the next item in the table, and let Y denote the number of occupied entries in the table. For a given number of occupied entries $Y = k$, if the number of probes is equal to r , then $(r - 1)$ given cells are known to be occupied and the last inspected cell is known to be unoccupied. Out of the remaining $M - r$ cells, $(k - r + 1)$ can be occupied in $\binom{M-r}{k-r+1}$ ways.

Therefore

$$\begin{aligned} P(X = r | Y = k) &= p_{X|Y}(r|k) \\ &= \frac{\binom{M-r}{k-r+1}}{\binom{M}{k}}, \quad 1 \leq r \leq M. \end{aligned} \quad (5.27)$$

This implies that

$$\begin{aligned} E[X|Y = k] &= \sum_{r=1}^M r p_{X|Y}(r|k) \\ &= \sum_{r=1}^M (M+1)p_{X|Y}(r|k) - \sum_{r=1}^M (M+1-r)p_{X|Y}(r|k). \end{aligned}$$

Now, since $p_{X|Y}$ is a pmf, the first sum on the right-hand side equals $M+1$. We substitute expression (5.27) in the second sum to obtain

$$\begin{aligned} E[X|Y = k] &= (M+1) - \sum_{r=1}^M (M+1-r) \frac{\binom{M-r}{k-r+1}}{\binom{M}{k}} \\ &= (M+1) - \sum_{r=1}^M \frac{(M+1-r)(M-r)!}{(k-r+1)!(M-k-1)!} \binom{M}{k} \\ &= (M+1) - \sum_{r=1}^M \frac{(M-r+1)!(M-k)}{(k-r+1)!(M-k)!} \binom{M}{k} \\ &= (M+1) - \sum_{r=1}^M \frac{(M-k) \binom{M-r+1}{M-k}}{\binom{M}{k}}. \end{aligned}$$

Now the sum becomes

$$\sum_{r=1}^M \binom{M-r+1}{M-k} = \sum_{i=1}^M \binom{i}{M-k} = \sum_{i=0}^M \binom{i}{M-k} = \binom{M+1}{M-k+1},$$

using a formula from Knuth [KNUT 1997]. After substitution and simplification, we have

$$E[X|Y = k] = \frac{M+1}{M-k+1}, \quad 0 \leq k \leq M-1.$$

Now, assuming that Y is uniformly distributed over $0 \leq k < N \leq M$, we get

$$p_Y(k) = \frac{1}{N},$$

$$\begin{aligned}
E[X] &= \sum_{k=0}^{N-1} \frac{1}{N} E[X|Y=k] \\
&= \frac{M+1}{N} \left(\frac{1}{M+1} + \frac{1}{M} + \cdots + \frac{1}{M-N+2} \right) \\
&= \frac{M+1}{N} (H_{M+1} - H_{M-N+1}) \\
&\simeq \frac{1}{\alpha} \ln \frac{1}{1-\alpha},
\end{aligned}$$

where $\alpha = N/(M+1)$, the table occupancy factor. This is the expected number of probes necessary to locate an entry in the table, provided the search is successful. Note that if the table occupancy factor is low (below 80 %), the average number of probes is nearly equal to 1. In other words, where applicable, this is an efficient method of search.

‡

Example 5.15

Define conditional mean exceedance (CME_x) of a random variable X as

$$\text{CME}_x = E[X - x | X \geq x]. \quad (5.28)$$

This is also called the *mean residual life*. For Pareto distribution with $\alpha > 1$, we have

$$\begin{aligned}
\text{CME}_x &= \frac{\int_x^\infty (t-x) \alpha k^\alpha t^{-\alpha-1} dt}{(\frac{k}{x})^\alpha} \\
&= \frac{x}{\alpha-1}.
\end{aligned}$$

The CME_x of a random variable following Pareto distribution with $\alpha > 1$ is an increasing function of x . This kind of distribution is called “**heavy-tailed**”. Assume the random variable X represents a waiting time. The heavy-tailed distribution means the longer a customer has waited, the longer is this customer’s expected future waiting time. Similarly, a “**light-tailed distribution**”, whose CME_x is a decreasing function of x , means that a customer who has waited for a long time will have a shorter expected future waiting time. The (memoryless) exponential distribution whose CME_x is a constant is called a “**medium-tailed**” distribution.

‡

Problems

1. Consider again problem 1 in Section 5.2. Compute the MTTF of the multiprocessor system first for the constant failure rate case and then for the Weibull failure time distribution case.
2. ★ The notion of a recovery block was introduced by Randell [RAND 1975] to facilitate software fault tolerance in presence of software design errors. This construct provides a “normal” algorithm to perform the required function together

with an acceptance test of its results. If the test results are unsatisfactory then an alternative algorithm is executed. Assume that X is the execution time of the normal algorithm and Y is the execution time of the alternative algorithm. Assume p is the probability that the results of the normal execution satisfy the acceptance test. Determine the distribution function of the total execution time T of the recovery block, assuming that X and Y are uniformly distributed over (a, b) . Repeat, assuming that X and Y are exponentially distributed with parameters λ_1 and λ_2 , respectively. In each case determine $E[T]$, $\text{Var}[T]$, and in the latter case $L_T(s)$.

3. Consider the flowchart model of fault recovery in a computer system (such as Bell System's Electronic Switching system) as shown in Figure 5.P.1. Assuming that the random variables D , L , R , M_D , and M_L are exponentially distributed with parameters δ , λ , ρ , μ_1 and μ_2 , determine the distribution function of the random variable X , denoting the total recovery time. Also compute $E[X]$ and $\text{Var}[X]$.

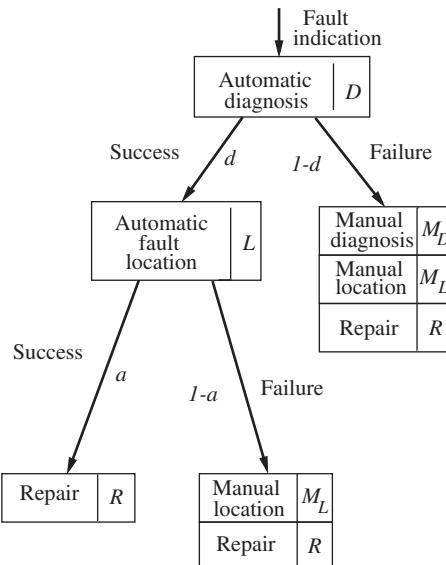


Figure 5.P.1. Flowchart of automatic fault recovery

4. *Linear searching problem.* We are given an unordered list with n distinct keys. We are searching linearly for a specific key that has a probability p of being present in the list (and probability q of being absent). Given that the key is in the list, the probability of its being in position i is $1/n$, $i = 1, 2, \dots, n$. Compute the expected number of comparisons for
 - A successful search.
 - An unsuccessful search.
 - A search (unconditionally).
5. Let V_1 be the random variable denoting the length (in bytes) of a source program [MEND 1979]. Let p be the probability of successful compilation of the program.

Let V_2 be the length of the compiled code (load module). Clearly, V_2 and V_1 will not be independent. Assume $V_2 = BV_1$ where B is a random variable, and B and V_1 are independent. After the compilation, the load module will be entered into a library. Let X be the length of a request for space allocation to the library due to the abovementioned source program. Determine $E[X]$ and $\text{Var}[X]$ in terms of $E[B]$, $E[V_1]$, $\text{Var}[B]$, and $\text{Var}[V_1]$.

5.4 IMPERFECT FAULT COVERAGE AND RELIABILITY

Reliability models of systems with dynamic redundancy (e.g., standby redundancy, hybrid k -out-of- n) developed earlier are not very realistic. It has been demonstrated that the reliability of such systems depends strongly on the effectiveness of recovery mechanisms. In particular, it may be impossible to switch in an existing spare module and thus recover from a failure. Faults such as these are said to be *not covered*, and the probability that a given fault belongs to this class is denoted by $1 - c$, where c denotes the probability of occurrence of *covered* faults, and is known as the **coverage factor** (or **coverage parameter**) [BOUR 1969].

In a fault tolerant system (see Figure 5.5), there are three common phases of recovery subsequent the occurrence of a fault: fault detection, fault location, and recovery for continued service. Each phase has a certain duration and success probability associated with it. The overall probability of successful system recovery is the product of the individual success probabilities of each phase. Thus, if the probabilities of successful detection, successful location and successful recovery are c_d , c_l , and c_r , respectively, then the overall coverage is given by

$$\begin{aligned} c &= P(\text{"system recovers"} \mid \text{"fault occurs"}) \\ &= P(\text{"fault is detected AND fault is located} \\ &\quad \text{AND fault is corrected"} \mid \text{"fault occurs"}) \end{aligned}$$

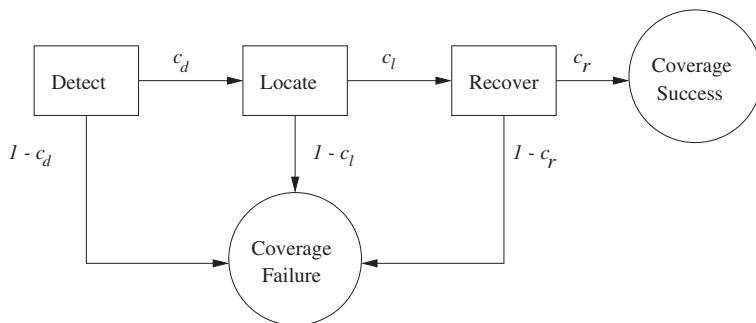


Figure 5.5. Three phases of fault handling

$$\begin{aligned}
&= (\text{detection coverage}) \times (\text{location coverage}) \times (\text{recovery coverage}) \\
&= c_d \cdot c_l \cdot c_r.
\end{aligned}$$

Sometimes, however, a second fault may occur while the system is processing the previous fault. This would normally result in a system failure. Such faults are termed **near-coincident** or **nearly concurrent** faults.

Let Y be the random time to handle (or process) a fault. Let X be the random variable that represents the time to occurrence of an interfering near-coincident fault. Assume that X is exponentially distributed with parameter γ . Thus, $F_X(t) = 1 - e^{-\gamma t}$. Since coverage is the probability that the permanent recovery is completed before the fault occurs, it follows that

$$\begin{aligned}
c_\gamma &= P(Y < X) \\
&= \int_0^\infty P(X > Y | Y = t) f_Y(t) dt
\end{aligned} \tag{5.29}$$

$$\begin{aligned}
&= \int_0^\infty P(X > t) f_Y(t) dt \\
&= \int_0^\infty e^{-\gamma t} f_Y(t) dt = L_Y(\gamma).
\end{aligned} \tag{5.30}$$

Thus the coverage from near-coincident faults is computed as the LST of the recovery time evaluated at the rate of occurrence of near-coincident faults.

Applying this idea to the three phases of fault handling, we have the overall coverage

$$c = c_d L_D(\gamma) c_l \quad L_L(\gamma) c_r L_R(\gamma)$$

Thus, successful recovery requires that each of the steps of fault handling is successful in the absence of a near-coincident fault and that a near-coincident fault does not occur during any of the phases of fault handling [DUGA 1989].

Example 5.16

Let X denote the lifetime of a system with two units, one active and the other a cold standby spare. The failure rate of an active unit is λ , and a cold spare does not fail. Let Y be the indicator random variable of the fault class:

$$\begin{aligned}
Y &= 0 \text{ if the fault is not covered,} \\
Y &= 1 \text{ if the fault is covered.}
\end{aligned}$$

Then

$$p_Y(0) = 1 - c \quad \text{and} \quad p_Y(1) = c.$$

To compute the MTTF of this system, we first obtain the conditional expectation of X given Y by noting that if a not-covered fault occurs, the mean life of the system

equals the mean life of the initially active unit:

$$E[X | Y = 0] = \frac{1}{\lambda}.$$

On the other hand, if a covered fault occurs, then the mean life of the system is the sum of the mean lives of the two units:

$$E[X | Y = 1] = \frac{2}{\lambda}.$$

Now, using the theorem of total expectation, we obtain the system MTTF as

$$E[X] = \frac{1-c}{\lambda} + \frac{2c}{\lambda} = \frac{1+c}{\lambda}. \quad (5.31)$$

Thus, when $c = 0$, the standby module does not contribute anything to system reliability, and when $c = 1$, the full potential of this module is realized. For $c < 0.5$, MTTF of a **parallel redundant configuration** with two units (static redundancy) is *higher* than that of a two-unit **standby redundant system** (dynamic redundancy).

Given that the fault was covered ($Y = 1$), the system lifetime, X , is the sum of two independent exponentially distributed random variables, each with parameter λ . Thus the conditional pdf of X given $Y = 1$ is the two-stage Erlang density:

$$f_{X|Y}(t|1) = \lambda^2 t e^{-\lambda t}.$$

On the other hand, given that a not-covered fault occurred, the system lifetime X is simply the lifetime of the initially active component. Hence

$$f_{X|Y}(t|0) = \lambda e^{-\lambda t}.$$

Then the joint density is computed by $f(t, y) = f_{X|Y}(t|y)p_Y(y)$ as

$$f(t, y) = \begin{cases} \lambda(1-c)e^{-\lambda t}, & t > 0, \quad y = 0, \\ \lambda^2 cte^{-\lambda t}, & t > 0, \quad y = 1, \end{cases}$$

and the marginal density of X is computed by summing over the joint density:

$$f_X(t) = f(t, 0) + f(t, 1) = \lambda^2 cte^{-\lambda t} + \lambda(1-c)e^{-\lambda t}.$$

Therefore, the system reliability is given by

$$\begin{aligned} R_X(t) &= (1-c)e^{-\lambda t} + ce^{-\lambda t}(1+\lambda t) \\ &= e^{-\lambda t} + c\lambda t e^{-\lambda t} \\ &= (1 + c\lambda t)e^{-\lambda t}. \end{aligned} \quad (5.32)$$

Figure 5.6 shows $R_X(t)$ as a function of t for various values of the coverage parameter.

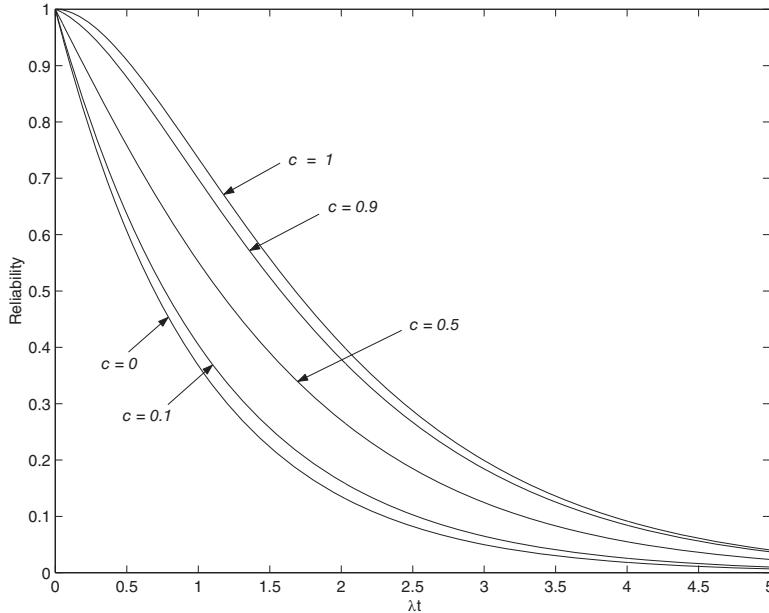


Figure 5.6. Reliability of two-component standby system with imperfect coverage

The conditional Laplace–Stieltjes transform of the lifetime X is given by

$$L_{X|Y}(s | 0) = \frac{\lambda}{s + \lambda} \quad \text{and} \quad L_{X|Y}(s | 1) = \left(\frac{\lambda}{s + \lambda} \right)^2.$$

Then the unconditional transform is computed using the theorem of total transform:

$$\begin{aligned} L_X(s) &= c \frac{\lambda^2}{(s + \lambda)^2} + (1 - c) \frac{\lambda}{s + \lambda} \\ &= \frac{\lambda}{s + \lambda} \left[c \frac{\lambda}{s + \lambda} + (1 - c) \right]. \end{aligned} \quad (5.33)$$

Let us rewrite this as follows:

$$L_X(s) = L_{Y_1}(s)L_{Y_2}(s), \quad (5.34)$$

where

$$L_{Y_1}(s) = \frac{\lambda}{s + \lambda} \quad (5.35)$$

and

$$L_{Y_2}(s) = \frac{c\lambda}{s + \lambda} + (1 - c). \quad (5.36)$$

Using the convolution theorem, we conclude that we can regard system lifetime X as the sum of two independent random variables Y_1 and Y_2 . From the

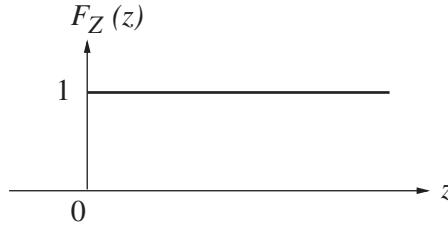


Figure 5.7. The unit-step function: CDF of a constant random variable, $Z = 0$

Laplace–Stieltjes transform of Y_1 , we see that $Y_1 \sim \text{EXP}(\lambda)$. Noting that

$$\lim_{\mu \rightarrow \infty} \frac{\mu}{s + \mu} = 1 , \quad (5.37)$$

we can rewrite

$$L_{Y_2}(s) = \lim_{\mu \rightarrow \infty} \left[c \frac{\lambda}{s + \lambda} + (1 - c) \frac{\mu}{s + \mu} \right] . \quad (5.38)$$

Thus, Y_2 may be regarded as the limit of a two-stage hyperexponentially distributed random variable with parameters λ and μ . Further thought reveals that in the limit $\mu \rightarrow \infty$, the distribution function of $Z \sim \text{EXP}(\mu)$ becomes

$$\begin{aligned} \lim_{\mu \rightarrow \infty} F_Z(z) &= \lim_{\mu \rightarrow \infty} [1 - e^{-\mu z}] \\ &= \begin{cases} 1, & z > 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (5.39)$$

This function is the unit-step function shown in Figure 5.7, and the corresponding random variable is the constant random variable $Z = 0$.

On the basis of the discussion above, we can visualize the system lifetime X as composed of exponential stages as shown in Figure 5.8.

#

Example 5.17

In the last example we assumed that a unit in a standby status does not fail. Now assume that such a unit can fail with a constant failure rate μ (presumably, $0 \leq \mu \leq \lambda$), thus making it a warm spare. If $\mu = \lambda$, we have the parallel redundancy—hence hot spare. Let c_1 be the probability of successful recovery on the failure of an active unit, and let c_2 be the probability of successful recovery following the failure of a spare unit. Note that Bouricius and others [BOUR 1969] assumed that $c_1 = c_2 = c$. Keeping the same notations and assumptions as before, we can compute the Laplace–Stieltjes transform of the system lifetime X as follows.

Let X_1 and X_2 denote the time to failure of the powered and unpowered units, respectively. Also let W be the residual lifetime of the unit in operation after a covered fault has occurred. We observe that $X_1 \sim \text{EXP}(\lambda)$, $X_2 \sim \text{EXP}(\mu)$, and, because of the memoryless property of the exponential distribution, $W \sim \text{EXP}(\lambda)$.

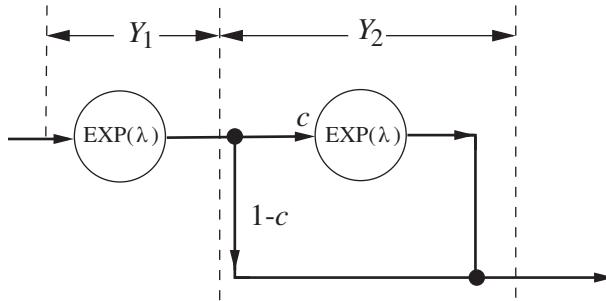


Figure 5.8. The stage-type distribution of system lifetime for the system in Example 5.16

Define the random variable Y so that

$$Y = \begin{cases} 0, & \text{not-covered failure in the active unit,} \\ 1, & \text{covered failure in the active unit,} \\ 2, & \text{not-covered failure in the standby unit,} \\ 3, & \text{covered failure in the standby unit.} \end{cases}$$

First, we compute the pmf of Y by noting that the probability of the active unit failing first is $\lambda/(\lambda + \mu)$ while the probability of the spare unit failing first is $\mu/(\lambda + \mu)$ (refer to Example 5.4). Then (see the tree diagram of Figure 5.9):

$$\begin{aligned} p_Y(1) &= \frac{\lambda c_1}{\lambda + \mu}, & p_Y(0) &= \frac{\lambda(1 - c_1)}{\lambda + \mu}, \\ p_Y(3) &= \frac{\mu c_2}{\lambda + \mu}, & \text{and } p_Y(2) &= \frac{\mu(1 - c_2)}{\lambda + \mu}. \end{aligned}$$

Now, if a not-covered fault has occurred (i.e., $Y = 0$ or $Y = 2$) the lifetime of the system is simply $\min\{X_1, X_2\}$, while a covered fault (i.e., $Y = 1$ or $Y = 3$) implies

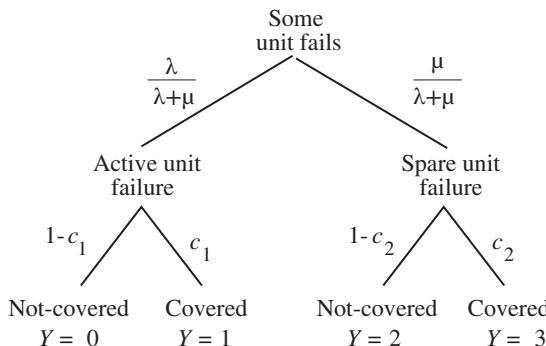


Figure 5.9. Tree diagram for Example 5.17

a system lifetime of

$$\min\{X_1, X_2\} + W.$$

Note that since X_1 and X_2 are exponentially distributed, $\min\{X_1, X_2\}$ is exponentially distributed with parameter $\lambda + \mu$. The conditional Laplace–Stieltjes transform of X for each type of not-covered fault is therefore

$$L_{X|Y}(s|Y=0) = \frac{\lambda + \mu}{s + (\lambda + \mu)} = L_{X|Y}(s|Y=2),$$

and for a covered fault, X is the sum of two independent exponentially distributed random variables and hence

$$L_{X|Y}(s|Y=1) = \frac{\lambda + \mu}{s + (\lambda + \mu)} \cdot \frac{\lambda}{s + \lambda} = L_{X|Y}(s|Y=3).$$

The unconditional Laplace–Stieltjes transform of X is then computed using the theorem of total transform:

$$L_X(s) = \frac{\lambda + \mu}{s + \lambda + \mu} \cdot \frac{\lambda(1 - c_1) + \mu(1 - c_2)}{\lambda + \mu} + \frac{(\lambda + \mu)\lambda}{(s + \lambda + \mu)(s + \lambda)} \cdot \frac{\lambda c_1 + \mu c_2}{\lambda + \mu} \quad (5.40)$$

Thus

$$\begin{aligned} L_X(s) &= \frac{\lambda + \mu}{s + \lambda + \mu} \left[\frac{\lambda c_1 + \mu c_2}{\lambda + \mu} \cdot \frac{\lambda}{s + \lambda} + \frac{\lambda(1 - c_1) + \mu(1 - c_2)}{\lambda + \mu} \right] \\ &= L_{Y_1}(s)L_{Y_2}(s), \end{aligned} \quad (5.41)$$

where

$$L_{Y_1}(s) = \frac{\lambda + \mu}{s + \lambda + \mu}$$

and

$$\begin{aligned} L_{Y_2}(s) &= \frac{\lambda c_1 + \mu c_2}{\lambda + \mu} \cdot \frac{\lambda}{s + \lambda} + \frac{\lambda(1 - c_1) + \mu(1 - c_2)}{\lambda + \mu} \\ &= c \frac{\lambda}{s + \lambda} + (1 - c), \end{aligned}$$

where the “equivalent” coverage c is given by

$$c = \frac{\lambda c_1 + \mu c_2}{\lambda + \mu}. \quad (5.42)$$

We conclude that the system lifetime $X = Y_1 + Y_2$, and it can be regarded as stage-type random variable as shown in Figure 5.9. Now, since

$$E[Y_1] = \frac{1}{\lambda + \mu}$$

and

$$E[Y_2] = -\frac{dLY_2}{ds}|_{s=0} = \frac{c}{\lambda},$$

we conclude that the MTTF of the system is given by

$$E[X] = \frac{1}{\lambda + \mu} + \frac{c}{\lambda}. \quad (5.43)$$

Comparing the form (5.40) of the Laplace–Stieltjes transform of X with the LST of a mixture distribution (5.18), we conclude that X is a mixture of $100(1 - c)\%$ of an exponential, $\text{EXP}(\lambda + \mu)$ (see Figure 5.10), with $100c\%$ of a hypoexponential, $\text{Hypo}(\lambda + \mu, \lambda)$. Therefore

$$f_X(t) = (1 - c)(\lambda + \mu)e^{-(\lambda + \mu)t} + c \frac{\lambda(\lambda + \mu)}{\mu}(e^{-\lambda t} - e^{-(\lambda + \mu)t}), \quad t > 0, \quad (5.44)$$

and the system reliability is given by

$$\begin{aligned} R_X(t) &= (1 - c)e^{-(\lambda + \mu)t} + c \frac{\lambda(\lambda + \mu)}{\mu} \left[\frac{1}{\lambda} e^{-\lambda t} - \frac{1}{\lambda + \mu} e^{-(\lambda + \mu)t} \right] \\ &= (1 - c)e^{-(\lambda + \mu)t} + \frac{c}{\mu} [(\lambda + \mu)e^{-\lambda t} - \lambda e^{-(\lambda + \mu)t}], \quad t \geq 0, \end{aligned} \quad (5.45)$$

where c is given by (5.42). ‡

Cox [COX 1955] has analyzed the more general stage-type distribution shown in Figure 5.11. The Laplace–Stieltjes transform of such a stage-type random variable X is given by

$$L_X(s) = \gamma_1 + \sum_{i=1}^r \beta_1 \beta_2 \cdots \beta_i \gamma_{i+1} \prod_{j=1}^i \frac{\mu_j}{s + \mu_j}, \quad (5.46)$$

where $\gamma_i + \beta_i = 1$ for $1 \leq i \leq r$ and $\gamma_{r+1} = 1$.

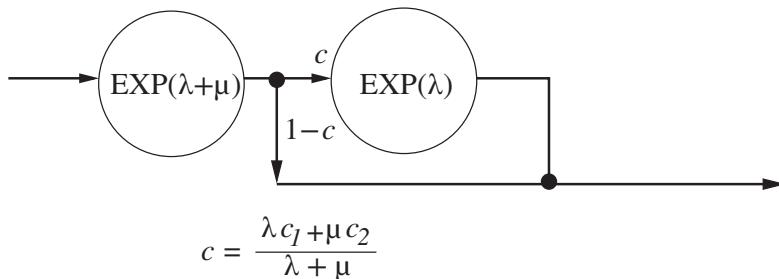


Figure 5.10. Lifetime distribution of a two-component warm standby system with imperfect coverage

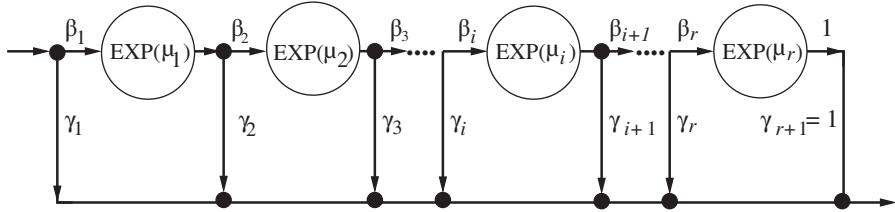


Figure 5.11. Coxian stage-type distribution

Example 5.18

Consider a warm standby redundant system with n units where one unit is initially active and $(n - 1)$ units are in a standby status. Assume that the failure rate of an active unit is λ while that of a standby unit is μ . For simplicity, we assume that the coverage factor is the same for active and spare failures and is denoted by c . By analogy with Example 5.17, we can say that the lifetime distribution of this system will be stage-type, as shown in Figure 5.12. Using the notation of Figure 5.11, the distribution of Figure 5.12 corresponds to the following parameters:

$$\begin{aligned}\beta_1 &= 1, \quad \gamma_1 = 0, \quad \gamma_{n+1} = 1, \\ \beta_i &= c, \quad \gamma_i = 1 - c, \quad 2 \leq i \leq n, \\ \mu_i &= \lambda + (n - i)\mu, \quad 1 \leq i \leq n.\end{aligned}$$

Using equation (5.46), we obtain the Laplace–Stieltjes transform of the system lifetime as

$$L_X(s) = \sum_{i=1}^{n-1} c^{i-1}(1-c) \prod_{j=1}^i \frac{\lambda + (n-j)\mu}{s + \lambda + (n-j)\mu} \quad (5.47)$$

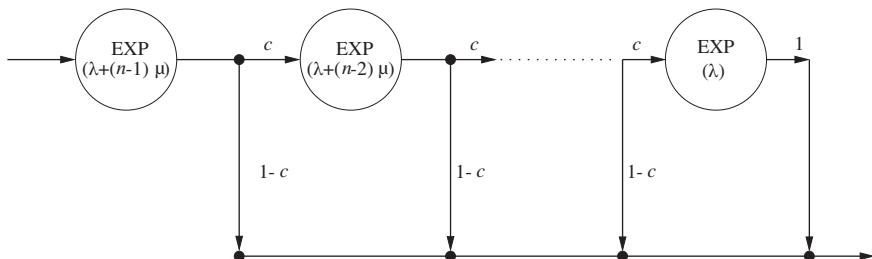


Figure 5.12. Lifetime distribution of an n -component standby redundant system with imperfect coverage

$$+c^{n-1} \prod_{j=1}^n \frac{\lambda + (n-j)\mu}{s + \lambda + (n-j)\mu}.$$

The system MTTF can be easily computed from expression (5.47) as

$$E[X] = \sum_{i=1}^{n-1} c^{i-1} (1-c) \sum_{j=1}^i \frac{1}{\lambda + (n-j)\mu} + c^{n-1} \sum_{j=1}^n \frac{1}{\lambda + (n-j)\mu}. \quad (5.48)$$

‡

Example 5.19

Now we consider a hybrid k -out-of- n system with imperfect coverage. Assume that the failure rate of an active unit is λ and the failure rate of a standby spare is μ . We continue with the assumption that the coverage factor is the same for an active-unit failure as that for a standby-unit failure. Initially, there are n active units and m spares. The lifetime of such a system has the stage-type distribution shown in Figure 5.13. Therefore, in the notation of the Coxian distribution of Figure 5.11, we have

$$\begin{aligned} \beta_1 &= 1, \quad \gamma_1 = 0, \\ \beta_i &= c, \quad \gamma_i = 1 - c, \quad 2 \leq i \leq m+1, \\ \beta_i &= 1, \quad \gamma_i = 0, \quad m+2 \leq i \leq (n-k)+m+1, \\ \gamma_{n-k+m+2} &= 1, \\ \mu_j &= n\lambda + (m-j+1)\mu, \quad 1 \leq j \leq m, \\ \mu_j &= n\lambda + (m-j+1)\mu, \quad m+1 \leq j \leq n-k+m+1. \end{aligned}$$

Then, using formula (5.46), we have

$$\begin{aligned} L_X(s) &= \sum_{i=1}^m c^{i-1} (1-c) \prod_{j=1}^i \frac{n\lambda + (m-j+1)\mu}{s + n\lambda + (m-j+1)\mu} \\ &\quad + c^m \prod_{j=1}^m \frac{n\lambda + j\mu}{s + (n\lambda + j\mu)} \prod_{j=1}^{n-k+1} \frac{(n-j+1)\lambda}{s + (n-j+1)\lambda}. \end{aligned} \quad (5.49)$$

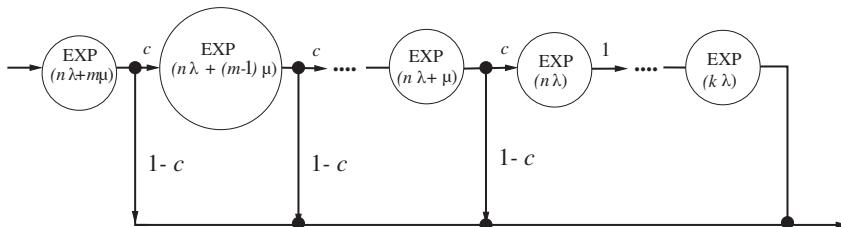


Figure 5.13. Lifetime distribution of hybrid k -out-of- n system with imperfect coverage

After a partial fraction expansion and inversion of the transform above, we can get the required expression for reliability of a hybrid k -out-of- n system with imperfect coverage. We omit the details here; the interested reader is referred to Ng's thesis [NG 1976]. We can easily compute the mean life from the expression of the Laplace-Stieltjes transform:

$$\begin{aligned}
 E[X] &= \sum_{i=1}^m c^{i-1}(1-c) \sum_{j=1}^i \frac{1}{n\lambda + (m-j+1)\mu} \\
 &\quad + c^m \left[\sum_{j=1}^m \frac{1}{n\lambda + j\mu} + \sum_{j=1}^{n-k+1} \frac{1}{(n-j+1)\lambda} \right] \\
 &= \sum_{i=1}^m c^{i-1}(1-c) \sum_{j=m-i+1}^m \frac{1}{n\lambda + j\mu} \\
 &\quad + c^m \left[\sum_{j=1}^m \frac{1}{n\lambda + j\mu} + \sum_{j=k}^n \frac{1}{j\lambda} \right]. \tag{5.50}
 \end{aligned}$$

‡

The method of stages works quite well for the reliability analysis of nonrepairable systems. When we deal with repairable systems, the number of stages will become infinite, and this method becomes cumbersome to use. We will analyze such systems in Chapter 8 using the theory of Markov chains.

Problems

1. We are given a system with three components. Whenever a component is energized, it has an exponential failure law with parameter λ . When a component is deenergized, it has an exponential failure law with parameter $\mu (< \lambda)$. The system will function correctly if two components are in proper working order. Consider two ways of using the spare unit: one way is to keep all three units energized, another way is to keep the third unit as a deenergized spare, and, when one of the operating units fails, switch the spare in. But the switching equipment need not be perfect; suppose that it has reliability 0.9. Find the reliability expressions for the two schemes and find conditions under which one scheme will be better than the other.
2. Starting from formula (5.47), use the moment generating property of the Laplace-Stieltjes transform to show (5.48). Also obtain a formula for the variance of system lifetime for the system of Example 5.18.

5.5 RANDOM SUMS

We have considered sums of N mutually independent random variables when N is a fixed constant. Here we are interested in the case where N itself is

random variable that is independent of X_k . Given a list X_1, X_2, \dots , of mutually independent identically distributed random variables with distribution function $F(x)$, mean $E[X]$, and variance $\text{Var}[X]$, consider the random sum:

$$T = X_1 + X_2 + \cdots + X_N. \quad (5.51)$$

Here the pmf of the discrete random variable $p_N(n)$ is assumed to be given.

For a fixed value $N = n$, the conditional expectation of T is easily obtained:

$$\begin{aligned} E[T|N = n] &= \sum_{i=1}^n E[X_i] \\ &= nE[X]. \end{aligned} \quad (5.52)$$

Then, using the theorem of total expectation, we get

$$\begin{aligned} E[T] &= \sum_n nE[X]p_N(n) \\ &= E[X] \sum_n np_N(n) \\ &= E[X]E[N]. \end{aligned} \quad (5.53)$$

Equation (5.53) is called the *Wald's equation* (see Cox [COX 1962]). In order to obtain the $\text{Var}[T]$, we first compute $E[T^2]$. Note that

$$E[T^2|N = n] = \text{Var}[T|N = n] + (E[T|N = n])^2 \quad (5.54)$$

but

$$\begin{aligned} \text{Var}[T|N = n] &= \sum_{i=1}^n \text{Var}[X_i] \\ &= n\text{Var}[X] \quad (\text{by independence of the } X_i\text{'s}). \end{aligned} \quad (5.55)$$

Substituting (5.52) and (5.55) in (5.54) yields

$$E[T^2|N = n] = n\text{Var}[X] + n^2(E[X])^2.$$

Now, using the theorem of total moments, we have

$$\begin{aligned} E[T^2] &= \sum_n [n\text{Var}[X] + n^2(E[X])^2]p_N(n) \\ &= \text{Var}[X]E[N] + E[N^2](E[X])^2. \end{aligned}$$

Finally, we obtain

$$\begin{aligned}\text{Var}[T] &= E[T^2] - (E[T])^2 \\ &= \text{Var}[X]E[N] + E[N^2](E[X])^2 - (E[X])^2(E[N])^2 \\ &= \text{Var}[X]E[N] + (E[X])^2\text{Var}[N].\end{aligned}\quad (5.56)$$

Assuming that the $\{X_i\}$ are continuous random variables with Laplace–Stieltjes transform $L_X(s)$, we obtain the conditional LST of T as

$$L_{T|N}(s|n) = [L_X(s)]^n.$$

Then, using the theorem of total LST, we have

$$\begin{aligned}L_T(s) &= \sum_n L_{T|N}(s|n)p_N(n) \\ &= \sum_n [L_X(s)]^n p_N(n) \\ &= G_N(L_X(s)).\end{aligned}\quad (5.57)$$

As a special case, assume that N has a geometric distribution with parameter p , so that

$$p_N(n) = (1-p)^{n-1}p$$

and

$$L_T(s) = \sum_{n=1}^{\infty} [L_X(s)]^n (1-p)^{n-1}p = \frac{pL_X(s)}{1-(1-p)L_X(s)}. \quad (5.58)$$

Next assume that the $\{X_i\}$ terms are discrete with the common generating function $G_X(z)$. Then, the conditional PGF of T is

$$G_{T|N}(z|n) = [G_X(z)]^n,$$

and using the theorem of total generating functions, we have the unconditional PGF of T :

$$G_T(z) = \sum_n [G_X(z)]^n p_N(n) = G_N[G_X(z)]. \quad (5.59)$$

Now, if N is geometrically distributed with parameter p , then this formula reduces to

$$G_T(z) = \frac{p G_X(z)}{1-(1-p)G_X(z)}. \quad (5.60)$$

Example 5.20

Consider a memory module with a time to failure X . Since the reliability $R(t)$ of the memory was found to be inadequate, addition of hardware to incorporate error-correcting capability was desired. Assume that the probability of an undetected and/or uncorrected error (not-covered error, for short) is p (and hence the conditional probability that a given error is covered by the coding scheme is $1 - p$). Now let N be the number of errors that are corrected before a not-covered error occurs, then N has a modified geometric distribution with parameter p and $E[N] = (1 - p)/p$. Let T denote the time to occurrence of a not-covered error. Denoting X_i to be the time between the occurrence of the $(i - 1)$ st and i th error, $T = X_0 + X_1 + \dots + X_N$ where X_0 defined to be 0. Then, using formula (5.53), we get

$$\begin{aligned}\text{MTTF}_{\text{with code}} &= \frac{\text{MTTF}_{\text{without code}} \times (1 - p)}{p} \\ &= \frac{\text{MTTF}_{\text{without code}} \times (1 - \text{Probability of a not-covered error})}{\text{Probability of a not-covered error}}.\end{aligned}$$

If we assume that X is exponentially distributed with parameter λ , then

$$L_X(s) = \frac{\lambda}{s + \lambda}$$

and, assuming that X_0, X_1, \dots , are mutually independent, and using formula (5.57), we have

$$\begin{aligned}L_T(s) &= \frac{p}{1 - (1 - p) \frac{\lambda}{s + \lambda}} \\ &= p + \frac{p\lambda(1 - p)}{s + p\lambda}.\end{aligned}\tag{5.61}$$

This implies that T has mixed distribution as in equation (3.2). ‡

Example 5.21

To employ wireless media for data transmission, we have to deal with limited bandwidth and time-varying high values of bit error rate (BER) [NANN 1998]. Consider a polling access method capable of efficient bandwidth utilization and error control scheme by using the *Go Back N* (GBN) ARQ (Automatic Repeat Request) technique for data transmission in wireless communication networks [BERT 1992]. Messages are composed of fixed-length data units called “packets”. The number of packets, N , contained in a message arriving at each mobile terminal is a general distributed random variable with mean $E[N]$ and variance $\text{Var}[N]$. In addition, we call the time interval between two consecutive packet transmission initiation instants a “slot”. In the GBN ARQ case, a slot comprises only the transmission time of a packet. An incorrectly received packet can be retransmitted several times until it is correctly received or the transmission time is greater than m slots. The time T_i , the number of slots necessary to accomplish i th error-free packet transmission, is also a random

variable with pmf as follows:

$$P(T_i = n) = \begin{cases} p(1-p)^{\frac{n-1}{k}}, & n = 1, \dots, (m-1)k+1 \\ 0, & \text{otherwise.} \end{cases}$$

where k is the number of slots the terminal has to wait for the ACK (acknowledgment) of the transmitted packet, including its transmission time. Then

$$\begin{aligned} E[T_i] &= \sum_n n \cdot P(T_i = n) = \sum_{n=1}^{(m-1)k+1} n \cdot p(1-p)^{\frac{n-1}{k}} \\ E[T_i^2] &= \sum_{n=1}^{(m-1)k+1} n^2 \cdot p(1-p)^{\frac{n-1}{k}} \\ \text{Var}[T_i] &= E[T_i^2] - E[T_i]^2 \end{aligned}$$

So, the time elapsed between the instants at which a terminal finishes its transmission and the channel become available to the next terminal is the random $T = T_1 + T_2 + \dots + T_N$. Assuming that T_1, T_2, \dots, T_N are mutually independent, then using formula (5.53), we obtain

$$\begin{aligned} E[T] &= E[T_i]E[N] \\ \text{Var}[T] &= \text{Var}[T_i]E[N] + (E[T_i])^2\text{Var}[N] \end{aligned}$$

The common generating function of T_i is

$$\begin{aligned} G_X(z) &= \sum_{n=0}^{\infty} P(T_i = n)z^n = \sum_{n=1}^{(m-1)k+1} p(1-p)^{\frac{n-1}{k}} z^n \\ &= \frac{p[1 - (1-p)^{[(m-1)k+2]/k} z^{(m-1)k+2}]}{(1-p)^{\frac{1}{k}} - (1-p)^{\frac{2}{k}} z} \end{aligned}$$

and using formula (5.60) above, we get

$$G_T(z) = \frac{p^2[1 - (1-p)^{[(m-1)k+2]/k} z^{(m-1)k+2}]}{(1-p)^{\frac{1}{k}} \left[1 - (1-p)^{\frac{1}{k}} z\right] - p(1-p)[1 - (1-p)^{[(m-1)k+2]/k} z^{(m-1)k+2}]}$$

¶

Example 5.22

Consider the following program segment consisting of a **do while** loop:

```
do S; while ( B );
```

Let X_i denote the execution time for the i th iteration of statement group S . Assume that the sequence of tests of the Boolean expression B defines a sequence of Bernoulli

trials with parameter p . Clearly, the number N of iterations of the loop is a geometric random variable with parameter p so that $E[N] = 1/p$. Letting T denote the total execution time of the loop, and using equation (5.53), the average of execution time T is easily determined to be:

$$E[T] = \frac{E[X]}{p}. \quad (5.62)$$

The variance of the execution time T is determined using (5.56), noting that $\text{Var}[N] = q/p^2$:

$$\text{Var}[T] = \frac{\text{Var}[X]}{p} + (E[X])^2 \frac{q}{p^2}. \quad (5.63)$$

Next assume that the X_i 's are exponentially distributed with parameter λ so that

$$L_X(s) = \frac{\lambda}{s + \lambda},$$

and, using formula (5.61), we get

$$L_T(s) = \frac{p\lambda}{s + p\lambda}. \quad (5.64)$$

Thus the total execution time of the **do while** loop is also exponentially distributed with parameter $p\lambda$. In this case, $E[T] = 1/(p\lambda)$ which agrees with (5.62) and from (5.63)

$$\text{Var}[T] = \frac{1}{\lambda^2 p} + \frac{1}{\lambda^2} \frac{q}{p^2} = \frac{1}{p^2 \lambda^2}$$

as expected.

#

Example 5.23

In measuring the execution time of the **do while** loop above, we use a real time clock with a resolution of $1 \mu\text{s}$. In this case, the execution times will be discrete random variables. Assume that the $\{X_i\}$ terms are geometrically distributed with parameter p_1 . Then

$$G_X(z) = \frac{zp_1}{1 - z(1 - p_1)},$$

and using formula (5.60) above, we get

$$\begin{aligned} G_T(z) &= \frac{pzp_1}{1 - z(1 - p_1) - (1 - p)zp_1} \\ &= \frac{zpp_1}{1 - z(1 - pp_1)}. \end{aligned}$$

Thus T is a geometrically distributed random variable with parameter pp_1 .

#

We are now in a position to obtain the distribution of the execution time of a **do while** loop and in a similar fashion, that of a **while** loop (see problem 1 at the end of this section). Our earlier methods allow us to compute the distribution of the execution time of a compound statement, that of a **for** loop (see the discussion of sums of independent random variables in Chapter 3), and that of an **if** and a **switch** statement (see the discussion of mixture distributions in this chapter). Thus we are now in a position to analyze a **structured program**—a program that uses only combinations of the above-listed control structures. We have summarized these results in Appendix E. We can also deal with the Pascal concurrent control statement **cobegin**. Programs that use unrestricted **gotos** can be analyzed by the methods of Chapter 7.

Example 5.24

Consider the following program:

```
{
  COMP;
  while ( B ) {
    switch (j) {
      case 1 : I/O1; break;
      case 2 : I/O2; break;
      .
      .
      .
      case m : I/Om; break;
    }
    COMP;
  }
}
```

Let the random variable C denote the time to execute the statement group COMP and let I_j ($1 \leq j \leq m$) denote the time to execute the statement group I/O_j . Assume that the condition test on B is a sequence of independent Bernoulli trials with the probability of failure p_0 , and let p'_j be the probability of executing the j th case, given that the **switch** statement is executed. Note that $\sum_{j=1}^m p'_j = 1$. Let the random variable I denote the execution time of the **switch** statement. Note that I is a mixture of random variables I_j ($1 \leq j \leq m$). Given the Laplace–Stieltjes transforms of C and I_j , we proceed to compute the LST of the overall execution time T of the program shown above.

Using the table in Appendix E, we have

$$L_I(s) = \sum_{j=1}^m p'_j L_{I_j}(s),$$

$$L_{\text{whilebody}}(s) = L_I(s) \cdot L_C(s),$$

$$\begin{aligned}
L_{\text{whileloop}}(s) &= \sum_{n=0}^{\infty} (1-p_0)^n p_0 [L_{\text{whilebody}}(s)]^n \\
&= \frac{p_0}{1 - (1-p_0)L_{\text{whilebody}}(s)}, \\
L_T(s) &= \frac{p_0 L_C(s)}{1 - (1-p_0)L_{\text{whilebody}}(s)} \\
&= \frac{p_0 L_C(s)}{1 - (1-p_0)L_C(s)L_I(s)} = \frac{U(s)}{V(s)}. \tag{5.65}
\end{aligned}$$

For a continuous random variable X it is known that $L_X(0) = 1$, $L'_X(0) = -E[X]$, and $L''_X(0) = E[X^2]$; hence

$$\begin{aligned}
V(0) &= 1 - (1-p_0)L_C(0)L_I(0) = p_0, \\
U(0) &= p_0 L_C(0) = p_0, \\
V'(0) &= -(1-p_0)[L_C(0)L'_I(0) + L'_C(0)L_I(0)] = (1-p_0)[E[C] + E[I]] \\
U'(0) &= p_0 L'_C(0) = -p_0 E[C], \\
V''(0) &= -(1-p_0)[L''_C(0)L_I(0) + 2L'_C(0)L'_I(0) + L_C(0)L''_I(0)] \\
&= -(1-p_0)(E[C^2] + 2E[C]E[I] + E[I^2]), \\
U''(0) &= p_0 L''_C(0) = p_0 E[C^2].
\end{aligned}$$

Now we can compute the first two moments of T :

$$\begin{aligned}
E[T] &= -L'_T(0) \\
&= \frac{-V(0)U'(0) + U(0)V'(0)}{V^2(0)} \\
&= \frac{p_0(p_0 E[C]) + (1-p_0)p_0(E[C] + E[I])}{p_0^2} \\
&= \frac{E[C]}{p_0} + \frac{(1-p_0)E[I]}{p_0}. \tag{5.66}
\end{aligned}$$

The terms on the right-hand side of (5.66) are easily interpreted; the first term is the expected value of the total time to execute the statement COMP, since $1/p_0$ is the average number of times COMP is executed, while $E[C]$ is the average time per execution. The second term is the expected total time of all I/O statements, since this term can be written as

$$\sum_{j=1}^m \frac{p_j}{p_0} E[I_j],$$

where p_j is defined to be $(1-p_0)p'_j$. Note that p_j/p_0 is the average number of executions of statement I/O_j , and $E[I_j]$ is the average time per execution.

Next we proceed to compute $E[T^2]$:

$$\begin{aligned}
 E[T^2] &= L''_T(0) \\
 &= \frac{V^2(0)U''(0) - U(0)V(0)V''(0) - 2V(0)V'(0)U'(0) + 2U(0)[V'(0)]^2}{V^3(0)} \\
 &= \frac{E[C^2]}{p_0} + \frac{2(1-p_0)}{p_0}(E[C])^2 + \frac{4(1-p_0)}{p_0^2}E[C]E[I] \\
 &\quad + \frac{1-p_0}{p_0}E[I^2] + \frac{2(1-p_0)}{p_0^2}(E[I])^2.
 \end{aligned} \tag{5.67}$$

Now, from (5.66) and (5.67), we compute the variance:

$$\text{Var}[T] = \frac{\text{Var}[C]}{p_0} + \frac{1-p_0}{p_0}\text{Var}[I] + \frac{1-p_0}{p_0^2}(E[C] + E[I])^2. \tag{5.68}$$

We will use these formulas in Chapter 9 in analyzing a queuing network in which individual programs will behave as discussed in this example.

#

We should caution the reader that several unrealistic assumptions have been made here. The assumption of independence, for example, is questionable. More importantly, we have associated a fixed probability with each conditional branch, independent of the current state of the program. For more involved analyses that attempt to remove such assumptions, see Hofri's treatise [HOFR 1987]. Our treatment of program analysis in this section was control-structure-based. Alternatively, we could perform a data-structure-oriented analysis as in the analysis of program MAX (Chapter 2). Further examples of this technique will be presented in Chapter 7. In practice, these two techniques need to be used in conjunction with each other.

Problems

- Carry out an analysis of the execution time of the **while** loop:

```
while ( B ) S;
```

following the analysis of the **do while** loop given in Example 5.22.

- A CPU burst of a task is exponentially distributed with mean $1/\mu$. At the end of a burst, the task requires another burst with probability p and finishes execution with probability $1-p$. Thus the number of CPU bursts required for task completion is a random variable with the image $\{1, 2, \dots\}$. Find the distribution function of the total service time of a task. Also compute its mean and variance.

3. The number of messages, N , arriving to a communications channel per unit time is Poisson distributed with parameter λ . The number of characters, X_i , in the i th message is geometrically distributed with parameter θ . Determine the distribution of the total number of characters, Y , that arrive per unit time. (*Hint:* Y is a random sum.) Determine $G_Y(z)$, $E[Y]$, and $\text{Var}[Y]$.
4. Consider the following concurrent program:

```

CPU1
if B then
  cobegin
    CPU2; I/O2
  coend
else
  I/O1
end.

```

Derive expressions for the completion time distribution and the mean completion time for the whole graph. Assume that the execution time of the statement group CPU_i ($i = 1, 2$) is $\text{EXP}(\mu_i)$, the execution time of the statement group I/O_j ($j = 1, 2$) is $\text{EXP}(\lambda_j)$, and $P(B = \text{true}) = p$. Assume all task executing times are mutually independent. For enhancing such models to allow for task failures, see Sahner et al. [SAHN 1996].

REFERENCES

- [BABL 1975] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models*, Holt, Rinehart & Winston, New York, 1975.
- [BERT 1992] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed., Vol. 1, Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [BOUR 1969] W. G. Bouricius, W. C. Carter, and P. R. Schneider, “Reliability modeling techniques for self-repairing computer systems,” *Proc. 24th Natl. Conf. of the ACM*, Aug. 1969, pp. 295–309.
- [CLAR 1970] A. B. Clarke and R. L. Disney, *Probability and Random Processes for Engineers and Scientists*, Wiley, New York, 1970.
- [COX 1955] D. R. Cox, “A use of complex probabilities in theory of stochastic processes,” *Proc. Cambridge Phil. Soc.*, Vol. **51**, 1955, pp. 313–319.
- [COX 1962] D. R. Cox, *Renewal Theory*, Spottiswoode Ballantyne, London, 1962.
- [DUGA 1989] J. B. Dugan and K. S. Trivedi, “Coverage modeling for dependability analysis of fault-tolerant systems,” *IEEE Trans. Comput.*, **38**(6), pp. 775–787, 1989.
- [GAVE 1973] D. P. Gaver and G. L. Thompson, *Programming and Probability Models in Operations Research*, Brooks/Cole, Monterey, CA, 1973.
- [HESS 2000] G. Hess, personal communication.

- [HOFR 1987] M. Hofri, *Probabilistic Analysis of Algorithms: On Computing Methodologies for Computer Algorithms Performance Evaluation*, Springer-Verlag, New York, 1987.
- [KNUT 1997] D. E. Knuth, *The Art of Computer Programming*, 3rd ed., Vol. I: *Fundamental Algorithms*, Addison-Wesley, Reading, MA, 1997.
- [KNUT 1998] D. E. Knuth, *The Art of Computer Programming*, 2nd ed., Vol. III: *Sorting and Searching*, Addison-Wesley, Reading, MA, 1998.
- [LITT 1973] B. Littlewood and J. L. Verrall, “A Bayesian reliability growth model for computer software”, *J. Royal Statist. Soc., Appl. Statist.*, **22**(3), 1973, pp. 332–346.
- [MEND 1979] H. Mendelson, J. S. Pliskin, and U. Yechiali, “Optimal storage allocation for serial files,” *Commun. ACM*, **22**(2), 1979, pp. 124–130.
- [MUPP 1996] J. Muppala, M. Malhotra, and K. S. Trivedi, “Markov dependability models of complex systems: Analysis techniques”, in S. Ozekici (ed.), *Reliability and Maintenance of Complex Systems*, Springer-Verlag, Berlin, 1996, pp. 442–486.
- [NANN 1998] S. Nannicini and T. Pecorella, “Performance evaluation of polling protocols for data transmission on wireless communication networks” *IEEE Intl. Conf. on Universal Personal Communications*, Vol. 2, 1998, pp. 1241–1245.
- [NG 1976] Y.-W. Ng, Reliability Modeling and Analysis for Fault-Tolerant Computers, Ph.D. dissertation, Computer Science Department, Univ. California at Los Angeles.
- [RAND 1975] B. Randell, “System structure for software fault tolerance,” *IEEE Trans. Software Eng.*, **SE-1**, pp. 202–232, 1975.
- [SAHN 1996] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer System: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.

Chapter 6

Stochastic Processes

6.1 INTRODUCTION

In the previous chapters we have seen the need to consider a collection or a family of random variables instead of a single random variable. A family of random variables that is indexed by a parameter such as time is known as a *stochastic process* (or **chance** or **random process**).

Definition (Stochastic Process). A stochastic process is a family of random variables $\{X(t) \mid t \in T\}$, defined on a given probability space, indexed by the parameter t , where t varies over an index set T .

The values assumed by the random variable $X(t)$ are called **states**, and the set of all possible values forms the **state space** of the process. The state space will be denoted by I .

Recall that a random variable is a function defined on the sample space S of the underlying experiment. Thus the above family of random variables is a family of functions $\{X(t, s) \mid s \in S, t \in T\}$. For a fixed $t = t_1$, $X_{t_1}(s) = X(t_1, s)$ is a random variable [denoted by $X(t_1)$] as s varies over the sample space S . At some other fixed instant of time t_2 , we have another random variable $X_{t_2}(s) = X(t_2, s)$. For a fixed sample point $s_1 \in S$, the expression $X_{s_1}(t) = X(t, s_1)$ is a single function of time t , called a **sample function** or a **realization** of the process. When both s and t are varied, we have the family of random variables constituting a stochastic process.

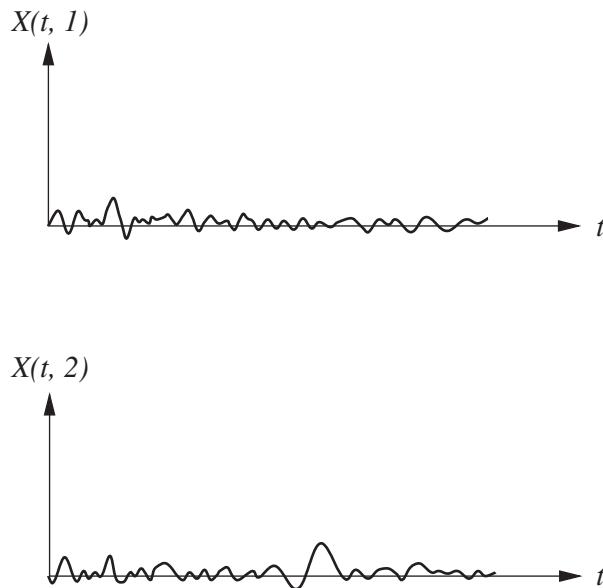


Figure 6.1. Noise voltages across two resistors

Example 6.1 (STAR 1979)

Consider the experiment of randomly choosing a resistor s from a set S of thermally agitated resistors and measuring the noise voltage $X(t, s)$ across the resistor at time t . Sample functions for two different resistors are shown in Figure 6.1.

At a fixed time $t = t_1$, suppose we measure the voltages across all the resistors in the set S , count the number of resistors with a voltage level less than or equal to x_1 and divide this count by the total number of resistors in S . Using the frequency interpretation of probability, this will give the distribution function, $F_{X(t_1)}(x_1) = P(X(t_1) \leq x_1)$, of the random variable $X(t_1)$. This calculation can be repeated at other instants of time t_2, t_3, \dots , to obtain the distribution functions of $X(t_2), X(t_3), \dots$. The joint distribution function of $X(t_1)$ and $X(t_2)$ can similarly be obtained by computing the relative frequency of the event [$X(t_1) \leq x_1$ and $X(t_2) \leq x_2$]. Continuing in this fashion, we can compute the joint distribution function of $X(t_1), X(t_2), \dots, X(t_n)$.

#

If the state space of a stochastic process is discrete, then it is called a **discrete-state process**, often referred to as a “**chain**”. In this case, the state space is often assumed to be $\{0, 1, 2, \dots\}$. Alternatively, if the state space is continuous, then we have a **continuous-state process**. Similarly, if the index set T is discrete, then we have a **discrete-time (parameter) process**; otherwise we have a **continuous-time (parameter) process**. A discrete-time process is also called a **stochastic sequence** and is denoted

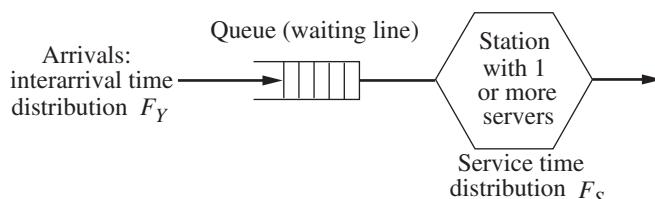
TABLE 6.1. A classification of stochastic processes

		Index set T	
		Discrete	Continuous
State Space	Discrete	Discrete-time stochastic chain	Continuous-time stochastic chain
	Continuous	Discrete-time continuous-state process	Continuous-time continuous-state process

by $\{X_n \mid n \in T\}$. This gives us four different types of stochastic processes, as shown in Table 6.1.

The theory of queues (or waiting lines) provides many examples of stochastic processes. Before introducing these processes, we present a notation to describe the queues. A queue may be generated when customers (jobs) arrive at a station (file server) to receive service (see Figure 6.2). Assume that successive interarrival times Y_1, Y_2, \dots , between jobs are independent identically distributed random variables having a distribution F_Y . Similarly, the service times S_1, S_2, \dots , are assumed to be independent identically distributed random variables having a distribution F_S . Let m denote the number of servers in the station. We use the notation $F_Y/F_S/m$ to describe the queuing system. To denote the specific types of interarrival time and service time distributions, we use the following symbols:

- M (for memoryless) for the exponential distribution
- D for a deterministic or constant interarrival or service time
- E_k for a k -stage Erlang distribution
- H_k for a k -stage hyperexponential distribution
- G for a general distribution
- GI for general independent interarrival times

**Figure 6.2.** A queuing system

Thus $M/G/1$ denotes a single-server queue with exponential interarrival times and an arbitrary service time distribution. The most frequent example of a queue that we will use is $M/M/1$. Besides the nature of the interarrival time and service time distributions, we also need to specify a scheduling discipline that decides how the server is to be allocated to the jobs waiting for service. Unless otherwise specified, we will assume that jobs are selected for service in the order of their arrivals; that is, we will assume FCFS (first-come, first-served) scheduling discipline. Now we will describe various stochastic processes associated with a queue.

Example 6.2

Consider a compute server with jobs arriving at random points in time, queuing for service, and departing from the system after service completion.

Let N_k be the number of jobs in the system at the time of the departure of the k th customer (after service completion). The stochastic process $\{N_k \mid k = 1, 2, \dots\}$ is a discrete-time, discrete-state process with the state space $I = \{0, 1, 2, \dots\}$ and the index set $T = \{1, 2, 3, \dots\}$. A realization of this process is shown in Figure 6.3.

Next let $X(t)$ be the number of jobs in the system at time t . Then $\{X(t) \mid t \in T\}$ is a continuous-time, discrete-state process with $I = \{0, 1, 2, \dots\}$ and $T = \{t \mid 0 \leq t < \infty\}$. A realization of this process is shown in Figure 6.4.

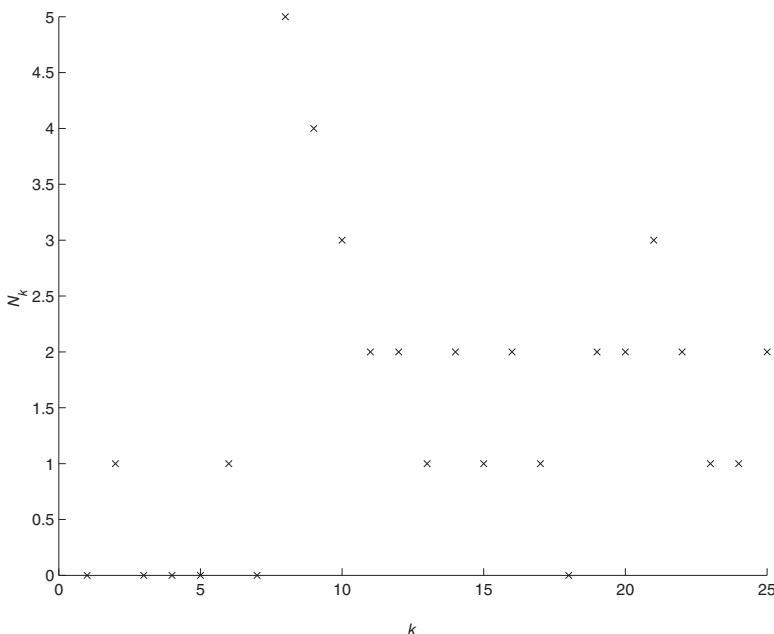


Figure 6.3. Typical sample function of a discrete-time, discrete-state process

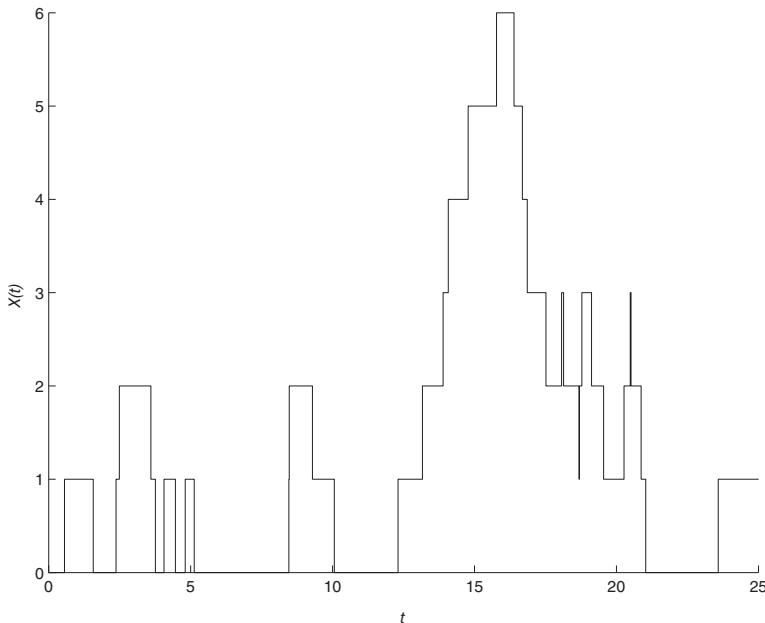


Figure 6.4. Typical sample function of a continuous-time, discrete-state process

Let W_k be the time that the k th customer has to wait in the system before receiving service. Then $\{W_k \mid k \in T\}$, with $I = \{x \mid 0 \leq x < \infty\}$ and $T = \{1, 2, 3, \dots\}$, is a discrete-time, continuous-state process. A realization of this process is shown in Figure 6.5. Finally, let $Y(t)$ denote the cumulative service requirement of all jobs in the system at time t . Then $\{Y(t) \mid 0 \leq t < \infty\}$ is a continuous-time, continuous-state process with $I = [0, \infty)$. A realization of this process is shown in Figure 6.6.

#

Problems

1. Write and run a program to simulate an $M/E_2/1$ queue and obtain realizations of the four stochastic processes defined in Example 6.2. Plot these realizations. You may use a simulation language such as SIMULA or GPSS or you may use one of the standard high-level languages. You will have to generate random deviates of the interarrival time distribution (assume arrival rate $\lambda = 1$ per second) and the service time distribution (assume mean service time 0.8 s) using methods of Chapter 3.
2. Study the process $\{N_k \mid k = 1, 2, \dots\}$ in detail as follows. By varying the seeds for generating random numbers, you get different realizations. For a fixed k , different observed values of N_k for these distinct realizations can be used to estimate the mean and variance of N_k . Using a sample size of 30, estimate $E[N_k]$, $\text{Var}[N_k]$ for $k = 1, 5, 10, 100, 200, 1000$. What can you conclude from this experiment?

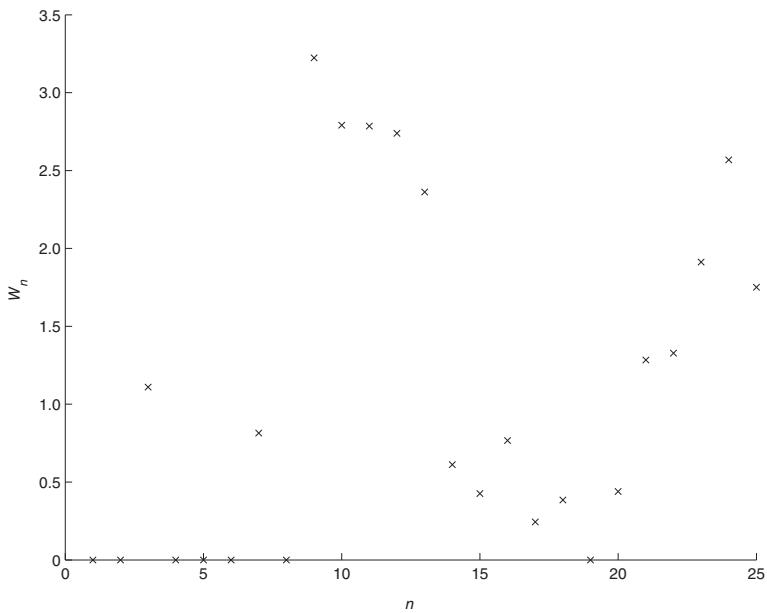


Figure 6.5. Typical sample function of a discrete-time, continuous-state process

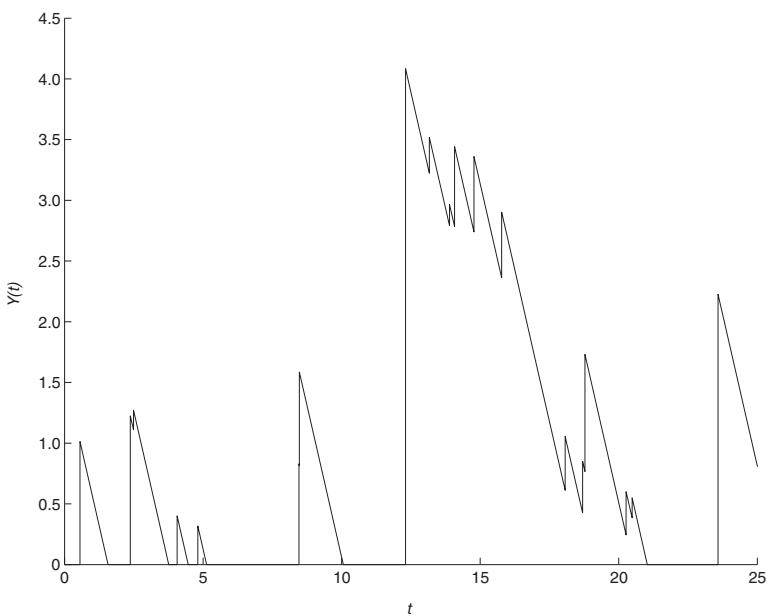


Figure 6.6. Typical sample function of a continuous-time, continuous-state process

6.2 CLASSIFICATION OF STOCHASTIC PROCESSES

For a fixed time $t = t_1$, the term $X(t_1)$ is a simple random variable that describes the state of the process at time t_1 . For a fixed number x_1 , the probability of the event $[X(t_1) \leq x_1]$ gives the CDF of the random variable $X(t_1)$, denoted by

$$F(x_1; t_1) = F_{X(t_1)}(x_1) = P[X(t_1) \leq x_1].$$

$F(x_1; t_1)$ is known as the *first-order distribution* of the process $\{X(t) \mid t \geq 0\}$. Given two time instants t_1 and t_2 , $X(t_1)$ and $X(t_2)$ are two random variables on the same probability space. Their joint distribution is known as the *second-order distribution* of the process and is given by

$$F(x_1, x_2; t_1, t_2) = P[X(t_1) \leq x_1, X(t_2) \leq x_2].$$

In general, we define the n th-order joint distribution of the stochastic process $X(t), t \in T$ by

$$F(\mathbf{x}; \mathbf{t}) = P[X(t_1) \leq x_1, \dots, X(t_n) \leq x_n] \quad (6.1)$$

for all $\mathbf{x} = (x_1, \dots, x_n) \in \Re^n$ and $\mathbf{t} = (t_1, t_2, \dots, t_n) \in T^n$ such that $t_1 < t_2 < \dots < t_n$. Such a complete description of a process is no small task. Many processes of practical interest, however, permit a much simpler description.

For instance, the n th-order joint distribution function is often found to be invariant under shifts of the time origin. Such a process is said to be a strict-sense stationary stochastic process.

Definition (Strictly Stationary Process). A stochastic process $\{X(t) \mid t \in T\}$ is said to be stationary in the strict sense if for $n \geq 1$, its n th-order joint CDF satisfies the condition:

$$F(\mathbf{x}; \mathbf{t}) = F(\mathbf{x}; \mathbf{t} + \boldsymbol{\tau})$$

for all vectors $\mathbf{x} \in \Re^n$ and $\mathbf{t} \in T^n$, and all scalars $\boldsymbol{\tau}$ such that $t_i + \tau \in T$. The notation $\mathbf{t} + \boldsymbol{\tau}$ implies that the scalar $\boldsymbol{\tau}$ is added to all components of vector \mathbf{t} .

We let $\mu(t) = E[X(t)]$ denote the time-dependent mean of the stochastic process. $\mu(t)$ is often called the **ensemble average** of the stochastic process. Applying the definition of strictly stationary process to the first-order CDF, we get $F(x; t) = F(x; t + \boldsymbol{\tau})$ or $F_{X(t)} = F_{X(t+\boldsymbol{\tau})}$ for all $\boldsymbol{\tau}$. It follows that a strict-sense stationary stochastic process has a time-independent mean; that is, $\mu(t) = \mu$ for all $t \in T$.

By restricting the nature of dependence among the random variables $\{X(t)\}$, a simpler form of the n th-order joint CDF can be obtained.

The simplest form of the joint distribution corresponds to a family of independent random variables. Then the joint distribution is given by the product of individual distributions.

Definition (Independent Process). A stochastic process $\{X(t) \mid t \in T\}$ is said to be an **independent process** provided its n th-order joint distribution satisfies the condition:

$$\begin{aligned} F(\mathbf{x}; \mathbf{t}) &= \prod_{i=1}^n F(x_i; t_i) \\ &= \prod_{i=1}^n P[X(t_i) \leq x_i]. \end{aligned} \quad (6.2)$$

As a special case we have the following definition.

Definition (Renewal Process). A **renewal process** is defined as a discrete-time independent process $\{X_n \mid n = 1, 2, \dots\}$ where X_1, X_2, \dots , are independent, identically distributed, nonnegative random variables.

As an example of such a process, consider a system in which the repair (or replacement) after a failure is performed, requiring negligible time. Now the times between successive failures might well be independent, identically distributed random variables $\{X_n \mid n = 1, 2, \dots\}$ of a renewal process.

Though the assumption of an independent process considerably simplifies analysis, such an assumption is often unwarranted, and we are forced to consider some sort of dependence among these random variables. The simplest and the most important type of dependence is the first-order dependence or **Markov dependence**.

Definition (Markov Process). A stochastic process $\{X(t) \mid t \in T\}$ is called a *Markov process* if for any $t_0 < t_1 < t_2 < \dots < t_n < t$, the conditional distribution of $X(t)$ for given values of $X(t_0), X(t_1), \dots, X(t_n)$ depends only on $X(t_n)$:

$$\begin{aligned} P[X(t) \leq x \mid X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0] \\ = P[X(t) \leq x \mid X(t_n) = x_n]. \end{aligned} \quad (6.3)$$

Although this definition applies to Markov processes with continuous state space, we will be mostly concerned with discrete-state Markov processes—specifically, Markov chains. We will study both discrete-time and continuous-time Markov chains.

In many problems of interest, the conditional distribution function (6.3) has the property of invariance with respect to the time origin t_n :

$$P[X(t) \leq x \mid X(t_n) = x_n] = P[X(t - t_n) \leq x \mid X(0) = x_n].$$

In this case the Markov chain is said to be (**time-**) **homogeneous**. Note that the stationarity of the conditional distribution function (6.3) does not imply the stationarity of the joint distribution function (6.1). Thus, a homogeneous Markov process need not be a stationary stochastic process.

For a homogeneous Markov chain, the past history of the process is completely summarized in the current state; therefore, the distribution for the time Y the process spends in a given state must be memoryless:

$$P[Y \leq r + t \mid Y \geq t] = P[Y \leq r]. \quad (6.4)$$

But this implies that the time that a homogeneous, continuous-time Markov chain spends in a given state has an exponential distribution. From (6.4) we have

$$P[Y \leq r] = \frac{P[t \leq Y \leq t + r]}{P[Y \geq t]},$$

that is

$$F_Y(r) = \frac{F_Y(t + r) - F_Y(t)}{1 - F_Y(t)}.$$

If we divide by r and take the limit as r approaches zero, we get

$$F'_Y(0) = \frac{F'_Y(t)}{1 - F_Y(t)},$$

a differential equation with a unique solution:

$$F_Y(t) = 1 - e^{-F'_Y(0)t}$$

Similarly, the time that a homogeneous, discrete-time Markov chain spends in a given state has a geometric distribution.

In modeling practical situations, the restriction on times between state transitions may not hold. A *semi-Markov* process is a generalization of a Markov process where the distribution of time the process spends in a given state is allowed to be general. Further generalization is provided by a Markov regenerative process [KULK 1995].

As a generalization in another direction, consider the number of renewals (repairs or replacements) $N(t)$ required in the interval $(0, t]$, always a quantity of prime interest in renewal processes. The continuous-time process $\{N(t) \mid t \geq 0\}$ is called a **renewal counting process**. Note that, if we restrict the times between renewals to have an exponential distribution, then the corresponding renewal counting process is a special case of a continuous-time Markov chain, known as the **Poisson process**.

A measure of dependence among the random variables of a stochastic process is provided by its **autocorrelation function** R , defined by

$$R(t_1, t_2) = E[X(t_1) \cdot X(t_2)]$$

Note that:

$$R(t_1, t_1) = E[X^2(t_1)]$$

and

$$\text{Cov}[X(t_1), X(t_2)] = R(t_1, t_2) - \mu(t_1)\mu(t_2).$$

The autocorrelation function $R(t_1, t_2)$ of a stationary process depends only on the time difference. (See problem 4 at the end of this section.) Thus, $R(t_1, t_2)$ is a one-dimensional function in this case and is written as $R(\tau)$.

Definition (Wide-Sense Stationary Process). A stochastic process is considered wide-sense stationary if

1. $\mu(t) = E[X(t)]$ is independent of t ,
2. $R(t_1, t_2) = R(0, t_2 - t_1) = R(\tau)$, $t_2 \geq t_1 \geq 0$,
3. $R(0) = E[X^2(t)] < \infty$ (finite second moment).

Note that a strict-sense stationary process with finite second moments is also wide-sense stationary, but the converse does not hold.

Example 6.3 [STAR 1979]

Consider the so-called random-telegraph process. This is a discrete-state, continuous-time process $\{X(t) \mid -\infty < t < \infty\}$ with the state space $\{-1, 1\}$. Assume that these two values are equally likely:

$$P[X(t) = -1] = \frac{1}{2} = P[X(t) = 1], \quad -\infty < t < \infty. \quad (6.5)$$

[This equation implies that the first-order distribution function is stationary in time, but since higher-order distributions may be nonstationary, the stochastic process $X(t)$ need not be stationary in the strict sense.] A typical sample function of the process is shown in Figure 6.7.

Assume that the number of flips, $N(\tau)$, from one value to another occurring in an interval of duration τ is Poisson distributed with parameter $\lambda\tau$. Thus

$$P[N(\tau) = k] = \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!}, \quad k = 0, 1, 2, \dots,$$

where λ is the average number of flips per unit time. Finally assume that the number of flips in a given time interval is statistically independent of the value assumed by the stochastic process $X(t)$ at the beginning of the interval.

For the telegraph process, we obtain

$$\mu(t) = E[X(t)] = -1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = 0, \quad \text{for all } t.$$

$$\begin{aligned} R(t_1, t_2) &= E[X(t_1)X(t_2)] \\ &= P[X(t_1) = 1, X(t_2) = 1] - P[X(t_1) = 1, X(t_2) = -1] \\ &\quad - P[X(t_1) = -1, X(t_2) = 1] + P[X(t_1) = -1, X(t_2) = -1]. \end{aligned}$$

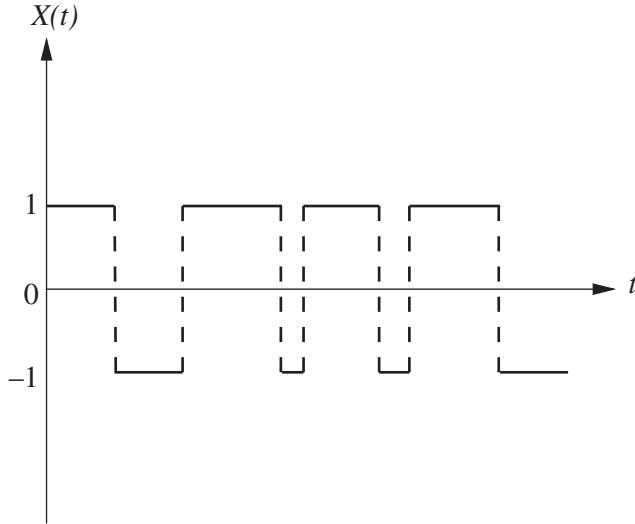


Figure 6.7. Typical sample function of the telegraph process

Since the marginal distribution functions of $X(t_1)$ and $X(t_2)$ are specified by equation (6.5), it can be shown that the events

$$[X(t_1) = 1, X(t_2) = 1] \text{ and } [X(t_1) = -1, X(t_2) = -1]$$

are equally likely. Similarly, the events

$$[X(t_1) = 1, X(t_2) = -1] \text{ and } [X(t_1) = -1, X(t_2) = 1]$$

are equally likely. It follows that the autocorrelation function:

$$\begin{aligned} R(t_1, t_2) &= 2\{P[X(t_1) = 1, X(t_2) = 1] - P[X(t_1) = 1, X(t_2) = -1]\} \\ &= 2\{P[X(t_2) = 1 | X(t_1) = 1]P[X(t_1) = 1] \\ &\quad - P[X(t_2) = -1 | X(t_1) = 1]P[X(t_1) = 1]\} \\ &= P[X(t_2) = 1 | X(t_1) = 1] - P[X(t_2) = -1 | X(t_1) = 1]. \end{aligned}$$

To evaluate the conditional probability $P[X(t_2) = 1 | X(t_1) = 1]$, we observe that the corresponding event is equivalent to the event “An even number of flips in the interval $(t_1, t_2]$.” Let $\tau = t_2 - t_1$. Then

$$\begin{aligned} P[X(t_2) = 1 | X(t_1) = 1] &= P[N(\tau) = \text{even}] \\ &= \sum_{k \text{ even}} \frac{e^{-\lambda\tau} (\lambda\tau)^k}{k!} \\ &= \frac{1 + e^{-2\lambda\tau}}{2} \end{aligned}$$

(by problem 2 at the end of this section). Similarly

$$\begin{aligned} P[X(t_2) = -1 \mid X(t_1) = 1] &= P[N(\tau) = \text{odd}] \\ &= \sum_{k \text{ odd}} \frac{e^{-\lambda\tau} (\lambda\tau)^k}{k!} \\ &= \frac{1 - e^{-2\lambda\tau}}{2}. \end{aligned}$$

Substituting, we get

$$R(t_1, t_2) = e^{-2\lambda\tau}, \quad \tau > 0.$$

Furthermore, since $R(0) = E[X^2(t)] = 1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = 1$ is finite, we conclude that the random-telegraph process is stationary in the wide sense. In Figure 6.8 we have plotted $R(\tau)$ as a function of τ .

‡

Problems

1. Show that the time that a discrete-time homogeneous Markov chain spends in a given state has a geometric distribution.
2. Assuming that the number of arrivals in the interval $(0, t]$ is Poisson distributed with parameter λt , compute the probability of an even number of arrivals. Also compute the probability of an odd number of arrivals.
3. Consider a stochastic process defined on a finite sample space with three sample points. Its description is provided by the specifications of the three sample functions:

$$X(t, s_1) = 3, \quad X(t, s_2) = 3 \cos(t), \quad X(t, s_3) = 4 \sin(t).$$

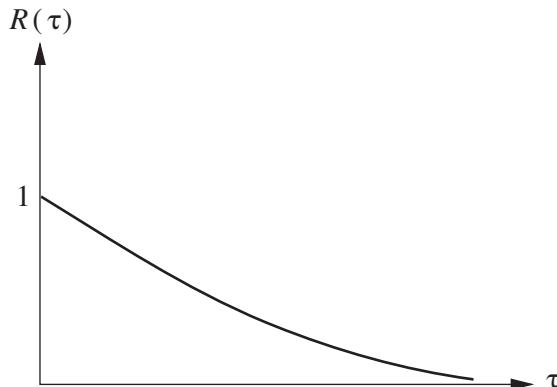


Figure 6.8. Autocorrelation function of the telegraph process

Also given is the probability assignment:

$$P(s_1) = P(s_2) = P(s_3) = \frac{1}{3}.$$

Compute $\mu(t) = E[X(t)]$ and the autocorrelation function $R(t_1, t_2)$. Now answer the following questions: Is the process strict-sense stationary? Is it wide-sense stationary?

4. * Show that the autocorrelation function $R(t_1, t_2)$ of a strict-sense stationary stochastic process depends only on the time difference $(t_2 - t_1)$, if it exists.

6.3 THE BERNOULLI PROCESS

Consider a sequence of independent Bernoulli trials and let the discrete random variable Y_i denote the result of the i th trial, so that the event $[Y_i = 1]$ denotes a success on the i th trial and the event $[Y_i = 0]$ denotes a failure on the i th trial. Further assume that the probability of success on the i th trial, $P[Y_i = 1]$, is p , which is independent of the index i . Then $\{Y_i \mid i = 1, 2, \dots\}$ is a discrete-state, discrete-time, stochastic process, which is stationary in the strict sense. Since the $\{Y_i\}$ are mutually independent, the above process is an independent process known as the **Bernoulli process**. We saw many examples of the Bernoulli process in Chapter 1. Since Y_i is a Bernoulli random variable, we recall that

$$\begin{aligned} E[Y_i] &= p, \\ E[Y_i^2] &= p, \\ \text{Var}[Y_i] &= p(1 - p), \end{aligned}$$

and

$$G_{Y_i}(z) = (1 - p) + pz.$$

On the basis of the Bernoulli process, we may form another stochastic process by considering the sequence of **partial sums** $\{S_n \mid n = 1, 2, \dots\}$, where $S_n = Y_1 + Y_2 + \dots + Y_n$. By rewriting $S_n = S_{n-1} + Y_n$, it is not difficult to see that $\{S_n\}$ is a discrete-state, discrete-time Markov process, since

$$\begin{aligned} P[S_n = k \mid S_{n-1} = k] &= P[Y_n = 0] \\ &= 1 - p, \end{aligned}$$

and

$$\begin{aligned} P[S_n = k \mid S_{n-1} = k - 1] &= P[Y_n = 1] \\ &= p. \end{aligned}$$

We showed in Chapter 2 that S_n is a binomial random variable, so $\{S_n \mid n = 1, 2, \dots\}$ is often called a **binomial process**. Clearly

$$P[S_n = k] = \binom{n}{k} p^k (1-p)^{n-k},$$

$$E[S_n] = np,$$

$$\text{Var}[S_n] = np(1-p),$$

and

$$G_{S_n}(z) = (1 - p + pz)^n.$$

If we refer to successes in a Bernoulli process as arrivals, then we are led to the study of the number of trials between successes or **interarrival times**. Define the discrete random variable T_1 , called the **first-order interarrival time**, to be the number of trials up to and including the first success. Clearly, T_1 is geometrically distributed, so that

$$P[T_1 = i] = p(1-p)^{i-1}, \quad i = 1, 2, \dots,$$

$$E[T_1] = \frac{1}{p},$$

$$\text{Var}[T_1] = \frac{1-p}{p^2},$$

and

$$G_{T_1}(z) = \frac{zp}{1 - z(1-p)}.$$

Now the total number of trials from the beginning of the process until and including the first success is a geometric random variable, and, owing to the mutual independence of successive trials, the number of trials after the $(i-1)$ st success up to and including the i th success has the same distribution as T_1 .

Recall that the geometric distribution possesses the memoryless property, so that the conditional pmf for the remaining number of trials up to and including the next success, given that there were no successes in the first m trials, is still geometric with parameter p . Since an arrival, as defined here, signals a change in state of the sum process $\{S_n\}$, we have that the occupancy time in state S_n , is memoryless.

The notion of the first-order interarrival time can be generalized to higher-order interarrival times. Define the **r th-order interarrival time**, T_r , as the number of trials up to and including the r th success. Clearly, T_r is the r -fold convolution of T_1 , with itself, and therefore T_r has the negative

binomial distribution [using Theorem 2.2(b)]. Then

$$\begin{aligned} P[T_r = i] &= \binom{i-1}{r-1} p^r (1-p)^{i-r}, \quad i = r, r+1, \dots, \quad r = 1, 2, \dots, \\ E[T_r] &= \frac{r}{p}, \\ \text{Var}[T_r] &= \frac{r(1-p)}{p^2}, \end{aligned}$$

and

$$G_{T_r}(z) = \left[\frac{zp}{1-z(1-p)} \right]^r.$$

Example 6.4

Consider a WWW (World Wide Web) cache proxy in a campus computer network [WILL 1996]. With WWW cache proxy, all the WWW page requests generated by the browsers are sent to the cache proxy first. If the cache proxy does not have a copy of the requested file, which is called a “miss”, the cache proxy will retrieve the file from the remote server for the browser. If the proxy has a copy, two cases are now possible; the copy is fresh, which means that it is consistent with the original one at the remote server or the copy is obsolete which means it is inconsistent with the original one at the remote server because the original one has been updated. The former is called a “hit”, so we do not need to retrieve a copy from the remote server and hence we reduce the retrieval latency significantly. The latter is also called a “miss”. We are interested in studying the hit ratio, the probability that the requested file can be served by local copy in the cache proxy. Assume that the probability that a requested file can be found in the cache proxy is $\frac{2}{3}$. Further assume that the two events “requested file is fresh” and “requested file can be found in the cache proxy” are independent, and that successive WWW requests are independent.

Using the tree diagram of Figure 6.9, we may consider the sequence of WWW request as a Bernoulli process with the hit ratio equal to $\frac{3}{4} \cdot \frac{2}{3} = 0.5$. The number of hit requests will then be geometrically distributed with parameter $p = 0.5$.

#

Several generalizations of the Bernoulli process are possible. One possibility is to allow each Bernoulli variable Y_i , a distinct parameter p_i . We retain the assumption that Y_1, Y_2, \dots , are independent random variables. Then the process $\{Y_i \mid i = 1, 2, \dots\}$ is an independent process called the **nonhomogeneous Bernoulli process**.

As an example, let us return to the analysis of program MAX (Chapter 2). Recall that the number of executions, X_n , of the **then** clause for a given array size n is a discrete random variable with the generating function:

$$G_{X_n}(z) = \prod_{i=2}^n \left(\frac{z+i-1}{i} \right).$$

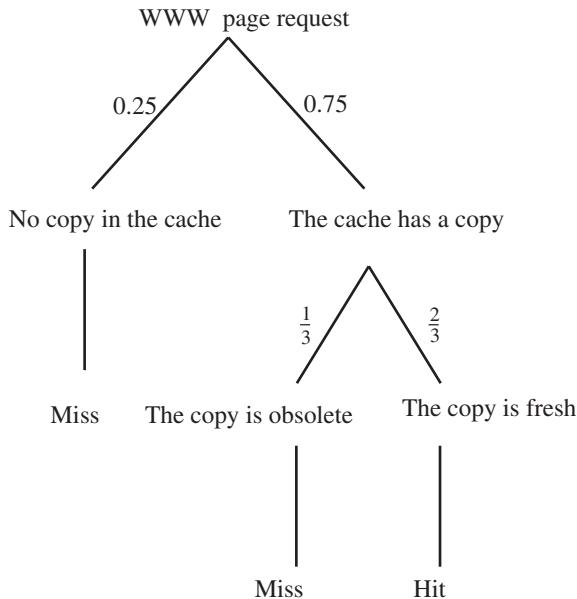


Figure 6.9. The tree diagram for WWW request

By the convolution property of transforms, we can write

$$X_n = \sum_{i=2}^n Y_i,$$

where Y_2, Y_3, \dots , are independent random variables such that

$$G_{Y_i}(z) = \frac{z + i - 1}{i}.$$

But this implies that Y_i is a Bernoulli random variable with parameter $1/i$. Thus, $\{Y_i \mid i = 2, 3, \dots, n\}$ is a nonhomogeneous Bernoulli process, and $\{X_i \mid i = 2, 3, \dots, n\}$ is the corresponding sum process.

Another generalization of the Bernoulli process is to assume that each trial has more than two possible outcomes. Let $\{Y_i \mid i = 1, 2, 3, \dots, n\}$ be a sequence of independent discrete random variables, and define the partial sum $S_n = \sum_{i=1}^n Y_i$. Then the sum process $\{S_n \mid n = 1, 2, \dots\}$ is a Markov chain known as a **random walk**.

Yet another generalization of the Bernoulli process is to study the limiting behavior of the discrete-time process into a continuous-time process. Recall, from Chapter 2, that the Poisson distribution can be derived as a limiting case of the binomial distribution. Now, since the sum process corresponding to the Bernoulli process is the binomial process, the pmf of S_n is $b(k; n, p)$. If n is large and p is small, $b(k; n, p)$ approaches the Poisson pmf $f(k; np)$. Thus, the

number of successes $N(t)$ is approximately Poisson distributed with parameter $\lambda t = np$.

The last generalizations of the Bernoulli process we mention here are the Markov modulated Bernoulli process (MMBP) and the interrupted Bernoulli process (IBP); these will be discussed in Chapter 7.

Problems

1. Show that the first-order interarrival time of the nonhomogeneous Bernoulli process is not memoryless.

6.4 THE POISSON PROCESS

The Poisson process is a continuous-time, discrete-state process that is a good model for many practical situations. Here, the interest is in counting the number of events $N(t)$ occurring in the time interval $(0, t]$. The event of interest may, for example, correspond to:

1. The number of incoming telephone calls to a trunk.
2. The number of job arrivals to a file server.
3. The number of failed components in a large group of initially fault-free components.

We now define the Poisson process. Suppose that the events occur successively in time, so that the intervals between successive events are independent and identically distributed according to an exponential distribution $F(x) = 1 - e^{-\lambda x}$. Let the number of events in the interval $(0, t]$ be denoted by $N(t)$. Then the stochastic process $\{N(t) \mid t \geq 0\}$ is a **Poisson process** with mean rate λ . In the first two situations listed above, λ is called the average arrival rate, while in the third situation λ is called the *failure rate*. From this definition, it is clear that a Poisson process is a renewal counting process for which the underlying distribution is exponential.

An alternative (and equivalent) definition of the Poisson process is as follows: As before, let $N(t)$ be the number of events that have occurred in the interval $(0, t]$. Let the event A denote the occurrence of exactly one event in the interval $(t, t + h]$. Similarly, let B and C , respectively, denote the occurrences of none and more than one events in the same interval. Let $P[A] = p(h)$, $P[B] = q(h)$, and $P[C] = \epsilon(h)$. $N(t)$ forms a Poisson process, provided the following four conditions are met:

1. $N(0) = 0$.
2. Events occurring in nonoverlapping intervals of time are mutually independent.
3. Probabilities $p(h)$, $q(h)$, and $\epsilon(h)$ depend only on the length h of the interval and not on the time origin t .

4. For sufficiently small values of h , we can write (for some positive constant λ):

$$\begin{aligned} p(h) &= P[\text{one event in the interval } (t, t+h)] = \lambda h + o(h), \\ q(h) &= P[\text{no events in the interval } (t, t+h)] = 1 - \lambda h + o(h), \text{ and} \\ \epsilon(h) &= P[\text{more than one event in the interval } (t, t+h)] = o(h) \end{aligned}$$

where $o(h)$ denotes any quantity having an order of magnitude smaller than h ,

$$\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0.$$

Let $p_n(t) = P[N(t) = n]$ be the pmf of $N(t)$. Because of condition 1 above, we have

$$p_0(0) = 1 \text{ and } p_n(0) = 0 \text{ for } n > 0. \quad (6.6)$$

Now consider two successive nonoverlapping intervals $(0, t]$ and $(t, t+\tau]$. To compute $p_n(t+\tau)$, the probability that n events occur in the interval $(0, t+\tau]$, we note that

$$\begin{aligned} &P[n \text{ events in } (0, t+\tau)] \\ &= \sum_{k=0}^n P[k \text{ events in } (0, t] \text{ and } n-k \text{ events in } (t, t+\tau)] \\ &= \sum_{k=0}^n P[k \text{ events in } (0, t)] P[n-k \text{ events in } (t, t+\tau)], \end{aligned}$$

by condition 2 above. Then by condition 3 we have

$$p_n(t+\tau) = \sum_{k=0}^n p_k(t) p_{n-k}(\tau). \quad (6.7)$$

State transitions of the Poisson random process may be visualized as in Figure 6.10. Using equation (6.7) for $n > 0$ and $\tau = h$, we obtain

$$\begin{aligned} p_n(t+h) &= P[N(t+h) = n] \\ &= P[N(t) = n] P[\text{no events in } (t, t+h)] \\ &\quad + P[N(t) = n-1] P[\text{one event in } (t, t+h)] \\ &\quad + \sum_{i=0}^{n-2} P[N(t) = i] P[n-i \text{ events in } (t, t+h)] \end{aligned}$$

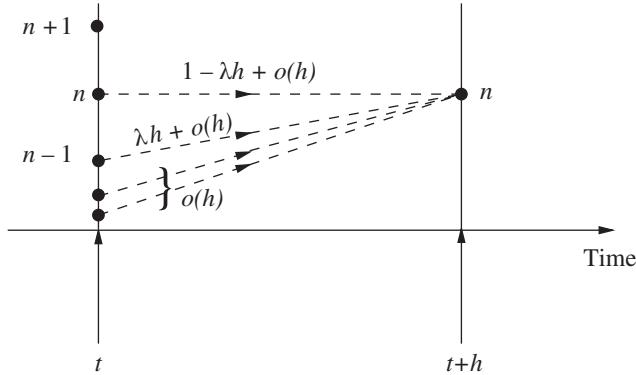


Figure 6.10. State transitions of the Poisson process

$$\begin{aligned}
 &= p_n(t) [1 - \lambda h + o(h)] + p_{n-1}(t) [\lambda h + o(h)] \\
 &\quad + \sum_{i=0}^{n-2} p_i(t) o(h) \\
 &= (1 - \lambda h)p_n(t) + \lambda h p_{n-1}(t) + o(h), \quad n > 0.
 \end{aligned}$$

Similarly

$$p_0(t+h) = (1 - \lambda h)p_0(t) + o(h).$$

After some algebra, we get

$$\lim_{h \rightarrow 0} \frac{p_0(t+h) - p_0(t)}{h} = -\lambda p_0(t)$$

and

$$\lim_{h \rightarrow 0} \frac{p_n(t+h) - p_n(t)}{h} = -\lambda p_n(t) + \lambda p_{n-1}(t).$$

This gives rise to the following differential equations:

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t)$$

and

$$\frac{dp_n(t)}{dt} = -\lambda p_n(t) + \lambda p_{n-1}(t), \quad n > 0. \quad (6.8)$$

It is not difficult to show by induction on n that the solution to equation (6.8) with initial condition (6.6) is

$$p_n(t) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad n > 0.$$

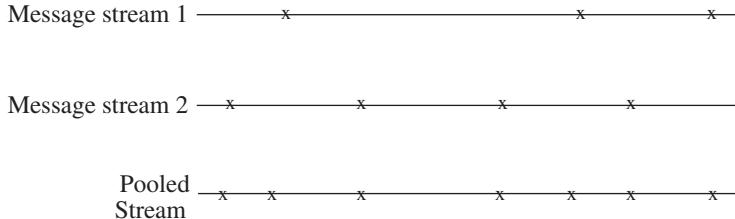


Figure 6.11. Pooling two Poisson streams

Therefore, the number of events $N(t)$ in the interval $(0, t]$ has a Poisson pmf with parameter λt . (Note that this implies that the Poisson process is not a stationary stochastic process.) From Chapter 4 we know that the mean and variance of this distribution are both equal to λt . Therefore, as t approaches infinity, $E[N(t)/t]$ approaches λ and $\text{Var}[N(t)/t]$ approaches zero. In other words, $N(t)/t$ converges to λ as t approaches infinity. Because of this, the parameter λ is called the *arrival rate* of the Poisson process.

The Poisson process plays an important role in queuing theory and reliability theory. One reason for its importance is its analytical tractability, and another reason is a result due to Palm [PALM 1943] and Khinchin [KHIN 1960], which states that under very general assumptions the sum of a large number of independent renewal processes behaves like a Poisson process.

An important generalization of the Poisson process occurs when the rate of arrivals λ is allowed to be a function of t . Such a Poisson process is called the **nonhomogeneous Poisson process** (NHPP). The number of arrivals $N(t)$ is Poisson distributed with parameter $m(t) = \int_0^t \lambda(x)dx$. The parameter $m(t)$ is called the **mean-value function** of the NHPP.

Often we are interested in the **superposition of independent Poisson processes**. For example, suppose that there are two independent Poisson message-arrival streams into a communication channel, with respective arrival rates λ_1 and λ_2 . We are interested in the pooled message-arrival stream. Figure 6.11 shows the arrival times of message stream 1, of message stream 2, and of the pooled stream.

Recall from Chapter 2 that the sum of n independent Poisson random variables is itself Poisson. From this result, it can be shown that the superposition of n independent Poisson processes with respective average rates $\lambda_1, \lambda_2, \dots, \lambda_n$, is also a Poisson process with the average rate $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$, [BTRL 1981]. The notion of superposition of Poisson processes is illustrated in Figure 6.12.

Example 6.5

There are n independent sources of environmental shocks to a component. The number of shocks from the i th source in the interval $(0, t]$, denoted $N_i(t)$, is governed

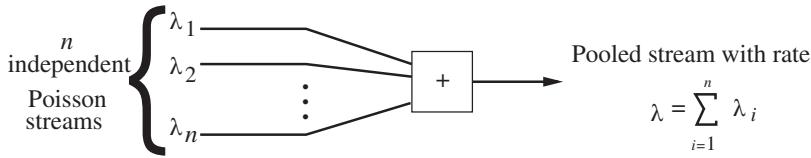


Figure 6.12. Superposition of independent Poisson processes

by a Poisson process with rate λ_i . Then the total number of shocks of all kinds in the interval $(0, t]$ forms a Poisson process with the rate $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$.

#

Example 6.6

Consider a series system of n independent components. The lifetime X_i of the i th component is exponentially distributed with parameter λ_i ($i = 1, 2, \dots, n$). Assume that on failure of the i th component, it is instantaneously replaced by a spare. Now, recalling the relation between the Poisson process and the exponential interevent times, we can conclude that for $i = 1, 2, \dots, n$, the number of failures, $N_i(t)$, of the i th component in the interval $(0, t]$ form a Poisson process with the rate λ_i . Then the total number of system failures $N(t)$ in the interval $(0, t]$ is a superposition of n independent Poisson processes, and hence it is a Poisson process with the rate

$$\lambda = \sum_{i=1}^n \lambda_i.$$

This implies that the times between system failures are exponentially distributed with parameter λ . But this result can be independently verified from our discussion in Chapter 3, where we demonstrated that the lifetime $X = \min\{X_1, X_2, \dots, X_n\}$ of a series system of n independent components with exponential lifetime distribution is itself exponentially distributed with parameter $\lambda = \sum_{i=1}^n \lambda_i$.

#

A similar result holds with respect to the **decomposition of a Poisson process**. Assume that a Poisson process with mean arrival rate λ branches out into n output paths as shown in Figure 6.13. We assume that the successive selections of an output stream form a sequence of generalized Bernoulli trials with p_k ($1 \leq k \leq n$) denoting the probability of the selection of output stream k . Let $\{N(t) \mid t \geq 0\}$ be the input Poisson process, and let $\{N_k(t) \mid t \geq 0\}$ for $1 \leq k \leq n$ denote the output processes. The conditional pmf of $N_k(t)$, $1 \leq k \leq n$, given that $N(t) = m$, is the multinomial pmf (see Chapter 2):

$$\begin{aligned} P[N_1(t) = m_1, N_2(t) = m_2, \dots, N_n(t) = m_n \mid N(t) = m] \\ = \frac{m!}{m_1!m_2!\dots m_n!} p_1^{m_1} p_2^{m_2} \cdots p_n^{m_n}, \end{aligned}$$

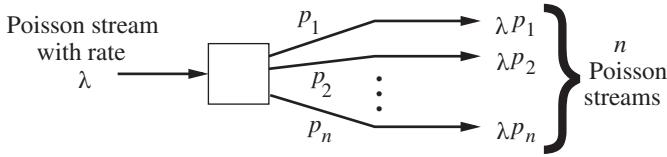


Figure 6.13. Decomposition of a Poisson process

where

$$\sum_{i=1}^n p_i = 1 \text{ and } \sum_{i=1}^n m_i = m.$$

Now, since

$$P[N(t) = m] = e^{-\lambda t} \frac{(\lambda t)^m}{m!},$$

we get the unconditional pmf:

$$\begin{aligned} P[N_1(t) = m_1, N_2(t) = m_2, \dots, N_n(t) = m_n] \\ &= \frac{m!}{m_1! m_2! \dots m_n!} p_1^{m_1} p_2^{m_2} \dots p_n^{m_n} e^{-\lambda t} \frac{(\lambda t)^m}{m!} \\ &= \prod_{i=1}^n e^{-p_i \lambda t} \frac{(p_i \lambda t)^{m_i}}{m_i!}. \end{aligned}$$

But this implies that the random variables $N_1(t), N_2(t), \dots, N_n(t)$ are mutually independent (for all $t \geq 0$) and have Poisson distributions with respective parameters $p_1 \lambda, p_2 \lambda, \dots, p_n \lambda$. This, in turn, verifies that the n output processes are all Poisson with the parameters listed above.

Example 6.7

The number of environmental shocks $N(t)$ experienced by a component in the interval $(0, t]$ is governed by a Poisson process with a rate λ . With probability p_1 , the component will continue to function in spite of the shock, and with probability $p_2 (= 1 - p_1)$ the shock is fatal. Then the random process corresponding to the arrival sequence of fatal shocks is a Poisson process with rate $p_2 \lambda$.

#

Example 6.8

The number of transactions arriving into a database system forms a Poisson process with rate λ . The database consists of n distinct files. An arriving transaction requests the i th file with probability p_i . With the usual independence assumption, the number of requests directed to the i th file ($1 \leq i \leq n$) forms a Poisson process of rate $p_i \lambda$.

#

We have noted in Chapter 3 that the times between successive events of a Poisson process with rate λ are mutually independent and exponentially distributed with parameter λ . In other words, the first-order interarrival time T_1 , of a Poisson process is exponentially distributed. It follows that the r th-order interarrival time, T_r , for this process is an r -stage Erlang random variable with parameter λ .

Thus, if T_k denotes the time of the k th event (from the beginning), then T_k is a k -stage Erlang random variable. Now suppose we know that n arrivals have occurred in the interval $(0, t]$. We wish to compute the conditional distribution of T_k ($1 \leq k \leq n$).

Suppose that exactly one arrival has occurred in the interval $(0, t]$. Then, because of the properties of the Poisson process, we can show that the conditional distribution of arrival time T_1 , is uniform over $(0, t)$:

$$\begin{aligned} P[T_1 \leq x \mid N(t) = 1] &= \frac{P[N(x) = 1 \text{ and } N(t) - N(x) = 0]}{P[N(t) = 1]} \\ &= \frac{\lambda x e^{-\lambda x} \cdot e^{-\lambda(t-x)}}{\lambda t \cdot e^{-\lambda t}} = \frac{x}{t}. \end{aligned}$$

This result is generalized in the following theorem

THEOREM 6.1. Given that $n \geq 1$ arrivals have occurred in the interval $(0, t]$, the conditional joint pdf of the arrival times T_1, T_2, \dots, T_n is given by

$$f[t_1, t_2, \dots, t_n \mid N(t) = n] = \frac{n!}{t^n}, \quad 0 \leq t_1 \leq \dots \leq t_n \leq t.$$

Proof: Let T_{n+1} be the time of the $(n+1)$ st arrival (which occurs after time t). Define the random variables:

$$Y_i = T_i - T_{i-1}, \quad i = 1, \dots, n+1,$$

where T_0 is defined to be equal to zero. It is clear that Y_1, Y_2, \dots, Y_{n+1} , are independent identically distributed random variables such that $Y_i \sim \text{EXP}(\lambda)$ for $i = 1, 2, \dots, n+1$.

Define events A and B such that

$$A = [t_i < T_i \leq t_i + h_i] \quad \text{for } i = 1, 2, \dots, n,$$

and

$$B = [N(t) = n] = [T_n \leq t < T_{n+1}].$$

Now, using the definition of Y_i , the event A can be rewritten as

$$A = [t_i - t_{i-1} - h_{i-1} \leq Y_i \leq t_i - t_{i-1} + h_i], \quad \text{for } i = 1, 2, \dots, n,$$

where t_0 and h_0 are defined to be equal to zero. Now

$$\begin{aligned} A \cap B &= [t_i - t_{i-1} - h_{i-1} \leq Y_i \leq t_i - t_{i-1} + h_i], \quad \text{for } i = 1, 2, \dots, n \\ &\quad \text{and } [T_n \leq t < T_{n+1}] \\ &= [t_i - t_{i-1} - h_{i-1} \leq Y_i \leq t_i - t_{i-1} + h_i], \quad \text{for } i = 1, 2, \dots, n \\ &\quad \text{and } [Y_{n+1} \geq t - t_n - h_n]. \end{aligned}$$

Since the Y_1, Y_2, \dots, Y_{n+1} are independent, we have

$$\begin{aligned} P(A \mid B) &= \frac{P(A \cap B)}{P(B)} \\ &= \frac{\left[\prod_{i=1}^n P(t_i - t_{i-1} - h_{i-1} \leq Y_i \leq t_i - t_{i-1} + h_i) \right] P(Y_{n+1} \geq t - t_n - h_n)}{P[N(t) = n]}. \end{aligned}$$

Dividing both sides by $h_1 h_2 \cdots h_n$, taking the limit as $h_i \rightarrow 0$ ($i = 1, 2, \dots, n$), and recalling that the Y_1, Y_2, \dots, Y_{n+1} are exponentially distributed with parameter λ , we have

$$\begin{aligned} f[t_1, t_2, \dots, t_n \mid N(t) = n] &= \frac{\left[\prod_{i=1}^n \lambda e^{-\lambda(t_i - t_{i-1})} \right] e^{-\lambda(t - t_n)}}{(\lambda t)^n e^{-\lambda t} / n!} \\ &= \frac{n!}{t^n}. \end{aligned}$$

A more general version of the above theorem also holds [BARD 1981]:

THEOREM 6.2. Given that n events have occurred in the interval $(0, t]$, the times of occurrence S_1, S_2, \dots, S_n , when *unordered*, are independent, uniformly distributed over the interval $(0, t]$. In fact, the random variables T_1, T_2, \dots, T_n of Theorem 6.1 are the order statistics of the random variables S_1, S_2, \dots, S_n .

Example 6.9 (The $M/G/\infty$ Queue)

Suppose that jobs arrive at a file server in accordance with a Poisson process of rate λ . The system has an abundance of file servers, so that a job is serviced immediately on its arrival (i.e., no queuing takes place). For analysis, we may assume that the number of servers is infinitely large. Job service times are assumed to be independent general random variables with a common distribution function G .

Let $X(t)$ denote the number of jobs in the system at time t , and let $N(t)$ denote the total number of job arrivals in the interval $(0, t]$. The number of departures $D(t) = N(t) - X(t)$. First we determine the conditional pmf of $X(t)$ given $N(t) = n$.

Consider a job that arrived at time $0 \leq y \leq t$. By Theorem 6.2, the time of arrival Y of the job is uniformly distributed over $(0, t)$; that is, $f_Y(y) = 1/t$, $0 < y < t$. The probability that this job is still undergoing service at time t given that it arrived at time y is $1 - G(t - y)$. Then the unconditional probability that the job is undergoing service at time t is (by the continuous version of the theorem of total probability)

$$\begin{aligned} p &= \int_0^t [1 - G(t - y)] f_Y(y) dy \\ &= \int_0^t \frac{1 - G(t - y)}{t} dy. \end{aligned}$$

Changing variables $x = t - y$, we have

$$p = \int_0^t \frac{1 - G(x)}{t} dx.$$

Since n jobs have arrived and each has independent probability p of not completing by time t , we have a sequence of n Bernoulli trials. Thus

$$P[X(t) = j \mid N(t) = n] = \begin{cases} \binom{n}{j} p^j (1-p)^{n-j}, & j = 0, 1, \dots, n, \\ 0, & \text{otherwise.} \end{cases}$$

Then by the theorem of total probability we have

$$\begin{aligned} P[X(t) = j] &= \sum_{n=j}^{\infty} \binom{n}{j} p^j (1-p)^{n-j} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \\ &= e^{-\lambda t} \frac{(\lambda t p)^j}{j!} \sum_{n=j}^{\infty} \frac{[\lambda t (1-p)]^{n-j}}{(n-j)!} \\ &= e^{-\lambda t p} \frac{(\lambda t p)^j}{j!}. \end{aligned}$$

Thus, the number of jobs in the system at time t has the Poisson distribution with parameter:

$$\lambda' = \lambda t p = \lambda \int_0^t [1 - G(x)] dx.$$

Noting that $\int_0^\infty [1 - G(x)] dx = 1/\mu$ is the average service time, we see that in the limit as t approaches infinity, $\lambda' = \lambda/\mu$. This implies that after a sufficiently long time, the number of jobs in an $M/G/\infty$ queue is Poisson distributed with parameter λ/μ .

#

Several generalizations of the Poisson process are known: the compound Poisson process (see problem 5 below), the nonhomogeneous Poisson process (see Section 8.3.1), and the Markov modulated Poisson process (see Section 8.4.2.1).

Problems

1. Consider a server with Poisson job-arrival stream at an average rate of 60 per hour. Determine the probability that the time interval between successive job arrivals is
 - (a) Longer than 4 min
 - (b) Shorter than 8 min
 - (c) Between 2 and 6 min
2. *Spare-parts problem* [BTRL 1981]. A system requires k components of a certain type to function properly. All components are assumed to have a constant failure rate λ , and their lifetimes are statistically independent. During a mission, component j is put into operation for t_j time units, and a component can fail only while in operation. Determine the number of spares needed (in a common supply of spares) in order to achieve a probability greater than α for the mission to succeed. As an example, let $k = 3$, $t_1 = 1300$, $t_2 = 1500$, $t_3 = 1200$, $\lambda = 0.002$, and $\alpha = 0.90$, and determine the number of needed spares n . Now consider the alternative strategy of keeping a separate supply of spares for each of the three component types. Find the number of spares n_1 , n_2 , and n_3 required to provide an assurance of more than 90% that no shortage for any component type will occur. Show that the former strategy is the better one.
3. When we considered the decomposition of a Poisson process in the text, we assumed that a generalized Bernoulli trial was performed to select the output stream an arriving job should be directed to. Let us now consider a cyclic method of decomposition in which each output stream receives the n th arrival so that the first, $(n + 1)$ st, $(2n + 1)$ st, ..., arrivals are directed to output stream 1, the second, $(n + 2)$ st, $(2n + 2)$ st, ..., arrivals are directed to stream 2, and so on. Show that the interarrival times of any output substream comprise an n -stage Erlang random variable. Note that none of the output streams is Poisson!
4. We are given two independent Poisson arrival streams $\{X_t \mid 0 \leq t < \infty\}$ and $\{Y_t \mid 0 \leq t < \infty\}$ with respective arrival rates λ_x , and λ_y . Show that the number of arrivals of the Y_t process occurring between two successive arrivals of X_t process has a modified geometric distribution with parameter $\lambda_x/(\lambda_x + \lambda_y)$.
5. Consider the generalization of the ordinary Poisson process, called the **compound Poisson process**. In an ordinary Poisson process, we assumed that the probability of occurrence of multiple events in a small interval is negligible with respect to the length of the interval. If the arrival of a message in a LAN (local area network) is being modeled, the counting process may represent the number of bytes (or packets) in a message. In this case suppose that the pmf of the number of bytes in a message is specified:

$$P[\text{number of bytes in a message} = k] = a_k, \quad k \geq 1.$$

Further assume that the message-arrivals form an ordinary Poisson process with rate λ . Then the process $\{X(t) \mid t \geq 0\}$, where $X(t) = \text{number of bytes arriving in}$

the interval $(0, t]$, is a compound Poisson process. Show that generating function of $X(t)$ is given by

$$G_{X(t)}(z) = e^{\lambda t[G_A(z)-1]},$$

where

$$G_A(z) = \sum_{k \geq 1} a_k z^k.$$

6. ★ Prove Theorem 6.1 starting with Theorem 6.2. (*Hint:* Refer to the section on order statistics in Chapter 3.)

6.5 RENEWAL PROCESSES

We have noted that successive interevent times of a Poisson process are independent exponentially distributed random variables. A natural generalization of the Poisson process is obtained by removing the restriction of exponential interevent times. Let X_i be the time between the $(i - 1)$ st and the i th events. Let $\{X_i \mid i = 1, 2, \dots\}$ be a sequence of independent nonnegative identically distributed random variables. This general independent process, $\{X_i \mid i = 1, 2, \dots\}$, is a **renewal process** or a **recurrent process** as defined in Section 6.2. The random variable X_i is interpreted as the time between the occurrence of the i th event and the $(i - 1)$ st event. Note that the restriction of exponential distribution is removed.

The recurrent events (also called *renewals*) may correspond to a job arrival in a server or a telephone call arrival to a trunk. The event may also correspond to the failure of a component in an environment with inexhaustible spares and an instant replacement of a faulty component with a spare one. Similarly, an event may correspond to a reference to a specific page in a paging system, where $\{X_i\}$ will then represent successive intervals between references to this specific page [COFF 1973]. In this case, there will be a distinct renewal process corresponding to each page in the address space of the program being modeled. Here it may be more appropriate to think of X_i as a discrete random variable counting the number of page references between two references to the specific page. In such a case, we have a discrete-state, discrete-time renewal process.

Our development here will assume that X_i is a continuous random variable with the distribution function $F(x)$, called the **underlying distribution** of the renewal process.

Let S_k denote the time from the beginning until the occurrence of the k th event

$$S_k = X_1 + X_2 + \cdots + X_k,$$

and let $F^{(k)}(t)$ denote the distribution function of S_k . Clearly, $F^{(k)}$ is the k -fold convolution of F with itself. For notational convenience, we define

$$F^{(0)}(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0. \end{cases}$$

Our primary interest here is in the number of renewals $N(t)$ in the interval $(0, t]$. The discrete-state, continuous-time process $\{N(t) \mid t \geq 0\}$ is called a **renewal counting process**. This is the generalization of the Poisson process that we alluded to at the beginning of the section. $N(t)$ is called the **renewal random variable** and is easily related to the random variables S_k and S_{k+1} by observing that $N(t) = n$ if and only if $S_n \leq t < S_{n+1}$. But then

$$\begin{aligned} P[N(t) = n] &= P[S_n \leq t < S_{n+1}] \\ &= P[S_n \leq t] - P[S_{n+1} \leq t] \\ &= F^{(n)}(t) - F^{(n+1)}(t). \end{aligned} \tag{6.9}$$

Define the **renewal function** $m(t)$ as the average number of renewals in the interval $(0, t]$:

$$m(t) = E[N(t)].$$

Thus, for example, if $N(t)$ is a Poisson process with rate λ , then its renewal function $m(t) = \lambda t$. Using relation (6.9), we have

$$\begin{aligned} m(t) &= \sum_{n=0}^{\infty} nP[N(t) = n] \\ &= \sum_{n=0}^{\infty} nF^{(n)}(t) - \sum_{n=0}^{\infty} nF^{(n+1)}(t) \\ &= \sum_{n=0}^{\infty} nF^{(n)}(t) - \sum_{n=1}^{\infty} (n-1)F^{(n)}(t) \\ &= \sum_{n=1}^{\infty} F^{(n)}(t) \\ &= F(t) + \sum_{n=1}^{\infty} F^{(n+1)}(t). \end{aligned} \tag{6.10}$$

Noting that $F^{(n+1)}$ is the convolution of $F^{(n)}$ and F , and letting f be the density function of F , we can write

$$F^{(n+1)}(t) = \int_0^t F^{(n)}(t-x)f(x)dx,$$

and therefore

$$m(t) = F(t) + \sum_{n=1}^{\infty} \int_0^t F^{(n)}(t-x)f(x)dx$$

$$\begin{aligned}
&= F(t) + \int_0^t [\sum_{n=1}^{\infty} F^{(n)}(t-x)]f(x)dx \\
&= F(t) + \int_0^t m(t-x)f(x)dx.
\end{aligned} \tag{6.11}$$

This last equation is known as the **fundamental renewal equation**. [In this and subsequent derivations we assume that $\sum F^{(n)}$ satisfies appropriate convergence conditions.]

We shall state without proof the following three theorems [ROSS 1992].

THEOREM 6.3 (The Elementary Renewal Theorem).

$$\lim_{t \rightarrow \infty} \frac{m(t)}{t} = \frac{1}{E[X]}.$$

THEOREM 6.4 (Blackwell's Theorem).

1. If F is not lattice¹, then

$$\lim_{t \rightarrow \infty} [m(t+a) - m(t)] = \frac{a}{E[X]}, \quad \text{for all } a \geq 0.$$

2. If F is lattice with period d , then

$$\lim_{n \rightarrow \infty} P[\text{renewal at time } nd] = \frac{d}{E[X]}.$$

THEOREM 6.5 (Key Renewal Theorem). If F is not lattice, and if $V(t)$ is directly Riemann integrable, then

$$\lim_{t \rightarrow \infty} \int_0^t V(t-x)dm(x) = \frac{1}{E[X]} \int_0^{\infty} V(t)dt.$$

Define the **renewal density** $d(t)$ to be the derivative of the renewal function $m(t)$:

$$d(t) = \frac{dm(t)}{dt}.$$

¹A nonnegative random variable X is said to be lattice if there exists $d \geq 0$ such that $\sum_{n=0}^{\infty} P[X = nd] = 1$. The largest d having this property is said to be the period of X . If X is lattice and F is the distribution function of X , then we say that F is lattice.

For small h , $d(t)h$ is interpreted as the probability of occurrence of a renewal in the interval $(t, t + h]$. Thus in the case of a Poisson process, renewal density $d(t)$ equals the Poisson rate λ . From equation (6.10) we have

$$d(t) = \sum_{n=1}^{\infty} f^{(n)}(t),$$

and, using equation (6.11), we have

$$d(t) = f(t) + \int_0^t d(t-x)f(x)dx. \quad (6.12)$$

The asymptotic renewal rate can be shown to be equal to $1/E[X]$:

$$\begin{aligned} \lim_{t \rightarrow \infty} d(t) &= \lim_{t \rightarrow \infty} \lim_{a \rightarrow 0} \frac{m(t+a) - m(t)}{a} \quad (\text{Theorem 6.4}) \\ &= \frac{1}{E[X]}. \end{aligned}$$

To solve the renewal equation (6.12), we will use Laplace transforms. Our notation here will be somewhat different from that in Chapter 4, since we now associate a Laplace transform with a function (distribution, density, renewal) rather than a random variable. Thus

$$L_f(s) = \int_0^{\infty} e^{-sx} f(x)dx$$

and

$$L_d(s) = \int_0^{\infty} e^{-sx} d(x)dx.$$

Now, using the convolution property of transforms, we get from equation (6.12):

$$L_d(s) = L_f(s) + L_d(s)L_f(s),$$

so that

$$L_d(s) = \frac{L_f(s)}{1 - L_f(s)} \quad (6.13)$$

and

$$L_f(s) = \frac{L_d(s)}{1 + L_d(s)} \quad (6.14)$$

Thus $d(t)$ may be determined from $f(t)$, and conversely, $f(t)$ can be determined from $d(t)$.

Example 6.10

The solution to equation (6.13) may be obtained in closed form for special cases. Assume that the interevent times are exponentially distributed so that

$$f(x) = \lambda e^{-\lambda x}$$

Then

$$L_f(s) = \frac{\lambda}{s + \lambda},$$

and, using (6.13)

$$\begin{aligned} L_d(s) &= \frac{\lambda}{s + \lambda - \lambda} \\ &= \frac{\lambda}{s}. \end{aligned}$$

Since the Laplace transform of a constant k is k/s , we conclude that

$$\begin{aligned} d(t) &= \lambda, \quad t \geq 0, \\ m(t) &= \lambda t, \quad t \geq 0. \end{aligned}$$

These results should be expected, since the renewal counting process in this case is a Poisson process where the average number of events in the interval $(0, t]$ is λt .

In this case, $F^{(n)}(t)$ is a convolution of n identical exponential distributions and hence is an n -stage Erlang distribution:

$$F^{(n)}(t) = 1 - \left[\sum_{k=0}^{n-1} \frac{(\lambda t)^k}{k!} \right] e^{-\lambda t}.$$

Then

$$\begin{aligned} P[N(t) = n] &= F^{(n)}(t) - F^{(n+1)}(t) \\ &= \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \end{aligned}$$

Thus $N(t)$ has a Poisson pmf with parameter λt , as expected.

#

Problems

- ★ Refer to Sevcik et al. [SEVC 1977]. In Section 6.4 we showed that the decomposition of a Poisson process using Bernoulli selection produces Poisson processes

(see Figure 6.13). Now consider a general renewal counting process $N(t)$ with an underlying distribution $F(x)$. Let us send an arrival into one of two streams using a Bernoulli filter, with respective probabilities p and $1 - p$. Show that the Laplace–Stieltjes transform of the interarrival times for the first stream is given by

$$L_{X_1}(s) = \frac{pL_X(s)}{1 - (1-p)L_X(s)}.$$

(Hint: The section on random sums in Chapter 5 will be useful.) Now show that the coefficient of variation of the interarrival time X_1 of the first output stream is given by

$$C_{X_1}^2 = 1 + p(C_X^2 - 1)$$

and the mean by

$$E[X_1] = \frac{1}{p}E[X].$$

2. *[BHAT 1984]. Consider a file server during peak load where the CPU is saturated. Assume that the processing requirement of a job is exponentially distributed with parameter μ . Further assume that a fixed time t_{sys} per job is spent performing overhead functions in the operating system. Let $N(t)$ be the number of jobs completed in the interval $(0, t]$. Show that

$$P[N(t) < n] = \sum_{k=0}^{n-1} e^{-\mu(t-nt_{sys})} \frac{[\mu(t-nt_{sys})]^k}{k!}, \quad \text{if } t \geq nt_{sys}.$$

6.6 AVAILABILITY ANALYSIS

Assume that on the failure of a component, it is repaired and restored to be “as good as new.” Let T_i be the duration of the i th functioning period, and let D_i be the system downtime for the i th repair or replacement. This can be visualized as in Figure 6.14, where \times symbols denote failure times and small circles denote the component repair is completed. We assume that the sequence of random variables $\{X_i = T_i + D_i \mid i = 1, 2, \dots\}$ is mutually independent. Further assume that the T_1, T_2, \dots are identically distributed with the common CDF $W(t)$ and common pdf $w(t)$. Similarly assume that the D_1, D_2, \dots are identically distributed with the common CDF $G(t)$ and pdf $g(t)$. Then X_1, X_2, \dots are also independent and identically distributed,

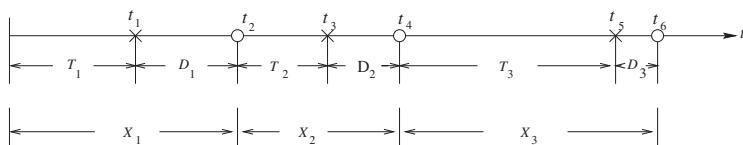


Figure 6.14. A realization of a sequence of failures and repairs

hence $\{X_i \mid i = 1, 2, \dots\}$ is a renewal process. This process is also known as the *alternating renewal process* as it models two states of a system.

Figure 6.14 provides an opportunity to point out the difference between MTTF (mean time to failure) and MTBF (mean time between failures):

$$\text{MTBF} = E[X_i] = E[T_i + D_i] = E[T_i] + E[D_i] = \text{MTTF} + \text{MTTR}.$$

A renewal point of this process corresponds to the event of the completion of a repair. The underlying density $f(t)$ of the renewal process is the convolution of w and g (assuming T_i and D_i are independent). Thus

$$L_f(s) = L_w(s)L_g(s) \quad (6.15)$$

and, using equation (6.13), we have

$$L_d(s) = \frac{L_w(s)L_g(s)}{1 - L_w(s)L_g(s)}. \quad (6.16)$$

The average number of repairs or replacements $m(t)$ in the interval $(0, t]$ has the Laplace transform:

$$L_m(s) = \frac{L_w(s)L_g(s)}{s[1 - L_w(s)L_g(s)]}. \quad (6.17)$$

Define the indicator random variable $I(t)$ of the component so that $I(t) = 1$ when the component is up and $I(t) = 0$ when the component is down. A realization of the stochastic process $\{I(t) \mid t \geq 0\}$ is shown in Figure 6.15. Now we define the **instantaneous availability** (or **point availability**) $A(t)$ of a component (or a system) as the probability that the component is properly functioning at time t , that is, $A(t) = P(I(t) = 1)$. Note that in the absence of a repair or a replacement, availability $A(t)$ is simply equal to the reliability $R(t) = 1 - W(t)$ of the component. The component may be functioning at time t by reason of two mutually exclusive cases: either the component has not

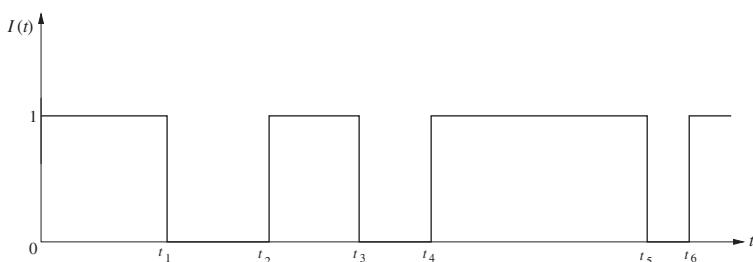


Figure 6.15. A realization of the $I(t)$ process corresponding to Figure 6.14

failed from the beginning (no renewals in the period $(0, t]$) with the associated probability $R(t)$, or the last renewal (repair) occurred at time x , $0 < x < t$, and the component has continued to function since that time. The probability associated with the second case is

$$\int_0^t R(t-x)d(x)dx.$$

Thus:

$$A(t) = R(t) + \int_0^t R(t-x)d(x)dx. \quad (6.18)$$

By conditioning on the first renewal at time x , an alternative form for $A(t)$ can be derived:

$$A(t) = R(t) + \int_0^t A(t-x)dH(x). \quad (6.19)$$

Here $H(t)$ is a convolution of W and G :

$$H(t) = \int_0^t W(t-x)g(x)dx.$$

Note that the instantaneous availability is always greater than or equal to the reliability. Taking Laplace transforms on both sides of equation (6.18), we get

$$\begin{aligned} L_A(s) &= L_R(s) + L_R(s)L_d(s) \\ &= L_R(s)[1 + L_d(s)] \\ &= L_R(s) \left[1 + \frac{L_w(s)L_g(s)}{1 - L_w(s)L_g(s)} \right] \\ &= \frac{L_R(s)}{1 - L_w(s)L_g(s)}, \end{aligned}$$

using equation (6.16). Now, since $R(t) = 1 - W(t)$:

$$\begin{aligned} L_R(s) &= \frac{1}{s} - L_W(s) \\ &= \frac{1}{s} - \frac{L_w(s)}{s} \\ &= \frac{1 - L_w(s)}{s}. \end{aligned}$$

Substituting, we get

$$L_A(s) = \frac{1 - L_w(s)}{s[1 - L_w(s)L_g(s)]}. \quad (6.20)$$

If we are given the failure time and repair time distributions, this equation enables us to compute the instantaneous availability $A(t)$ as a function of time.

Often we are interested in the state of the system after a sufficiently long time has elapsed. For this purpose, we define the *limiting* or *steady-state* availability (or simply *availability*) A as the limiting value of $A(t)$ as t approaches infinity. Here we point out another distinction between the notions of reliability and availability. The “limiting reliability” is given by

$$\lim_{t \rightarrow \infty} R(t) = \lim_{t \rightarrow \infty} [1 - W(t)] = 0,$$

whereas the limiting availability $\lim_{t \rightarrow \infty} A(t)$ is usually nonzero.

In order to derive an expression for the limiting availability, we make use of the following result, known as the *final-value theorem* of Laplace transforms. Let $H(t) = \int_0^t h(x)dx + H(0^-)$. Then, using a table of Laplace transforms (see Appendix D), we get

$$sL_H(s) - H(0^-) = L_h(s) = \int_0^\infty e^{-st} h(t) dt$$

and hence

$$\begin{aligned} \lim_{s \rightarrow 0} sL_H(s) &= \int_0^\infty h(t) dt + H(0^-) \\ &= \lim_{t \rightarrow \infty} \left[\int_0^t h(x) dx \right] + H(0^-) = \lim_{t \rightarrow \infty} H(t). \end{aligned}$$

It follows that the limiting availability A is given by

$$\begin{aligned} A &= \lim_{t \rightarrow \infty} A(t) \\ &= \lim_{s \rightarrow 0} sL_A(s). \end{aligned}$$

Now for small values of s , the following approximations can be used [APOS 1974]:

$$e^{-st} \simeq 1 - st,$$

so:

$$\begin{aligned} L_w(s) &= \int_0^\infty e^{-st} w(t) dt \\ &\simeq \int_0^\infty w(t) dt - s \int_0^\infty tw(t) dt \\ &\simeq 1 - \frac{s}{\lambda}, \end{aligned}$$

where $1/\lambda$ is the mean time to failure (MTTF). Also

$$L_g(s) \simeq 1 - \frac{s}{\mu}$$

where $1/\mu$ is the mean time to repair (MTTR).

Then the limiting availability is

$$\begin{aligned} A &= \lim_{s \rightarrow 0} \left[\frac{1 - \left(1 - \frac{s}{\lambda}\right)}{1 - \left(1 - \frac{s}{\lambda}\right) \left(1 - \frac{s}{\mu}\right)} \right] \\ &= \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} \\ &= \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}. \end{aligned} \quad (6.21)$$

The limiting unavailability is $1 - A = \text{MTTR}/(\text{MTTF} + \text{MTTR})$. This shows that the limiting unavailability depends only on the mean time to failure and mean time to repair, and not on the nature of the distributions of failure times and repair times.

Equation (6.21), for steady-state availability, can be derived without making any distribution assumptions by applying the key renewal theorem with $V(t) = 1 - W(t)$.

Example 6.11

Assume exponential failure and repair time distributions. Then

$$\begin{aligned} w(t) &= \lambda e^{-\lambda t}, \\ g(t) &= \mu e^{-\mu t}, \\ L_w(s) &= \frac{\lambda}{s + \lambda}, \\ L_g(s) &= \frac{\mu}{s + \mu}, \end{aligned}$$

and from equation (6.16) we have

$$\begin{aligned} L_d(s) &= \frac{L_w(s)L_g(s)}{1 - L_w(s)L_g(s)} \\ &= \frac{\lambda\mu}{s[s + (\lambda + \mu)]}. \end{aligned}$$

This can be rewritten as

$$L_d(s) = \frac{\lambda\mu}{(\lambda + \mu)s} - \frac{\lambda\mu}{(\lambda + \mu)^2} \frac{\lambda + \mu}{s + \lambda + \mu}.$$

Inverting yields

$$d(t) = \frac{\lambda\mu}{\lambda + \mu} - \frac{\lambda\mu}{\lambda + \mu} e^{-(\lambda+\mu)t}, \quad t \geq 0.$$

Thus the limiting rate of repairs is given by

$$\begin{aligned} \lim_{t \rightarrow \infty} d(t) &= \frac{\lambda\mu}{\lambda + \mu} \\ &= \frac{1}{\text{MTTF} + \text{MTTR}}. \end{aligned}$$

Now, from equation (6.20), we get

$$\begin{aligned} L_A(s) &= \frac{1 - \frac{\lambda}{(s + \lambda)}}{s \left[1 - \frac{\lambda\mu}{(s + \lambda)(s + \mu)} \right]} \\ &= \frac{s + \mu}{s[s + (\lambda + \mu)]}. \end{aligned}$$

The last expression can be rewritten as

$$L_A(s) = \frac{\frac{\mu}{(\lambda + \mu)}}{s} + \frac{\frac{\lambda}{(\lambda + \mu)}}{s + (\lambda + \mu)}.$$

Inverting, we get

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}.$$

Sometimes we are interested in the expected fraction of time the component is up in a given interval $(0, t]$. The **interval (or average)availability** $A_I(t) = \frac{1}{t} \int_0^t A(x) dx$ can be used for this purpose. With the given assumptions, we have the interval availability:

$$A_I(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{(\mu + \lambda)^2 t} (1 - e^{-(\lambda+\mu)t}).$$

The limiting availability

$$A = \lim_{t \rightarrow \infty} A_I(t) = \lim_{t \rightarrow \infty} A(t) = \frac{\mu}{\lambda + \mu}$$

is exactly the value obtained by the earlier analysis based on a small value of s . Values $A(t)$, $A_I(t)$ and $R(t) = e^{-\lambda t}$ are plotted as functions of time in Figure 6.16. Note that the availability $A(t)$ approaches the reliability $R(t)$ as μ approaches zero;

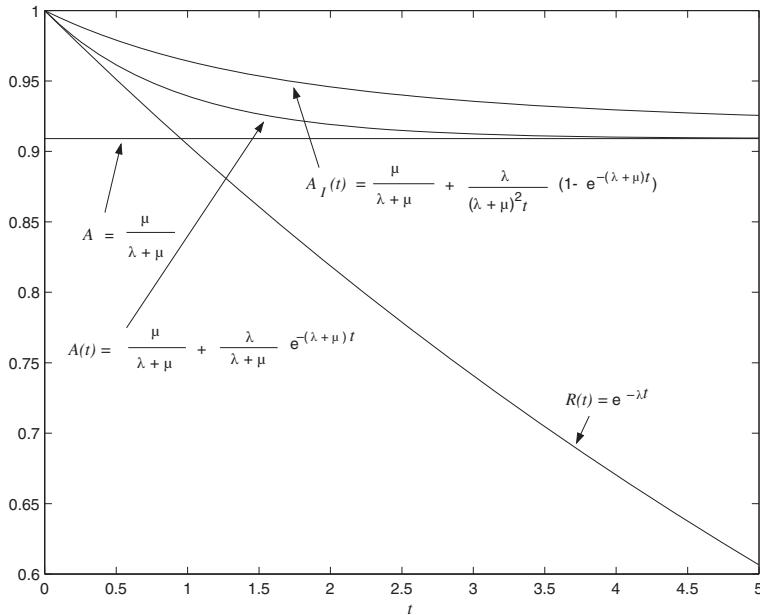


Figure 6.16. Reliability, instantaneous availability, interval availability, and limiting availability as functions of time ($\lambda = 0.1$ and $\mu = 1.0$)

thus in a nonmaintained system the notions of reliability, interval availability and instantaneous availability are synonymous.

¶

Example 6.12

For the workstation–file server (WFS) example, we studied its reliability in Chapter 3 (Example 3.21) and its mean time to failure in Chapter 4 (Example 4.16). Now we study the availability for the WFS example. We consider the simple case of two workstations and a single file server here. Assume that MTTF of an individual workstation is $MTTF_w$ and that of a file server is $MTTF_f$. Let $MTTR_w$ and $MTTR_f$ denote their respective mean times to repair. Then using the reliability block diagram (Figure 3.30) the limiting availability of the WFS system is given by

$$A_{SS} = [1 - (1 - A_w)^2]A_f$$

where $A_w = MTTF_w/(MTTF_w + MTTR_w)$ is the limiting availability of the workstation and $A_f = MTTF_f/(MTTF_f + MTTR_f)$ is the limiting availability of the file server. Note that no distributional assumption was made above. The main assumption is that of independent failures and independent repairs. If we make the assumption that times to failure and times to repair are all exponentially distributed, then the instantaneous availabilities for the workstation and the file server

are given by

$$A_w(t) = \frac{\mu_w}{\lambda_w + \mu_w} + \frac{\lambda_w}{\lambda_w + \mu_w} e^{-(\lambda_w + \mu_w)t}$$

and

$$A_f(t) = \frac{\mu_f}{\lambda_f + \mu_f} + \frac{\lambda_f}{\lambda_f + \mu_f} e^{-(\lambda_f + \mu_f)t},$$

where $\lambda_w = 1/\text{MTTF}_w$, $\lambda_f = 1/\text{MTTF}_f$, $\mu_w = 1/\text{MTTR}_w$ and $\mu_f = 1/\text{MTTR}_f$. Retaining the independence assumption, system instantaneous availability is given by

$$\begin{aligned} A(t) &= [1 - (1 - A_w(t))^2]A_f(t) \\ &= \left[1 - \left(1 - \frac{\mu_w}{\lambda_w + \mu_w} - \frac{\lambda_w}{\lambda_w + \mu_w} e^{-(\lambda_w + \mu_w)t} \right)^2 \right] \\ &\quad \times \left(\frac{\mu_f}{\lambda_f + \mu_f} + \frac{\lambda_f}{\lambda_f + \mu_f} e^{-(\lambda_f + \mu_f)t} \right) \end{aligned}$$

and the interval availability can then be obtained by

$$A_I(t) = \frac{1}{t} \int_0^t A(x) dx.$$

#

Example 6.13 (Random-Request Availability) [LEE 2000]

As an extension of Example 6.11, we consider a repairable system that processes a stream of jobs or tasks arriving randomly during a fixed mission of length T . For such a system Lee [LEE 2000] proposed a new availability measure, random-request availability, which is built on three basic elements: random task arrivals, system state, and operational requirements of the system.

A nonhomogeneous Poisson process with mean value function, $m(T)$ ($= \int_0^T \lambda(t) dt$), was assumed to model the random task arrivals. If no tasks arrive during the entire mission, the mission is considered to be a success (case *a*). Alternatively, one or more tasks arrive during the mission (case *b*). The system was assumed to have up and down states as shown in Figure 6.15. Therefore, at each task arrival time the system can be in one of two states: up or down. For the three types of operational requirements of the system, namely, perfect, $r(n)$ -out-of- n , and scoring systems, the random-request availabilities $A(t_1, t_2, \dots, t_n)$ are first defined for fixed task arrival times t_1, t_2, \dots, t_n .

- Perfect system: $A(t_1, t_2, \dots, t_n)$ is defined by the probability that the system is up at every task arrival time t_1, t_2, \dots, t_n .
- $r(n)$ -out-of- n system: $A(t_1, t_2, \dots, t_n)$ is defined by the probability that the system is up at the time of at least $r(n)$ task arrival times out of n task arrivals.

- Scoring system: $A(t_1, t_2, \dots, t_n)$ is defined by the sum of the products from the probability of the system being up at j out of n task arrival times and the score $s_{j,n}$. A score $s_{j,n}$ denotes the probability of successful completion of the mission.

Now the random-request availabilities for mission of length T , for cases a and b can be expressed as:

$$A_a(T) = \exp [-m(T)] + \sum_{n=1}^{\infty} \left[\exp [-m(T)] \frac{[m(T)]^n}{n!} \times \int \int \cdots \int_{0 \leq t_1 < t_2 < \cdots < t_n \leq T} A(t_1, t_2, \dots, t_n) f(t_1, t_2, \dots, t_n \mid N(T) = n) dt_1 dt_2 \cdots dt_n \right]$$

and

$$A_b(T) = \frac{A_a(T) - \exp [-m(T)]}{1 - \exp [-m(T)]},$$

where $f(t_1, t_2, \dots, t_n \mid N(T) = n)$ is the conditional joint pdf of the task arrival times, given $N(T) = n$ ($n = 1, 2, \dots$). It can be expressed as $(n!/[m(T)]^n) \prod_{i=1}^n \lambda(t_i)$, $0 \leq t_1 < t_2 < \cdots < t_n \leq T$, based on an extension of Theorem 6.1 to the nonhomogeneous Poisson process with mean-value function $m(T)$.

Using the Markov and time-homogeneous properties and following the procedure to derive $A(t)$ of Example 6.11, $A(t_1, t_2, \dots, t_n)$ can be easily obtained for each of the three systems. For instance, given $n = 2$, $r(2) = 1$, and $s_{j,2} = j/2$, $A(t_1, t_2)$ is shown in Table 6.2.

#

Problems

- Return to the base repeater problem of problem 7 in section 1.10. From the fault tree model, derive the expression of steady-state system availability by assuming independent repair.
- You are given a system with n components. The mean time between failures for each component is 100 h and the mean time to repair is 5 h, and each component has its own repair facility. Derive expressions for the limiting availability of the system when
 - All n components are required for the system to function.
 - At least one of the n components should function for the system to function correctly.

Now, assuming that the times to failure and the times to repair for each component are exponentially distributed, write down expressions for the instantaneous availability and the interval availability for both the cases above.

- * Following the development of Section 6.6, derive an expression for $A(t)$, $A_I(t)$ and A , assuming
 - $T_i \sim \text{EXP}(\lambda)$ but the repair times D_1, D_2, \dots are constant at $1/\mu$.
 - T_1, T_2, \dots are two-stage Erlang and $D_i \sim \text{EXP}(\mu)$.

TABLE 6.2. Random-request Availability, $A(t_1, t_2)$

System types	$A(t_1, t_2)$
Perfect	$\left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right)$
$r(n)$ -out-of- n	$\begin{aligned} & \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \\ & + \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \\ & + \left(\frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\mu}{\lambda + \mu} - \frac{\mu}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \end{aligned}$
Scoring	$\begin{aligned} & \frac{1}{2} \left\{ \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \right. \\ & \left. + \left(\frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\mu}{\lambda + \mu} - \frac{\mu}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \right\} \\ & + 1.0 \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t_1} \right) \left(\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)(t_2 - t_1)} \right) \end{aligned}$

4. Define $U_c(t) = \int_c^{c+t} I(\tau) d\tau$ as the cumulative uptime in the interval $(c, c+t]$, and let $D_c(t)$ be the cumulative downtime in the interval $(c, c+t]$. Under the assumption of $T_i \sim \text{EXP}(\lambda)$ and $D_i \sim \text{EXP}(\mu)$, write down expressions for $E[U_c(t)]$ and $E[D_c(t)]$. Show that limiting values (as $c \rightarrow \infty$) of $E[U_c(t)]$ and $E[D_c(t)]$, respectively, are At and $(1-A)t$.
5. Assuming respective steady-state availabilities of .99, .999, .9999, .99999, and .999999, compute the limiting expected downtime (in minutes) for an interval of duration one year.
6. Let \bar{A}_i denote the steady-state unavailability of a component. Then, for a series system of n independent components, show that:

$$\begin{aligned} \bar{A} &= 1 - \prod_{i=1}^n A_i \\ &= \bar{A}_1 + \sum_{i=2}^n A_1 A_2 \dots A_{i-1} \bar{A}_i \\ &\leq \bar{A}_1 + \sum_{i=2}^n \bar{A}_i = \sum_{i=1}^n \bar{A}_i \end{aligned}$$

[Hint: Use the principle of mathematical induction.]

7. Recall Problem 6 at the end of Section 1.10. Assuming that λ_c and λ_v are respective failure rates of a control channel and a voice channel, and that μ_c and μ_v are corresponding repair rates, write down expressions for the steady-state and the instantaneous availabilities for the system.
8. Assuming that $\lambda_i, (i \in \{x, p, d, c\})$ are respective failure rates of an XCVR, a pass-thru, a duplexer, and a combiner, and $\mu_i, (i \in \{x, p, d, c\})$ are corresponding repair rates, write down expressions for the steady-state and the instantaneous availabilities for the system discussed in Example 1.21.
9. Consider a repairable system that processes jobs arriving randomly during a fixed mission duration, as discussed in Example 6.13. Use the following parameters:

- Two types of job arrivals with arrival rate

$$\lambda_1(t) = 0.014t \quad \text{and}$$

$$\lambda_2(t) = 0.14 - 0.014t$$

- $T = 10$

- $\lambda = \frac{1}{2}$ and $\mu = 3$

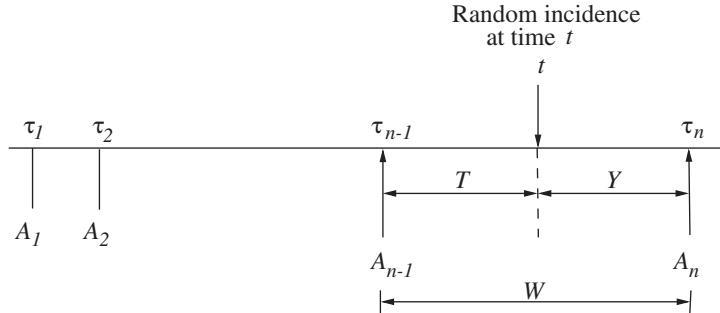
- $r(n) = \begin{cases} [n/2], & \text{for an even } n, \\ [n/2] + 1, & \text{for an odd } n, \end{cases}$

$$s_{j,n} = j/n.$$

- (a) Assuming the system is up at time 0, calculate the random-request availabilities $A_a(T)$ and $A_b(T)$ for the two different job arrival patterns for three types of systems: perfect, $r(n)$ -out-of- n , and scoring system. Show that job arrival with rate $\lambda_2(t)$ gives the higher random-request availabilities than that with the rate $\lambda_1(t)$.
- (b) Assuming the system is down at time 0, repeat the calculations of $A_a(T)$ and $A_b(T)$ as in part (a). Show that job arrivals with rate $\lambda_1(t)$ produce higher random-request availabilities than that with rate $\lambda_2(t)$.
- (c) Interpret the results of part (a) and (b).

6.7 RANDOM INCIDENCE

We have noted that the first-order interevent times of a renewal counting process are independent identically distributed random variables with the density $f_X(x)$. Now consider the experiment of **random incidence** where we pick a random time instant and wait until the occurrence of the next event. Let the random variable Y denote the waiting time until the next event following random entry (see Figure 6.17). Y is often called the **residual lifetime** or the **forward recurrence time**. Let T be the time of the random entry measured from the last event. The random variable T is known as the **backward recurrence time**. For the special case of a Poisson process, X is exponentially distributed and the memoryless property implies that Y is also exponentially distributed. We are interested in deriving the density function $f_Y(y)$ of the residual lifetime for the general renewal counting process.



A_i : The occurrence of the i th event

Figure 6.17. Random incidence

We proceed to obtain the density $f_Y(y)$ in two steps. First we compute the density of the random variable W , denoting the length of the interevent time into which we enter by random incidence. Having obtained $f_W(w)$, we next compute the conditional density $f_{Y|W}$. From these two densities we can obtain the joint density $f_{W,Y}$ and subsequently the marginal (or the unconditional) density $f_Y(y)$.

Since both W and X are first-order interevent times, we may be tempted to conclude that they have the same distributions. In fact, W and X do not have identical distributions, since the experiments on which they are defined are different. The interevent interval W in which our random entry occurs is not a typical interval. A long interval is more likely to be “intercepted” than a short one. We assume that the probability that our random entry falls in an interevent gap of length w is proportional to the gap length w , and to the relative occurrence of such intervals [which is given by $f_X(w)dw$]. Then

$$f_W(w)dw = \frac{wf_X(w)dw}{E[X]},$$

where the denominator provides the normalization factor. Thus the density of W is given by

$$\begin{aligned} f_W(w) &= \frac{wf_X(w)}{E[X]} \\ &= \frac{wf_X(w)}{\int_0^\infty xf_X(x)dx}. \end{aligned} \tag{6.22}$$

Example 6.14

Assume that X is exponentially distributed so that

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0.$$

Then, using equation (6.22), we obtain

$$\begin{aligned} f_W(w) &= \frac{w\lambda e^{-\lambda w}}{\frac{1}{\lambda}} \\ &= \lambda^2 w e^{-\lambda w}, \quad w > 0. \end{aligned}$$

Thus W has the two-stage Erlang distribution. The expected value of W is $E[W] = 2/\lambda$, which is twice as large as $E[X]$. This confirms the assumption that the larger values of w are more likely to be “intercepted” by the random entry. We could have obtained this result from the memoryless property of the exponential distribution by observing that both the forward and the backward recurrence times have exponential distributions with parameter λ and W is the sum of these two independent random variables.

‡

Next we proceed to determine the conditional density of the residual lifetime. Assume that an interevent gap of duration w is intercepted by the random entry. Since a randomly chosen point is selected in this interval, it must be uniformly distributed. Thus

$$f_{Y|W}(y|w) = \begin{cases} 1/w, & 0 < y \leq w, \\ 0, & \text{otherwise.} \end{cases}$$

Now using formula (6.22), we get the joint density:

$$\begin{aligned} f_{W,Y}(w,y) &= f_W(w)f_{Y|W}(y|w) \\ &= \frac{wf_X(w)}{wE[X]} \\ &= \frac{f_X(w)}{E[X]}, \quad 0 < y \leq w < \infty. \end{aligned}$$

Remembering that $y \leq w$, and integrating with respect to w , we get

$$\begin{aligned} f_Y(y) &= \int_{w=y}^{\infty} \frac{f_X(w)dw}{E[X]} \\ &= \frac{1 - F_X(y)}{E[X]} \\ &= \frac{R_X(y)}{E[X]}. \end{aligned} \tag{6.23}$$

For the special case of the Poisson process

$$f_X(y) = \lambda e^{-\lambda y}$$

and from equation (6.23)

$$\begin{aligned} f_Y(y) &= \frac{e^{-\lambda y}}{1/\lambda} \\ &= \lambda e^{-\lambda y} \\ &= f_X(y), \end{aligned}$$

confirming our earlier derivation of the memoryless property! Using a very similar procedure, we can show that the distribution of the backward recurrence time T is identical to the distribution of the forward recurrence time Y .

Our discussion has been directed to continuous-time renewal processes. Corresponding results for discrete-time renewal processes can be easily derived. For example, the discrete analog of equation (6.22) is the pmf:

$$p_w(w) = \frac{w p_x(w)}{E[X]}. \quad (6.24)$$

The discrete analog of equation (6.23) is

$$p_Y(y) = \frac{1 - F_X(y)}{E[X]}. \quad (6.25)$$

Problems

- * We have shown that if two independent Poisson streams are merged, we still get a Poisson stream. Now consider two independent renewal counting processes $N_1(t)$ and $N_2(t)$ being merged into the process $N(t)$ [SEVC 1977]. Let the underlying distribution functions be F_{X_1} , F_{X_2} , and F_X . First show that, conditioned on the fact that the last event in the output stream was contributed by the first stream, the time to next event in the output stream is the minimum of X_1 and $Y_2 = Y(X_2)$, where $Y(X_2)$ is the residual time to next event in input stream 2. Next show that x

$$F_X(t) = \frac{E[X_2]F_{Z_1}(t) + E[X_1]F_{Z_2}(t)}{E[X_1] + E[X_2]}$$

where

$$F_{Z_1}(t) = F_{X_1}(t) + \frac{1 - F_{X_1}(t)}{E[X_2]} \int_0^t [1 - F_{X_2}(x)] dx$$

and

$$F_{Z_2}(t) = F_{X_2}(t) + \frac{1 - F_{X_2}(t)}{E[X_1]} \int_0^t [1 - F_{X_1}(x)] dx.$$

2. Using the results of problem 1, verify that the result of merging two independent Poisson processes produces a Poisson process.
3. Derive equations (6.24) and (6.25).

6.8 RENEWAL MODEL OF PROGRAM BEHAVIOR

We are interested in modeling the memory-referencing behavior of a program executing in a paged virtual memory system. Let $N = \{1, 2, \dots, n\}$ denote the logical address space of an n -page program. For the present model, the dynamic behavior of the program is captured by the reference string w , which has the following sequence:

$$w = r_1 \ r_2 \ \cdots \ r_t \ \cdots ,$$

where each r_t is in N . If $r_t = i$ then a reference is made to the page indexed by i at the t th reference. Clearly, the reference string w is a discrete-time, discrete-state stochastic process.

It is convenient to decompose the stochastic process w into n distinct stochastic processes. For the i th stochastic process ($i = 1, 2, \dots, n$), the event of interest is a reference to the i th page. Assume that the time intervals between references to page i are independent identically distributed random variables with distribution function F_i . Let X_{ij} denote the time between the j th and the $(j-1)$ st reference to page i . Then for each $i = 1, 2, \dots, n$, $\{X_{ij} \mid j = 1, 2, \dots\}$ is a discrete-time renewal process with underlying distribution F_i . We assume that the n renewal processes are independent.

The first-order interevent times of the i th renewal process are interpreted as the interreference intervals for the i th page. $F_i(t)$ and $p_i(t)$ are the corresponding interreference distribution and interreference pmf. The mean interreference interval for page i is given by

$$E[X_i] = \sum_x x p_i(x).$$

Let $d_i(t)$ denote the renewal pmf of the i th process, that is, $d_i(t)$ is the probability of a reference to page i at time t . By using the discrete analog of the key renewal theorem, we obtain the asymptotic value of the renewal rate:

$$d_i = \frac{1}{E[X_i]}. \quad (6.26)$$

Here d_i may be interpreted as the long-term average number of references to page i per unit time. We impose the normalizing condition

$$\sum_{i=1}^n d_i(t) = 1, \quad t \geq 0 \quad (6.27)$$

which assures that one reference (to some page) occurs at every time instant t .

Virtual memory systems usually retain only a portion of a program's logical address space in main memory. For each instant of time, the subset of the address space to retain in main memory is determined by the **paging algorithm**. A popular paging algorithm is the working-set (WS) algorithm, which we shall analyze [COFF 1973].

A program's working set $W(t, \tau)$ at time t is defined to be the set of distinct pages referenced in the time interval $(t - \tau, t]$. For $t < \tau$, $W(t, \tau)$ contains the distinct pages among r_1, r_2, \dots, r_t . The parameter τ is known as the **window size**. The working-set (WS) paging algorithm assures us that a program's working set at time t is in main memory at time t . Now if the next page to be referenced is not in the working set [i.e., $r_{t+1} \notin W(t, \tau)$], then a **page fault** is said to have occurred. Following a page fault, the required page will be loaded by the operating system. Let $w(t, \tau)$ be the size of the working set $W(t, \tau)$. Important measures of the WS algorithm are the asymptotic average working-set size $s(\tau)$:

$$s(\tau) = \lim_{t \rightarrow \infty} E[w(t, \tau)]$$

and the asymptotic average page-fault rate $q(\tau)$.

To compute the average working-set size $s(\tau)$, consider a random incidence in an interreference interval of page i . Let T_i be the backward recurrence time. Then, using (the backward analogue of) equation (6.25), we obtain the pmf of T_i as

$$p_{T_i}(j) = \frac{1 - F_i(j)}{E[X_i]}. \quad (6.28)$$

Given a window size τ , the probability that page i is in memory (i.e., in the working set) at the time of random entry is given by

$$\begin{aligned} P[T_i < \tau] &= \sum_{j=0}^{\tau-1} p_{T_i}(j) \\ &= \sum_{j=0}^{\tau-1} \frac{1 - F_i(j)}{E[X_i]}. \end{aligned} \quad (6.29)$$

Define the indicator random variable Y_{it} = "page i is in memory at time t ." Then the expected value of Y_{it} is given by $E[Y_{it}] = P[Y_{it} = 1]$ which, in the limit, equals $P[T_i < \tau]$. Now the average working-set size is

$$\begin{aligned} s(\tau) &= \sum_{i=1}^n P[T_i < \tau] \\ &= \sum_{i=1}^n \sum_{j=0}^{\tau-1} \frac{1 - F_i(j)}{E[X_i]}. \end{aligned} \quad (6.30)$$

To compute the average page-fault rate $q(\tau)$, we first compute the conditional probability of a page fault given that the i th page is referenced at time t . The required probability is $1 - F_i(\tau)$, since this is the probability that the interreference interval of the page exceeds the window size, that is, the page has not been referenced during the last τ references. Thus

$$P[\text{"page fault at time } t \text{"} \mid r_t = i] = 1 - F_i(\tau).$$

Using the theorem of total probability, we obtain

$$\begin{aligned} P[\text{"page fault at time } t \text{"}] &= \sum_{i=1}^n [1 - F_i(\tau)] P(r_t = i) \\ &= \sum_{i=1}^n [1 - F_i(\tau)] d_i(t). \end{aligned}$$

Taking the limit as t approaches infinity, the left-hand side is the asymptotic average page-fault rate $q(\tau)$, and, using equation (6.26), we get

$$q(\tau) = \sum_{i=1}^n d_i [1 - F_i(\tau)] = \sum_{i=1}^n \frac{1 - F_i(\tau)}{E[X_i]}. \quad (6.31)$$

REFERENCES

- [WILL 1996] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. Fox , “Removal policies in network caches for WWW documents,” *Proc. ACM SIGCOMM’96*, MA, 1996, pp. 293–305.
- [APOS 1974] G. Apostolakis, *Mathematical Methods of Probabilistic Safety Analysis*, Technical Report, School of Engineering and Applied Science, Univ. California at Los Angeles.
- [BARL 1981] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing, to Begin with*, c/o Gordon Pledger, Silver Spring, MD, 1981.
- [BHAT 1984] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd Ed, Wiley, New York, 1984.
- [COFF 1973] E. G. Coffman, Jr. and P. J. Denning, *Operating System Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [KHIN 1960] A. Y. Khintchine, *Mathematical Methods in Queuing Theory*, Griffen, London, 1960.
- [KULK 1995] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, Chapman & Hall, London, 1995.
- [LEE 2000] K. W. Lee, “Stochastic models for random-request availability,” *IEEE Tran. Reliability* **49**(1), 80–84 (2000).

- [PALM 1943] C. Palm, "Intensitätsschwankungen im fernsprechverkehr," *Ericsson Technics* **44**(3), 189 (1943).
- [ROSS 1992] S. M. Ross, *Applied Probability Models with Optimization Applications*, Dover, New York, 1992.
- [SEVC 1977] K. C. Sevcik, A. Levy, S. K. Tripathi, and J. Zahorjan, "Improving approximations of aggregated queuing network subsystems," in *Computer Performance*, M. Reiser and K. M. Chandy (eds.), North-Holland, Amsterdam, 1977.
- [STAR 1979] H. Stark and F. B. Tuteur, *Modern Electrical Communications*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

Chapter 7

Discrete-Time Markov Chains

7.1 INTRODUCTION

A **Markov process** is a stochastic process whose dynamic behavior is such that probability distributions for its future development depend only on the present state and not on how the process arrived in that state. If we assume that the state space, I , is discrete (finite or countably infinite), then the Markov process is known as a **Markov chain**. If we further assume that the parameter space, T , is also discrete, then we have a **discrete-time Markov chain** (DTMC). Such processes are the subject of this chapter. Since the parameter space is discrete, we will let $T = \{0, 1, 2, \dots\}$ without loss of generality.

We choose to observe the state of a system at a discrete set of time points. The successive observations define the random variables $X_0, X_1, X_2, \dots, X_n, \dots$, at time steps $0, 1, 2, \dots, n, \dots$, respectively. If $X_n = j$, then the state of the system at time step n is j . X_0 is the initial state of the system. The Markov property can then be succinctly stated as

$$\begin{aligned} P(X_n = i_n | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \\ = P(X_n = i_n | X_{n-1} = i_{n-1}). \end{aligned} \tag{7.1}$$

Intuitively, equation (7.1) implies that given the “present” state of the system, the “future” is independent of its “past.”

We let $p_j(n)$ denote the pmf of the random variable X_n

$$p_j(n) = P(X_n = j). \tag{7.2}$$

and let the conditional pmf:

$$p_{jk}(m, n) = P(X_n = k \mid X_m = j), \quad 0 \leq m \leq n \quad (7.3)$$

denote the probability that the process makes a transition from state j at step m to state k at step n . Thus, $p_{jk}(m, n)$ is known as the **transition probability function** of the Markov chain. We will only be concerned with **homogeneous Markov chains**—those in which $p_{jk}(m, n)$ depends only on the difference $n - m$ (in this case, the Markov chain is said to have stationary transition probabilities). For such chains, we use the simpler notation

$$p_{jk}(n) = P(X_{m+n} = k \mid X_m = j) \quad (7.4)$$

to denote the **n -step transition probabilities**. In words, $p_{jk}(n)$ is the probability that a homogeneous Markov chain will move from state j to state k in exactly n steps. The one-step transition probabilities $p_{jk}(1)$ are simply written as p_{jk} , thus:

$$p_{jk} = p_{jk}(1) = P(X_n = k \mid X_{n-1} = j), \quad n \geq 1. \quad (7.5)$$

It is convenient to define 0-step transition probabilities by

$$p_{jk}(0) = \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{otherwise.} \end{cases}$$

Since equation (7.1) holds for all values of n , we can use the generalized multiplication rule (GMR of Chapter 1) to obtain the joint probability:

$$\begin{aligned} & P(X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) \\ &= P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \\ & \quad \cdot P(X_n = i_n \mid X_0 = i_0, \dots, X_{n-1} = i_{n-1}) \\ &= P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \\ & \quad \cdot P(X_n = i_n \mid X_{n-1} = i_{n-1}) \\ &= P(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) p_{i_{n-1}, i_n} \\ & \quad \vdots \\ &= p_{i_0}(0) p_{i_0, i_1} \cdots p_{i_{n-1}, i_n}. \end{aligned} \quad (7.6)$$

This implies that all joint probabilities of interest are determined from the initial pmf $p_{i_0}(0) = P(X_0 = i_0)$, and the one-step transition probabilities p_{ij} .

The pmf of the random variable X_0 , often called the **initial probability vector**, is specified as

$$\mathbf{p}(0) = [p_0(0), p_1(0), \dots].$$

The one-step transition probabilities are compactly specified in the form of a **transition probability matrix**:

$$P = [p_{ij}] = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdot & \cdot \\ p_{10} & p_{11} & p_{12} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

The entries of the matrix P satisfy the following two properties:

$$0 \leq p_{ij} \leq 1, \quad i, j \in I; \quad \text{and} \quad \sum_{j \in I} p_{ij} = 1, \quad i \in I.$$

Any such square matrix that has nonnegative entries with row sums all equal to unity is called a *stochastic matrix*.

An equivalent description of the one-step transition probabilities can be given by a directed graph called the *state-transition diagram* (*state diagram*, for short) of the Markov chain. A node labeled i of the state diagram represents state i of the Markov chain and a branch labeled p_{ij} from node i to j implies that the conditional probability (or the one-step transition probability) is

$$P(X_n = j | X_{n-1} = i) = p_{ij}.$$

In order to derive useful performance measures from a DTMC, we assign a *reward* r_i to each state i of the DTMC. Then $Z_n = r_{X_n}$ is the instantaneous reward at step n and $Y_n = \sum_{k=0}^{n-1} Z_k$ is the accumulated reward in the interval $[0, n)$.

Example 7.1

We observe the state of a system (or a component) at discrete points in time. We say that the system is in state 0 if it is operational. If the system is undergoing repair (following a breakdown), then the system state is denoted by state 1. If we assume that the system possesses the Markov property, then we have a two-state discrete-time Markov chain (see Figure 7.1). Further assuming that the Markov chain is homogeneous, we could specify its transition probability matrix by

$$P = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}, \quad 0 \leq a, b \leq 1.$$

The actual values of the entries will have to be estimated from the measurements made on the system using statistical techniques (see Chapter 10).

‡

Example 7.2

Another example of a two-state Markov chain is provided by a communication net consisting of a sequence (or a cascade) of stages of binary communication channels.

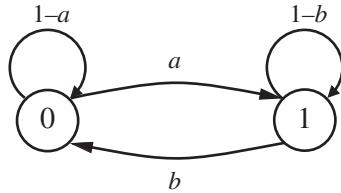


Figure 7.1. The state diagram of a two-state Markov chain

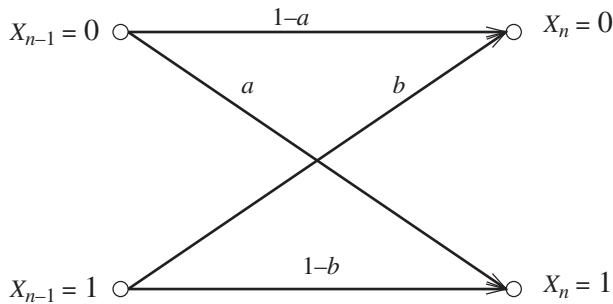


Figure 7.2. A channel diagram

Here X_n denotes the digit leaving the n th stage of the system and X_0 denotes the digit entering the first stage. Assume that the binary communication channels are stochastically identical. The transition probability matrix of the corresponding Markov chain of the communication net can be read off from the channel diagram (see Figure 7.2).

‡

Example 7.3

Consider a sequence of successive software runs at discrete points in time. Each software run has two possible outcomes: success and failure. Associate with the n th software run a binary valued random variable X_n that distinguishes whether the outcome of that particular run resulted in success or failure:

$$X_n = \begin{cases} 0 & \text{denotes a success on the } n\text{th run} \\ 1 & \text{denotes a failure on the } n\text{th run.} \end{cases}$$

The standard way of looking at the sequence of software runs $\{X_n, n \geq 0\}$ is to consider it as a sequence of independent Bernoulli trials, where each trial has probability of success $P(X_n = 0) = p$ and probability of failure $P(X_n = 1) = 1 - p = q$.

The software reliability modeling framework proposed by Goševa-Popstojanova and Trivedi [GOSE 2000] extends the classical software reliability theory in order to consider a sequence of possibly *dependent* software runs, that is, failure correlation.

Suppose that the outcome of each run depends on the outcome of the previous run. Then a sequence of software runs is defined as a sequence of dependent Bernoulli trials where the occurrence of a failure at any particular run depends on the outcome of the previous run:

$$\begin{aligned} P(X_{n+1} = 1 | X_n = 0) &= 1 - p, \\ P(X_{n+1} = 1 | X_n = 1) &= q. \end{aligned}$$

The sequence of dependent Bernoulli trials $\{X_n\}$ defines a discrete-time Markov chain with two states. One of the states denoted by 0 is regarded as success, and the other denoted by 1 as failure. Assuming that the software is not changing, that is, the Markov chain is homogeneous, the transition probability matrix is specified by

$$P = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}, \quad 0 \leq p, q \leq 1. \quad (7.7)$$

The unconditional probability of failure on the $(n+1)$ st run is

$$\begin{aligned} P(X_{n+1} = 1) &= P(X_{n+1} = 1, X_n = 1) + P(X_{n+1} = 1, X_n = 0) \\ &= P(X_{n+1} = 1 | X_n = 1) P(X_n = 1) \\ &\quad + P(X_{n+1} = 1 | X_n = 0) P(X_n = 0) \\ &= q P(X_n = 1) + (1-p) P(X_n = 0) \\ &= q P(X_n = 1) + (1-p) [1 - P(X_n = 1)] \\ &= (1-p) + (p+q-1) P(X_n = 1). \end{aligned} \quad (7.9)$$

In the special case when $q = 1 - p$, the Markov chain describes a sequence of independent Bernoulli trials. In that case the equation (7.9) reduces to $P(X_{n+1} = 1) = 1 - p = q$, which means that the occurrence of failure at any run does not depend on the outcome of the previous run. Thus, subsequent runs are independent with probabilities p and $q = 1 - p$ of being a success and failure.

If $q \neq 1 - p$ then the discrete-time Markov chain describes the sequence of dependent Bernoulli trials that accommodates dependence among successive runs. In this case the outcome of the software run (success or failure) depends on the outcome of the previous run as in equation (7.9). When $q > 1 - p$ runs are positively correlated, that is, if software failure occurs in the n th run, there would be an increased chance that another failure will occur in the next run. It is obvious that in this case failures occur in clusters. When $q < 1 - p$ successive software runs are negatively correlated. In other words, if software failure occurs in the n th run, there would be an increased chance that a success will occur in $(n+1)$ st run, that is, there is a lack of clustering.

If we focus attention on failures and score 1 each time a failure occurs and 0 otherwise, then the accumulated reward $Y_m = X_0 + X_1 + \dots + X_{m-1}$ is the number of runs that have resulted in a failure among m successive software runs. Here we define $X_0 = 0$. If $q = 1 - p$, the number of failures in m runs Y_m is a sum of m mutually independent Bernoulli random variables and has binomial pmf as shown in Chapter 6. When $q \neq 1 - p$, the pmf of Y_m can be derived using the observation that each visit to a given state of the discrete-time Markov chain is a possibly delayed recurrent event [FELL 1968].

7.2 COMPUTATION OF n -STEP TRANSITION PROBABILITIES

We are interested in obtaining an expression for evaluating the n -step transition probability $p_{ij}(n)$ from the one-step transition probabilities $p_{ij}(1) = p_{ij}$. Recall that

$$p_{ij}(n) = P(X_{m+n} = j | X_m = i).$$

Now the probability that the process goes to state k at the m th step, given that $X_0 = i$, is $p_{ik}(m)$; and the probability that the process reaches state j at step $(m + n)$, given that $X_m = k$, is given by $p_{kj}(n)$. The Markov property implies that these two events are independent. Now, using the theorem of total probability (or the fact that the chain must be in some state at step m), we get

$$p_{ij}(m+n) = \sum_k p_{ik}(m)p_{kj}(n). \quad (7.10)$$

This equation is one form of the well-known **Chapman–Kolmogorov equation**, which provides an efficient means of calculating the n -step transition probabilities. This equation need *not* apply to the more general stochastic processes discussed in Chapter 6.

If we let $P(n)$ be the matrix whose (i, j) entry is $p_{ij}(n)$, that is, let $P(n)$ be the matrix of n -step transition probabilities, then we can write equation (7.10) in matrix form (with $m = 1$ and n replaced by $n - 1$):

$$P(n) = P \cdot P(n-1) = P^n. \quad (7.11)$$

Thus the matrix of n -step transition probabilities is obtained by multiplying the matrix of one-step transition probabilities by itself $n - 1$ times. In other words, the problem of finding the n -step transition probabilities is reduced to one of forming powers of a given matrix. It should be clear that the matrix $P(n)$ consists of probabilities and its row sums are equal to unity, so it is a stochastic matrix (see problem 4 at the end of this section).

We can obtain the (marginal) pmf of the random variable X_n from the n -step transition probabilities and the initial probability vector as follows:

$$\begin{aligned} p_j(n) &= P(X_n = j) = \sum_i P(X_0 = i)P(X_n = j | X_0 = i) \\ &= \sum_i p_i(0)p_{ij}(n). \end{aligned} \quad (7.12)$$

If the pmf of X_n (the state of the system at time n) is expressed as the row vector

$$\mathbf{p}(n) = [p_0(n), p_1(n), \dots, p_j(n), \dots],$$

then, from (7.12), we get

$$\mathbf{p}(n) = \mathbf{p}(0)P(n),$$

and from (7.11) we have

$$\mathbf{p}(n) = \mathbf{p}(0)P^n. \quad (7.13)$$

This implies that step-dependent probability vectors of a homogeneous Markov chain are completely determined from the one-step transition probability matrix P and the initial probability vector $\mathbf{p}(0)$.

If the state space I of a Markov chain $\{X_n\}$ is finite, then computing P^n is relatively straightforward, and an expression for the pmf of X_n (for $n \geq 0$) can be obtained using equation (7.13). For Markov chains with a countably infinite state space, computation of P^n poses problems. Therefore, alternative methods for determining the asymptotic behavior (i.e., as n approaches infinity) of P^n and $\mathbf{p}(n)$ have been developed (see Sections 7.3, 7.7, and 7.8).

To illustrate, we will compute P^n for the two-state Markov chain of Examples 7.1 and 7.2, with the transition probability matrix P given by

$$P = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}, \quad 0 \leq a, b \leq 1.$$

A graphical description of the Markov chain is provided by its state diagram shown in Figure 7.1.

The following theorem gives an explicit expression for P^n and hence for $\mathbf{p}(n)$. We will impose the condition $|1 - a - b| < 1$ on the one-step transition probabilities. Since a and b are probabilities, this last condition can be violated only if $a = b = 0$ or $a = b = 1$. These two cases will be treated separately.

THEOREM 7.1. Given a two state Markov chain with the transition probability matrix

$$P = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}, \quad 0 \leq a, b \leq 1, \quad |1-a-b| < 1, \quad (7.14)$$

the n -step transition probability matrix $P(n) = P^n$ is given by

$$P(n) = \begin{bmatrix} \frac{b + a(1-a-b)^n}{a+b} & \frac{a - a(1-a-b)^n}{a+b} \\ \frac{b - b(1-a-b)^n}{a+b} & \frac{a + b(1-a-b)^n}{a+b} \end{bmatrix}.$$

Proof: It is common to give a proof by induction, but we prefer a constructive proof here [BHAT 1984]. Note that

$$\begin{aligned} p_{00}(1) &= p_{00} = 1-a, & p_{01}(1) &= p_{01} = a, \\ p_{10}(1) &= p_{10} = b, & p_{11}(1) &= p_{11} = 1-b. \end{aligned}$$

Using equation (7.10), we get

$$\begin{aligned} p_{00}(1) &= 1 - a, \\ p_{00}(n) &= (1 - a)p_{00}(n - 1) + bp_{01}(n - 1), \quad n > 1. \end{aligned} \quad (7.15)$$

Now since the row sums of P^{n-1} are unity, we have

$$p_{01}(n - 1) = 1 - p_{00}(n - 1),$$

hence (7.15) reduces to

$$\begin{aligned} p_{00}(1) &= 1 - a, \\ p_{00}(n) &= b + (1 - a - b)p_{00}(n - 1), \quad n > 1. \end{aligned} \quad (7.16)$$

This implies that

$$\begin{aligned} p_{00}(n) &= b + b(1 - a - b) + b(1 - a - b)^2 + \dots \\ &\quad + b(1 - a - b)^{n-2} + (1 - a)(1 - a - b)^{n-1} \\ &= b \left[\sum_{k=0}^{n-2} (1 - a - b)^k \right] + (1 - a)(1 - a - b)^{n-1}. \end{aligned}$$

By the formula for the sum of a finite geometric series, we obtain

$$\sum_{k=0}^{n-2} (1 - a - b)^k = \frac{1 - (1 - a - b)^{n-1}}{1 - (1 - a - b)} = \frac{1 - (1 - a - b)^{n-1}}{a + b}.$$

Thus we get

$$p_{00}(n) = \frac{b}{a + b} + \frac{a(1 - a - b)^n}{a + b}.$$

Now $p_{01}(n)$ can be obtained by subtracting $p_{00}(n)$ from unity. Expressions for the two remaining entries can be derived in a similar way. Students familiar with determinants, however, can use the following simpler derivation.

Since $\det(P) = 1 - a - b$, $\det(P^n) = (1 - a - b)^n$, but since P^n is a stochastic matrix, we have

$$\det(P^n) = p_{00}(n) - p_{10}(n),$$

so

$$p_{10}(n) = p_{00}(n) - (1 - a - b)^n.$$

Finally

$$p_{11}(n) = 1 - p_{10}(n).$$

Example 7.4

Consider a cascade of binary communication channels as in Example 7.2. Assume that $a = \frac{1}{4}$ and $b = \frac{1}{2}$. Then, since $|1 - a - b| = \frac{1}{4} < 1$, Theorem 7.1 applies, and

$$P(n) = P^n = \begin{bmatrix} \frac{2}{3} + \frac{1}{3}(\frac{1}{4})^n & \frac{1}{3} - \frac{1}{3}(\frac{1}{4})^n \\ \frac{2}{3} - \frac{2}{3}(\frac{1}{4})^n & \frac{1}{3} + \frac{2}{3}(\frac{1}{4})^n \end{bmatrix}, \quad n \geq 0.$$

Since:

$$P(X_2 = 1 | X_0 = 1) = p_{11}(2) = \frac{3}{8}$$

and

$$P(X_3 = 1 | X_0 = 1) = p_{11}(3) = \frac{11}{32},$$

a digit entering the system as a 1 ($X_0 = 1$) has probability $\frac{3}{8}$ of being correctly transmitted over two stages and probability $\frac{11}{32}$ of being correctly transmitted over three stages.

Assuming the initial probabilities, $P(X_0 = 0) = \frac{1}{3}$ and $P(X_0 = 1) = \frac{2}{3}$ (i.e., $\mathbf{p}(0) = [\frac{1}{3}, \frac{2}{3}]$) we get

$$\mathbf{p}(n) = \mathbf{p}(0)P^n = \left[\frac{2}{3} - \frac{1}{3}(\frac{1}{4})^n, \frac{1}{3} + \frac{2}{3}(\frac{1}{4})^n \right].$$

It is interesting to observe that the two rows of P^n match in their corresponding elements in the limit $n \rightarrow \infty$, and that $\mathbf{p}(n)$ approaches $(\frac{2}{3}, \frac{1}{3})$ as n approaches infinity. In other words, the pmf of X_n becomes independent of n for large values of n . Furthermore, we can verify that with any other initial probability vector, the same limiting pmf of $\mathbf{p}(n)$ is obtained. This important property of *some* Markov chains will be studied in the next section.

#

Example 7.5

Now we consider a cascade of error-free binary communication channels, that is, $a = b = 0$. Clearly, $|1 - a - b| = 1$, and therefore Theorem 7.1 does not apply. The transition probability matrix P is the identity matrix:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The state diagram is shown in Figure 7.3. The two states do not communicate with each other. P^n is easily seen to be the identity matrix. In other words, the chain never changes state, and a transmitted digit is correctly received after an arbitrary number (n) of stages.

#



Figure 7.3. The state diagram of the two-state Markov chain of Example 7.5

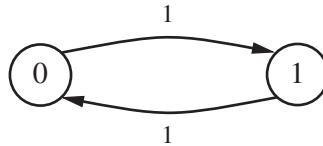


Figure 7.4. The state diagram of a two-state periodic Markov chain

Example 7.6

Consider a cascade of binary channels that are so noisy that the digit transmitted is always complemented. In other words, $a = b = 1$. Once again, Theorem 7.1 does not apply. The matrix P is given by

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and the state diagram is given in Figure 7.4. It can be verified by induction that

$$P^n = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & \text{if } n \text{ is even,} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & \text{if } n \text{ is odd.} \end{cases}$$

This Markov chain has an interesting behavior. Starting in state 0 (or 1), we return to state 0 (state 1) after an even number of steps. Therefore, the time between visits to a given state exhibits a periodic behavior. Such a chain is called a **periodic Markov chain** (with period 2). (Formal definitions are given in the next section.)

#

Example 7.7

Returning to Example 7.3, we now consider the boundary cases when equality in equation (7.7) holds. First, consider the case when $p = q = 1$. This means that if the sequence of software runs starts with failure, all successive runs will fail; or if it starts with success, all successive runs will succeed—that is, the Markov chain remains forever in its initial state as shown in Figure 7.3. Next, when $p = q = 0$, the

outcomes of software runs alternate deterministically between success and failure (as in Figure 7.4); that is, the Markov chain is periodic.

Since the boundary cases are somewhat trivial with no practical interest, the condition $0 < p, q < 1$ is imposed on transition probabilities, which means that the Markov chain considered here is *irreducible* and *aperiodic*. (Formal definitions are given in the next section.)

‡

Problems

- For a cascade of binary communication channels, let $P(X_0 = 1) = \alpha$ and $P(X_0 = 0) = 1 - \alpha, \alpha \geq 0$, and assume that $a = b$. Compute the probability that a 1 was transmitted, given that a 1 was received after the n th stage; that is, compute:

$$P(X_0 = 1 | X_n = 1).$$

- Refer to the Clarke–Disney text [CLAR 1970]. Modify the system of Example 7.1 so that the operating state 0 is split into two states: (a) running and (b) idle. We observe the system only when it changes state. Define X_n as the state of the system after the n th state change, so that

$$X_n = \begin{cases} 0, & \text{if system is running,} \\ 1, & \text{if system is under repair,} \\ 2, & \text{if system is idle.} \end{cases}$$

Assume that the matrix P is

$$P = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Draw the state diagram and compute the matrix P^n .

- Define the **vector** z transform (generating function):

$$G_{\mathbf{p}}(z) = \sum_{n=0}^{\infty} \mathbf{p}(n) z^n.$$

Show that

- $G_{\mathbf{p}}(z) = \mathbf{p}(0)[I - zP]^{-1}$, where I is the identity matrix. Thus P^n is the coefficient of z^n in the matrix power-series expansion of $(I - zP)^{-1}$.
 - Using the result obtained above, give an alternative proof of Theorem 7.1.
- Using equation (7.10) and the principle of mathematical induction, show that
- $$\sum_j p_{ij}(n) = 1 \quad \text{for all } i.$$
- Rewrite equation (7.13) so that only vector-matrix multiplications are involved. Do complexity analysis and write a program to compute $\mathbf{p}(n)$. (First assume that the matrix P is a full matrix, then assume that the matrix P is sparse.)

7.3 STATE CLASSIFICATION AND LIMITING PROBABILITIES

We observed an interesting property of the two-state Markov chain of Example 7.4, in the last section. As $n \rightarrow \infty$, the n -step transition probabilities $p_{ij}(n)$ become independent of both n and i . In other words, all rows of matrix P^n converge toward a common limit (as vectors; i.e., matching in corresponding elements). Now, using the definition of $p_{ij}(n)$ and the theorem of total probability, we have

$$p_j(n) = P(X_n = j) = \sum_i p_i(0)p_{ij}(n),$$

and since $p_{ij}(n)$ depends on neither n nor i in the limit, we conclude that $p_j(n)$ approaches a constant as $n \rightarrow \infty$. This constant is independent of the initial probability vector. We denote the **limiting state probabilities** by

$$v_j = \lim_{n \rightarrow \infty} p_j(n), \quad j = 0, 1, \dots$$

Many (but not all) Markov chains exhibit such a behavior. In order to pursue this topic, we need to classify the states of a Markov chain into those that the system visits infinitely often and those that it visits only a finite number of times. To study the long-run behavior, we need only concentrate on the former type.

Definition (Transient State). A state i is said to be *transient* (or *nonrecurrent*) if and only if there is a positive probability that the process will not return to this state.

For example, if we model a program as a Markov chain, then all except the final state will be transient states. Otherwise, the program has an infinite loop. In general, for a finite Markov chain, we expect that after a sufficient number of steps the probability that the chain is in any transient state approaches zero independent of the initial state.

Let X_{ji} be the number of visits to the state i , starting at j . Then it can be shown [ASH 1970] that

$$E[X_{ji}] = \sum_{n=0}^{\infty} p_{ji}(n).$$

It follows that if the state i is a transient state, then $\sum_{n=0}^{\infty} p_{ji}(n)$ is finite for all j ; hence $p_{ji}(n)$ approaches 0 as n approaches infinity.

Definition (Recurrent State). A state i is said to be **recurrent** if and only if, starting from state i , the process eventually returns to state i with probability one.

An alternative characterization of a recurrent state is that $E[X_{ii}] = \sum_{n=0}^{\infty} p_{ii}(n)$ is infinite. It can be verified from the form of P^n that both states of the chains in Examples 7.4–7.6 are recurrent.

For recurrent states, the time to reentry is important. Let $f_{ij}(n)$ be the conditional probability that the first visit to state j from state i occurs in exactly n steps. If $i = j$, then we refer to $f_{ii}(n)$ as the probability that the first return to state i occurs in exactly n steps. These probabilities are related to the transition probabilities by [PARZ 1962]

$$p_{ij}(n) = \sum_{k=1}^n f_{ij}(k)p_{jj}(n-k), n \geq 1.$$

Let f_{ij} denote the probability of ever visiting state j , starting from state i . Then

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}(n).$$

It follows that state i is recurrent if $f_{ii} = 1$ and transient if $f_{ii} < 1$. If $f_{ii} = 1$, define the **mean recurrence time** of state i by

$$\mu_i = \sum_{n=1}^{\infty} n f_{ii}(n).$$

A recurrent state i is said to be **recurrent nonnull** (or **positive recurrent**) if its mean recurrence time μ_i is finite and is said to be **recurrent null** if its mean recurrence time is infinite.

Definition. For a recurrent state i , $p_{ii}(n) > 0$ for some $n \geq 1$. Define the **period** of state i , denoted by d_i , as the greatest common divisor of the set of positive integers n such that $p_{ii}(n) > 0$.

Definition. A recurrent state i is said to be **aperiodic** if its period $d_i = 1$, and **periodic** if $d_i > 1$.

In Example 7.6, both states 0 and 1 are periodic with period 2. States of Examples 7.4 and 7.5 are all aperiodic.

Definition. A state i is said to be an **absorbing** state if and only if $p_{ii} = 1$.

Both states of the chain in Example 7.5 are absorbing. Once a Markov chain enters such a state, it is destined to remain there forever.

Both transient and recurrent states may coexist in the same Markov chain. State classification of discrete-time Markov chains is summarized in Figure 7.5 [MOLL 1989]. Having defined the properties of individual states, we now define relationship between states.

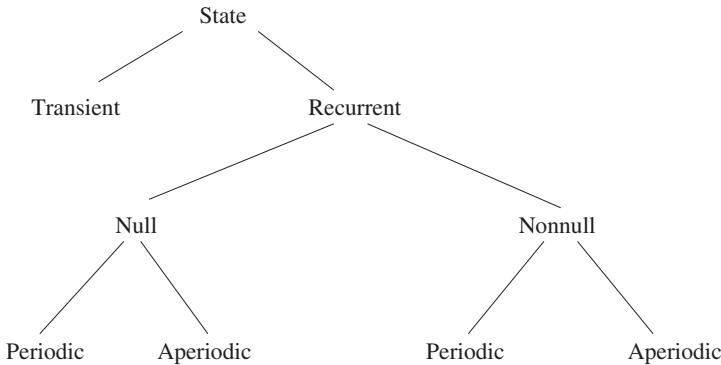


Figure 7.5. The state classification of discrete-time Markov chains

Definition (Communicating States). Two states i and j communicate if directed paths from i to j and vice-versa exist.

Definition (Closed Set of States). A set C of communicating states is a closed set if no state outside C can be reached from any state in C .

The states of a finite Markov chain can be partitioned in a unique manner into subsets C_1, C_2, \dots, C_k so that C_k consists of all transient states and for $i = 1, 2, \dots, k-1$, C_i is a closed set of recurrent nonnull states. It is possible for C_i to contain only one state in which case it will be an absorbing state. It is also possible for C_k to be the empty set and for $k = 2$ in which case all states belong to a single communicating class; such chains are known as *irreducible Markov chains*.

Definition (Irreducible Markov Chain). A Markov chain is said to be **irreducible** if every state can be reached from every other state in a finite number of steps. In other words, for all $i, j \in I$, there is an integer $n \geq 1$ such that $p_{ij}(n) > 0$.

Markov chains of Examples 7.4 and 7.6 are both irreducible. Feller [FELL 1968] has shown that all states of an irreducible Markov chain are of the same type. Thus, if one state of an irreducible chain is aperiodic, then so are all the states, and such a Markov chain is called *aperiodic*. The Markov chain of Example 7.4 is both irreducible and aperiodic. Similarly, if one state of an irreducible chain is periodic, then all states are periodic and have the same period; if one state is transient, then so are all states; and if one state is recurrent, then so are all states.

The n -step transition probabilities $p_{ij}(n)$ of finite, irreducible, aperiodic Markov chains become independent of i and n as $n \rightarrow \infty$. The limiting state probability is

$$\begin{aligned}
v_j &= \lim_{n \rightarrow \infty} p_j(n) = \lim_{n \rightarrow \infty} \sum_i p_i(0)p_{ij}(n) \\
&= \sum_i p_i(0) \left[\lim_{n \rightarrow \infty} p_{ij}(n) \right] \\
&= \lim_{n \rightarrow \infty} p_{ij}(n) \sum_i p_i(0) \\
&= \lim_{n \rightarrow \infty} p_{ij}(n).
\end{aligned}$$

But this implies that P^n converges to a matrix V (with identical rows $\mathbf{v} = [v_0, v_1, \dots]$) as $n \rightarrow \infty$.

Assume that for a given Markov chain the limiting probabilities v_j exist for all states $j \in I$ (where v_j do not depend on the initial state i). Then it can be shown [ASH 1970] that $\sum_{j \in I} v_j \leq 1$. Furthermore, either all $v_j = 0$ (this can happen only for a chain with an infinite number of states) or $\sum_{j \in I} v_j = 1$. In the latter case, the numbers $v_j, j \in I$, are said to form a **steady-state probability vector**. Thus we require that the limiting probabilities exist, that they are independent of the initial state, and that they form a probability vector. Over a long period the influence of the initial state (or the effect of “startup” transients) has died down and the Markov chain has reached a *steady state*. The probability v_j is sometimes interpreted as the *long-run proportion* of time the Markov chain spends in state j .

Now from the theorem of total probability, we have

$$p_j(n) = \sum_i p_i(n-1)p_{ij}.$$

Then, if we have

$$\lim_{n \rightarrow \infty} p_j(n) = v_j = \lim_{n \rightarrow \infty} p_j(n-1),$$

we get

$$v_j = \sum_i v_i p_{ij}, \quad j = 0, 1, 2, \dots, \tag{7.17}$$

or in matrix notation

$$\mathbf{v} = \mathbf{v}P. \tag{7.18}$$

(In other words, \mathbf{v} is a left eigenvector of P associated with the eigenvalue $\lambda = 1$.) This gives us a system of linear equations in the unknowns $[v_0, v_1, \dots]$. Since \mathbf{v} is a probability vector, we also expect that

$$v_j \geq 0, \sum_j v_j = 1. \tag{7.19}$$

Any vector \mathbf{x} that satisfies the properties (7.18) and (7.19) is also known as a **stationary probability** vector of the Markov chain.

We state the following important theorems without proof [PARZ 1962]

THEOREM 7.2. For an aperiodic Markov chain, the limits $v_j = \lim_{n \rightarrow \infty} p_j(n)$ exist.

THEOREM 7.3. For any irreducible, aperiodic Markov chain, the limiting state probabilities $v_j = \lim_{n \rightarrow \infty} p_j(n) = \lim_{n \rightarrow \infty} p_{ij}(n)$ exist and are independent of the initial probability vector $\mathbf{p}(0)$.

THEOREM 7.4. For an irreducible, aperiodic Markov chain, with all states recurrent non-null, the limiting probability vector $\mathbf{v} = [v_0, v_1, \dots]$ is the unique stationary probability vector [satisfying equations (7.18) and (7.19)], hence \mathbf{v} is also known as the steady-state probability vector.

It can be shown that all states of a finite, irreducible Markov chain are recurrent nonnull. Then, for a finite, aperiodic, irreducible Markov chain, we can obtain the steady-state probabilities rather easily by solving a system of linear equations, since Theorem 7.4 applies. Starting with an initial (guess) probability vector $\mathbf{v}^{(0)} = \mathbf{p}(0)$, we use successive substitution to solve the fixed-point equation (7.18):

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)}P, \quad k = 0, 1, 2, \dots \quad (7.20)$$

until convergence is reached. This method of solution for the steady-state probability vector of a homogeneous DTMC is known as the *power method*. For further details, see works by Bolch et al. and Stewart [BOLC 1998, STEW 1994].

For chains with an infinite number of states, we can often solve the equations by using the method of generating functions [recall problem 3(b) at the end of the previous section] or by exploiting the special structure of the matrix P (see, for example, the section 7.8 on birth-death processes).

From the steady-state probability vector, the steady-state expected reward can be calculated:

$$E[Z] = \lim_{n \rightarrow \infty} E[Z_n] = \sum_j v_j r_j. \quad (7.21)$$

Example 7.8

Returning to the periodic Markov chain of Example 7.6, we see that Theorem 7.2 does not apply. In fact, if we let the initial probability vector $\mathbf{p}(0) = [p, 1 - p]$, then

$$\mathbf{p}(n) = \begin{cases} [p, 1 - p], & \text{if } n \text{ is even,} \\ [1 - p, p], & \text{if } n \text{ is odd.} \end{cases}$$

Thus $\mathbf{p}(n)$ does not have a limit. It is interesting to note that, although limiting probabilities do not exist, stationary probabilities are unique and are easily computed to be $v_0 = v_1 = \frac{1}{2}$ with the use of (7.18) and (7.19).

‡

Example 7.9

Returning to the Markov chain of Example 7.5, we see that it is not irreducible (since we cannot go from one state to another) and that Theorem 7.3 does not apply. Although $\mathbf{p}(n)$ has a limit [in fact, $\mathbf{p}(n) = \mathbf{p}(0)$], the limit is dependent on the initial probability vector $\mathbf{p}(0)$.

‡

Example 7.10

We consider the two-state Markov chain of the last section (Examples 7.1 and 7.2 and Theorem 7.1) with the condition $0 < a, b < 1$. This implies that $|1 - a - b| < 1$ and Theorem 7.1 applies. From this we conclude that

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \frac{b}{a+b} & \frac{a}{a+b} \\ \frac{b}{a+b} & \frac{a}{a+b} \end{bmatrix} = \begin{bmatrix} v_0 & v_1 \\ v_0 & v_1 \end{bmatrix}.$$

Thus the steady-state probability vector is

$$\mathbf{v} = [v_0, v_1] = \left[\frac{b}{a+b}, \frac{a}{a+b} \right].$$

This result can also be derived using Theorem 7.4, since the chain is irreducible, finite, and aperiodic. Then, using equation (7.18), we have

$$[v_0, v_1] = [v_0, v_1] \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}$$

or

$$v_0 = (1-a)v_0 + bv_1$$

and

$$v_1 = av_0 + (1-b)v_1.$$

After rearranging the above equations, we have

$$av_0 - bv_1 = 0,$$

$$-av_0 + bv_1 = 0.$$

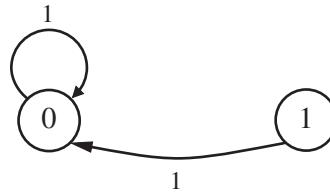


Figure 7.6. The state diagram of a Markov chain with one absorbing state and one transient state

Note that these two equations are linearly dependent, and thus we need one more equation [supplied by condition (7.19)]:

$$v_0 + v_1 = 1.$$

Solving the system of equations, we get the stationary probability vector as derived earlier:

$$[v_0, v_1] = \left[\frac{b}{a+b}, \frac{a}{a+b} \right]$$

‡

Example 7.11

Consider a two-state Markov chain with $a = 0$ and $b = 1$, so that

$$P = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

with the state diagram shown in Figure 7.6.

In the case shown in Figure 7.6, state 1 is transient and state 0 is absorbing. The chain is not irreducible, but the limiting state probabilities exist (since Theorem 7.2 applies) and are given by $v_0 = 1$ and $v_1 = 0$. This says that eventually the chain will remain in state 0 (after at most one transition).

‡

Example 7.12

Consider a model of a program executing on a computer system with m I/O devices and a CPU. The program will be in one of the $m+1$ states denoted by $0, 1, \dots, m$, so that in state 0 the program is executing on the CPU, and in state i ($1 \leq i \leq m$) the program is performing an I/O operation on device i . Assume that the request for device i occurs at the end of a CPU burst with probability q_i , independent of the past history of the program. The program will finish execution at the end of a CPU burst with probability q_0 so that $\sum_{i=0}^m q_i = 1$. We assume that the system is saturated so that on completion of one program, another statistically identical program will enter the system instantaneously. With these assumptions, the system

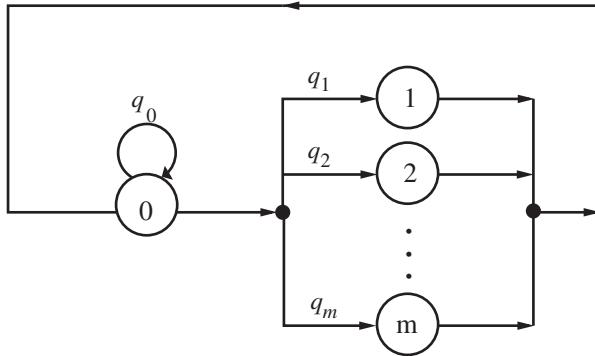


Figure 7.7. A discrete-time Markov model of a program

can be modeled as a discrete-time Markov chain with the state diagram shown in Figure 7.7.

The transition probability matrix P of the Markov chain is given by

$$P = \begin{bmatrix} q_0 & q_1 & \cdot & \cdot & \cdot & q_m \\ 1 & 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & \vdots & & & \vdots & \\ 1 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix}.$$

If we assume that $0 < q_i < 1 (i = 0, 1, \dots, m)$, then it is easy to verify that this finite Markov chain is both irreducible and aperiodic. Therefore, Theorem 7.4 applies. The unique steady-state probability vector, \mathbf{v} , is obtained by solving the system of linear equations:

$$\mathbf{v} = \mathbf{v}P$$

or

$$\begin{aligned} v_0 &= v_0 q_0 + \sum_{j=1}^m v_j, \\ v_j &= v_0 q_j, \quad j = 1, 2, \dots, m. \end{aligned}$$

Using the normalization condition

$$\sum_{j=0}^m v_j = 1,$$

we have

$$v_0 + v_0 \sum_{j=1}^m q_j = 1.$$

Noting that $\sum_{j=1}^m q_j = 1 - q_0$, we get

$$v_0(1 + 1 - q_0) = 1$$

or

$$v_0 = \frac{1}{2 - q_0},$$

and

$$v_j = \frac{q_j}{2 - q_0}, \quad j = 1, 2, \dots, m.$$

The interpretation is that in a real-time interval T , the average number of visits to device j will be $v_j T$ in the long run.

#

Problems

- Consider a system with two components [ASH 1970]. We observe the state of the system every hour. A given component operating at time n has probability p of failing before the next observation at time $n + 1$. A component that was in a failed condition at time n has a probability r of being repaired by time $n + 1$, independent of how long the component has been in a failed state. The component failures and repairs are mutually independent events. Let X_n be the number of components in operation at time n . $\{X_n \mid n = 0, 1, \dots\}$ is a discrete-time homogeneous Markov chain with the state space $I = \{0, 1, 2\}$. Determine its transition probability matrix P , and draw the state diagram. Obtain the steady-state probability vector, if it exists.
- Assume that a computer system is in one of three states: busy, idle, or undergoing repair, respectively denoted by states 0, 1, and 2. Observing its state at 2 P.M. each day, we believe that the system approximately behaves like a homogeneous Markov chain with the transition probability matrix:

$$P = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.1 & 0.8 & 0.1 \\ 0.6 & 0.0 & 0.4 \end{bmatrix}.$$

Prove that the chain is irreducible, and determine the steady-state probabilities.

- Any transition probability matrix P is a stochastic matrix; that is, $p_{ij} \geq 0$ for all i and all j , and $\sum_j p_{ij} = 1$, for all i . If, in addition, the column sums are also unity—that is:

$$\sum_i p_{ij} = 1, \quad \text{for all } j,$$

then matrix P is called *doubly stochastic*. If a Markov chain with doubly stochastic P is irreducible, aperiodic, and finite with n states, show that the steady-state probability is given by

$$v_j = \frac{1}{n}, \quad \text{for all } j.$$

- Show that the Markov chain of Example 7.12 is irreducible and aperiodic if $0 < q_i < 1$ for all i . Also show that if for some j , $1 \leq j \leq m$, $q_j = 0$, then the chain is not irreducible. Finally show that if $q_j = 1$ for some j , $1 \leq j \leq m$, then the chain is periodic.

7.4 DISTRIBUTION OF TIMES BETWEEN STATE CHANGES

We have noted that the entire past history of the homogeneous Markov chain is summarized in its current state. Assume that the state at the n th step is $X_n = i$. But then the probability that the next state is j ; that is, $X_{n+1} = j$, should depend only on the current state i and not on the time the chain has spent in the current state. Let the random variable T_i denote the time the Markov chain spends in state i during a single visit to state i . (In other words, T_i is one plus the number of transitions $i \rightarrow i$ made before leaving state i .) It follows that the distribution of T_i should be memoryless for $\{X_n \mid n = 0, 1, \dots\}$ to form a (homogeneous) Markov chain.

Given that the chain has just entered state i at the n th step, it will remain in this state at the next step with probability p_{ii} and it will leave the state at the next step with probability $\sum_{j \neq i} p_{ij} = 1 - p_{ii}$. Now if the next state is also i (i.e., $X_{n+1} = i$), then the same two choices will be available at the next step. Furthermore, the probabilities of events at the $(n + 1)$ st step are independent of the events at the n th step, because $\{X_n\}$ is a homogeneous Markov chain.

Thus, we have a sequence of Bernoulli trials with the probability of success $1 - p_{ii}$, where success is defined to be the event that the chain leaves state i . The event $T_i = n$ corresponds to n trials up to and including the first success. Hence, T_i has the geometric pmf, so that

$$P(T_i = n) = (1 - p_{ii})^{n-1}, \quad i \in I. \quad (7.22)$$

Using the properties of the geometric distribution, the expected number of steps the chain spends in state i , per visit to state i , is given by

$$E[T_i] = \frac{1}{1 - p_{ii}}, \quad i \in I, \quad (7.23)$$

and the corresponding variance is

$$\text{Var}[T_i] = \frac{p_{ii}}{(1 - p_{ii})^2}, \quad i \in I. \quad (7.24)$$

If we define an “event” to be a change of state, then the successive interevent times of a discrete-time Markov chain are independent, geometrically distributed random variables. Unlike the special case of the Bernoulli process, however, the successive interevent times do not, in general, have identical distributions.

Example 7.13

We return to our example of a communication net consisting of a cascade of binary communication channels with the matrix P given by

$$P = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}.$$

Assuming that a 0 was transmitted, that is, $X_0 = 0$, and the number of stages before the first error is $S_0 = T_0 - 1$. The average number of stages over which a 0 can be transmitted without an error is given by

$$E[S_0] = E[T_0] - 1 = \frac{1-a}{a},$$

and the average number of stages over which a 1 can be transmitted without an error is given by

$$\frac{1-b}{b}.$$

Note that T_0 has a geometric distribution with parameter a , while T_1 has a geometric distribution with parameter b . These two interevent times, although possessing the memoryless distribution, have different parameters associated with them.

#

Example 7.14

Consider the software reliability model from Example 7.3 [GOSE 2000]. During the testing phase, software is subjected to a sequence of runs, making no changes if there is no failure. When a failure occurs on any run, an attempt will be made to fix the underlying fault that will cause the probabilities of success and failure on the next run to change. In other words, the transition probability matrix P_i given by

$$P_i = \begin{bmatrix} p_i & 1-p_i \\ 1-q_i & q_i \end{bmatrix}$$

defines the values of conditional probabilities p_i and q_i for the testing runs that follow the occurrence of the i th failure up to the occurrence of the next $(i+1)$ st failure. Thus, the software reliability growth model in discrete time can be described with a sequence of dependent Bernoulli trials with state-dependent probabilities. The underlying stochastic process is a nonhomogeneous discrete-time Markov chain.

The sequence $\{Y_m\}$, defined in Example 7.3, provides an alternative description of the reliability growth model, as presented in Figure 7.8. Both states i and i_s represent that failure state has been occupied i times. The state i represents the first trial for which the accumulated number of failures Y_{m+1} equals i , while i_s represents all subsequent trials for which $Y_{m+1} = i$, that is, all subsequent successful runs before the occurrence of next $(i+1)$ st failure. Without loss of generality, it is assumed that the first run is successful, that is, 0 is the initial state.

Focusing on the occurrence of failures, it is of particular interest to derive the pmf of the discrete random variable N_{i+1} defined as the number of runs between two successive visits to the failure state of the discrete-time Markov chain, that is, the number of runs between the i th and $(i+1)$ st failures. Clearly (see Figure 7.8), the random variable N_1 has the pmf

$$P(N_1 = k) = f_{01}(k) = p_0^{k-1}(1-p_0), \quad k \geq 1,$$

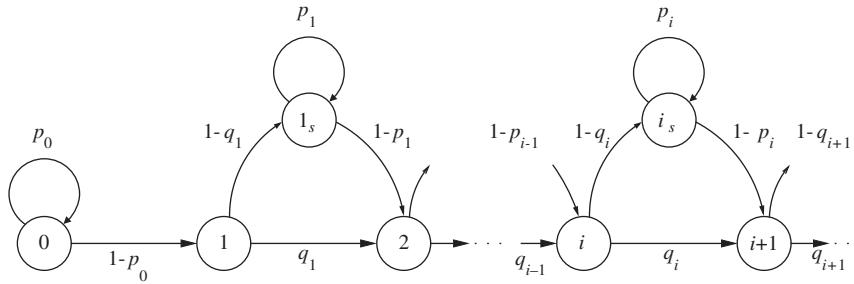


Figure 7.8. Non-homogeneous DTMC model for software reliability growth

and the remaining N_{i+1} ($i \geq 1$) have the pmfs

$$P(N_{i+1} = k) = f_{11}(k) = \begin{cases} q_i & \text{if } k = 1 \\ (1 - q_i)p_i^{k-2}(1 - p_i) & \text{if } k \geq 2. \end{cases}$$

Note that $f_{ij}(k)$ is the conditional probability that the first visit to state j from state i in exactly k steps.

#

7.5 MARKOV MODULATED BERNOUlli PROCESS

The Markov modulated Bernoulli process (MMBP) is a generalization of the Bernoulli process where the parameter of the Bernoulli process varies according to a homogeneous DTMC. The Bernoulli process and the DTMC, which controls (we call *modulates* here) the parameter of the Bernoulli process, are assumed to be independent [ONVU 1993].

The MMBP is used extensively to model traffic in ATM networks where the time is discretized into fixed-length slots, the packets are segmented into fixed-length cells, and each cell is transmitted within a slot. For the MMBP traffic model, the probability that a slot contains a cell is a Bernoulli process with a parameter modulated by an r -state DTMC. At the end of each slot, the DTMC moves from state i to state j with probability p_{ij} , or stays at state i with probability p_{ii} . While in state i , a cell arrives with probability c_i and no cell arrives with probability $1 - c_i$. Then the MMBP is characterized by the transition probability matrix P of the DTMC and the diagonal matrix C of cell arrival probabilities:

$$C = \begin{bmatrix} c_1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & c_2 & 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & \vdots & \vdots & \cdot & \cdot & \cdot & \vdots \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & c_r \end{bmatrix}.$$

Consider the interrupted Bernoulli process (IBP), the simplest case of MMBP, with the transition probability matrix P and the diagonal matrix C of arrival probabilities given by

$$P = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix},$$

$$C = \begin{bmatrix} c & 0 \\ 0 & 0 \end{bmatrix}.$$

If the DTMC is in state 1, it will remain in that state in the next slot with probability $1-a$ or change state with probability a . The cell arrival probability in state 1 is c . When the DTMC is in state 2, it will remain in that state with probability $1-b$ or change state with probability b . No cell arrivals will occur while the DTMC is in state 2. The steady-state probability that the IBP is in state i , v_i was obtained in Example 7.10:

$$v_1 = \frac{b}{a+b} \quad v_2 = \frac{a}{a+b}.$$

By assigning reward $r_1 = c$ and $r_2 = 0$, the average cell arrival probability is the expected steady-state reward

$$E[Z] = v_1 c = \frac{bc}{a+b}.$$

Next we study the cell interarrival time distribution of IBP. Let T_1 be the time interval to the next arrival, given that the DTMC is in state 1, and T_2 be the time interval to the next arrival, given that the DTMC is in state 2.

Consider the time an arrival occurs when the DTMC is in state 1. In the next slot

- IBP remains in state 1 and an arrival occurs, which happens with probability $(1-a)c$.
- IBP remains in state 1 and no arrival occurs, which happens with probability $(1-a)(1-c)$.
- IBP moves to state 2 and no arrival occurs, which happens with probability a .

Hence we have

$$T_1 = \begin{cases} 1 & \text{with probability } (1-a)c \\ 1+T_1 & \text{with probability } (1-a)(1-c) \\ 1+T_2 & \text{with probability } a. \end{cases} \quad (7.25)$$

Consider the time when the DTMC is in state 2; then, in the next slot

- IBP remains in state 2, which happens with probability $(1-b)$.

- IBP moves to state 1 and no arrival occurs, which happens with probability $b(1 - c)$.
- IBP moves to state 1 and an arrival occurs, which happens with probability bc .

Hence we have

$$T_2 = \begin{cases} 1 + T_2 & \text{with probability } (1 - b) \\ 1 + T_1 & \text{with probability } b(1 - c) \\ 1 & \text{with probability } bc. \end{cases} \quad (7.26)$$

From the preceding coupled recurrence equations we now obtain the probability generating functions (PGF) for T_1 and T_2 . First note that the PGF of constant 1 is z . Now, noting the convolution property of PGF, the recurrence equations give us

$$G_{T_1}(z) = (1 - a)c z + (1 - a)(1 - c)z G_{T_1}(z) + a z G_{T_2}(z) \quad (7.27)$$

and

$$G_{T_2}(z) = bcz + (1 - b)z G_{T_2}(z) + b(1 - c)z G_{T_1}(z). \quad (7.28)$$

After some manipulation, the probability generating functions of T_1 and T_2 can be shown to be

$$G_{T_1}(z) = \frac{(a + b - 1)cz^2 + (1 - a)cz}{(1 - a - b)(1 - c)z^2 - (2 - a - b - c + ac)z + 1} \quad (7.29)$$

$$G_{T_2}(z) = \frac{bcz}{(1 - a - b)(1 - c)z^2 - (2 - a - b - c + ac)z + 1}. \quad (7.30)$$

Since when an arrival occurs the IBP must be in state 1, it follows that the interarrival interval, T , is equal to T_1 . Then the probability generating function of the interarrival time of the IBP is

$$G_T(z) = \frac{(a + b - 1)cz^2 + (1 - a)cz}{(1 - a - b)(1 - c)z^2 - (2 - a - b - c + ac)z + 1}. \quad (7.31)$$

Using the moment generating property of the PGF, we obtain

$$E[T] = \frac{a + b}{bc}. \quad (7.32)$$

If we assume $a = 0$ and $b = 1$, then

$$G_T(z) = \frac{cz}{1 - (1 - c)z},$$

in this case the IBP becomes a Bernoulli process.

If we assume $c = 1$ and $a + b = 1$, then

$$G_T(z) = \frac{(1-a)z}{1-az},$$

the IBP once again becomes a Bernoulli process.

Problems

1. Starting with equations (7.27) and (7.28), show that equation (7.29) holds.
2. Starting with equation (7.31), derive equation (7.32).

7.6 IRREDUCIBLE FINITE CHAINS WITH APERIODIC STATES

In this section we consider some examples of finite Markov chains that satisfy the conditions of Theorem 7.4 so that the unique steady-state probabilities can be obtained by solving the system of linear equations (7.18) and (7.19).

7.6.1 Memory Interference in Multiprocessor Systems

Consider the shared memory multiprocessor system shown in Figure 7.9. The processors' ability to share the entire memory space provides a convenient means of sharing information and provides flexibility in memory allocation. The price of sharing is the contention for the shared resource. To reduce contention, the memory is usually split up into modules, which can be accessed

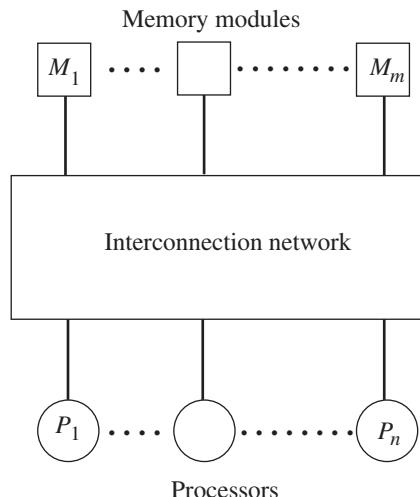


Figure 7.9. A multiprocessor system with multimodule memory

independently and concurrently with other modules. When more than one processor attempts to access the same module, only one processor can be granted access, while other processors must await their turn in a queue. The effect of such contention, or interference, is to increase the average memory access time.

Assume that the time to complete a memory access is a constant and that all modules are synchronized. Processors are assumed to be fast enough to generate a new request as soon as their current request is satisfied. A processor cannot generate a new request when it is waiting for the current request to be completed. The operation of the system can be visualized as a discrete-time queuing network as shown in Figure 7.10.

The memory modules are the servers, and the fixed number, n , of processors constitute the “jobs” or “customers” circulating in this closed queuing network. The symbol q_i denotes the probability that a processor generated request is directed at memory module i , $i = 1, 2, \dots, m$. Thus $\sum_{i=1}^m q_i = 1$.

As an example, consider a system with two memory modules and two processors. Let the number of processors waiting or being served at module i ($i = 1, 2$) be denoted by N_i . Clearly, $N_i \geq 0$ and $N_1 + N_2 = 2$. The pair (N_1, N_2) denotes the state of the system, and the state space $I = \{(1, 1), (0, 2), (2, 0)\}$. The operation of the system is described by a discrete-time Markov chain whose state diagram is shown in Figure 7.11.

The transition probability matrix of this chain is given by

$$P = \begin{pmatrix} (1, 1) & (0, 2) & (2, 0) \\ (1, 1) & \begin{bmatrix} 2q_1 q_2 & q_2^2 & q_1^2 \\ q_1 & q_2 & 0 \\ q_2 & 0 & q_1 \end{bmatrix} \\ (0, 2) & & \\ (2, 0) & & \end{pmatrix}.$$

Number of "customers" = number of processors, n

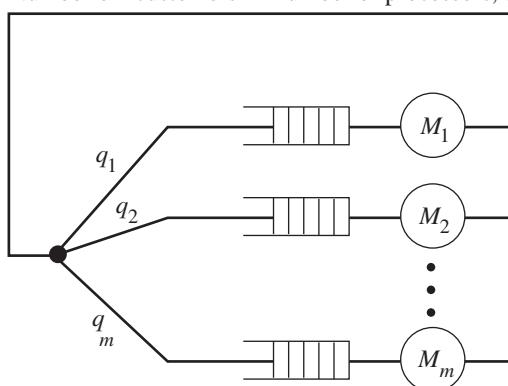


Figure 7.10. A discrete-time queuing network representation of multiprocessor memory interference

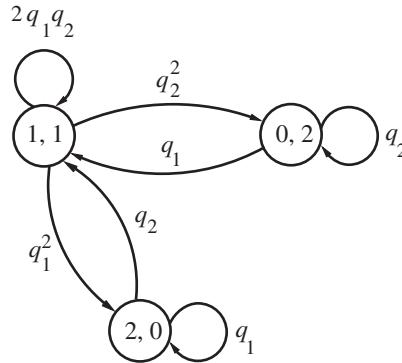


Figure 7.11. The state diagram for an example of the memory interference problem

We will explain the elements in the top row of this matrix; the remaining entries can be explained in a similar way. Assume that the system is in state $(1,1)$ at time k , hence both memory modules and both processors are busy. At the end of this period both processors will be independently generating their new requests. Generation of a new request by a processor may be thought of as a Bernoulli trial with probability q_i of accessing module i . Thus we have a sequence of two Bernoulli trials. The probability that both processors will simultaneously request memory module i is q_i^2 . If $i = 1$, the state of the system at time $k + 1$ will be $(2,0)$ while if $i = 2$, the new state will be $(0,2)$. If the two processors request access to distinct memory modules, the next state will be $(1,1)$. The probability of this last event is easily seen to be $2q_1q_2$.

To obtain the steady-state probability vector $\mathbf{v} = [v_{(1,1)}, v_{(0,2)}, v_{(2,0)}]$ we use

$$\mathbf{v} = \mathbf{v}P \quad \text{and} \quad \sum_{(i,j) \in I} v_{(i,j)} = 1$$

or

$$\begin{aligned} v_{(1,1)} &= 2q_1q_2v_{(1,1)} + q_1v_{(0,2)} + q_2v_{(2,0)}, \\ v_{(0,2)} &= q_2^2v_{(1,1)} + q_2v_{(0,2)}, \\ v_{(2,0)} &= q_1^2v_{(1,1)} + q_1v_{(2,0)}, \\ v_{(1,1)} + v_{(0,2)} + v_{(2,0)} &= 1. \end{aligned}$$

Thus

$$v_{(2,0)} = \frac{q_1^2}{1 - q_1}v_{(1,1)}, \quad v_{(0,2)} = \frac{q_2^2}{1 - q_2}v_{(1,1)},$$

which implies that

$$v_{(1,1)} = \frac{1}{1 + \frac{q_1^2}{1-q_1} + \frac{q_2^2}{1-q_2}} = \frac{q_1q_2}{1 - 2q_1q_2}.$$

Let the random variable B denote the number of memory requests completed per memory cycle in the steady state. We are interested in computing the average number, $E[B]$, of memory requests completed per memory cycle. Note that in state (1,1) two requests are completed, while in states (2,0) or (0,2) only one request each is completed. Therefore, the conditional expectations of B are given by

$$E[B \mid \text{system in state}(1, 1)] = 2,$$

$$E[B \mid \text{system in state}(2, 0)] = 1,$$

$$E[B \mid \text{system in state}(0, 2)] = 1.$$

We assign rewards to the three states of the DTMC as follows: $r_{(1,1)} = 2$, $r_{(2,0)} = 1$, and $r_{(0,2)} = 1$. Then the expected steady-state reward is

$$\begin{aligned} E[Z] &= E[B] = 2v_{(1,1)} + v_{(0,2)} + v_{(2,0)} \\ &= \left(2 + \frac{q_1^2}{1-q_1} + \frac{q_2^2}{1-q_2} \right) v_{(1,1)} \\ &= \frac{1 - q_1 q_2}{1 - 2q_1 q_2}. \end{aligned}$$

The quantity $E[B]$ achieves its maximum value, $\frac{3}{2}$, when $q_1 = q_2 = \frac{1}{2}$. This is considerably smaller than the capacity of the memory system, which is two requests per cycle. For a deeper study of this problem, see Ajimone-Marsan et al. [AJMO 1986].

Problems

- For the example of multiprocessor memory interference with two processors and two memory modules, explicitly solve the following optimization problem:

$$\begin{aligned} \max : & E[B] \\ \text{s.t. :} & q_1 + q_2 = 1, \\ & q_1, q_2 \geq 0. \end{aligned}$$

- Modify the multiprocessor memory interference example so that processor 1 has associated probabilities r_1 and r_2 respectively, for accessing module 1 and 2, and processor 2 has distinct probabilities q_1 and q_2 associated with it. Construct the Markov chain state diagram, solve for the steady-state probabilities, and compute $E[B]$. For those with extra energy, solve an optimization problem analogous to problem 1 above.
- Consider another modification to the memory interference example where the processor requires nonzero amount of time to generate a memory request. Simplify the problem by assuming that the processor cycle time is identical to the memory cycle time. Once again go through all the steps as in problem 2 above and compute $E[B]$.

7.6.2 Models of Program Memory Referencing Behavior

In Chapter 6 we considered the renewal model of page referencing behavior of programs. In the renewal model, the successive intervals between references to a given page were assumed to be independent identically distributed random variables. The first model we consider in this section is a special case of the renewal model, where the above intervals are geometrically distributed. This is known as the *independent reference model* (IRM) of program behavior. Although simple to analyze, such a model is not very realistic. The LRU (least recently used) stack model, which is a better approximation to the behavior of real programs, is considered next. In-depth treatment of such models are available in [SPIR 1977].

7.6.2.1 The Independent Reference Model. A program's address space typically consists of continuous pages represented by the indices $1, 2, \dots, n$. For the purpose of studying a program's reference behavior, it can be represented by the reference string $w = x_1, x_2, \dots, x_t, \dots$. Successive references are assumed to form a sequence of independent, identically distributed random variables with the pmf

$$P(X_t = i) = \beta_i, \quad 1 \leq i \leq n; \quad \sum_{i=1}^n \beta_i = 1,$$

It is clear then that the interval between two successive references to page i is geometrically distributed with parameter β_i . Using the theory of finite, irreducible, and aperiodic Markov chains developed earlier, we can analyze the performance of several paging algorithms, assuming the independent reference model of program behavior.

We assume that a fixed number, $m (1 \leq m \leq n)$, of page frames have been allocated to the program. The internal state of the paging algorithm at time t , denoted by $\mathbf{q}(t)$, is an ordered list of the m pages currently in main memory. If the next page referenced (x_{t+1}) is not in main memory, then a *page fault* is said to have occurred, and the required page will be brought from secondary storage into main memory. This will, in general, require the replacement of an existing page from main memory. We will assume that the rightmost page in the ordered list $\mathbf{q}(t)$ will be replaced. On the other hand, if the next page referenced (x_{t+1}) is in main memory, no page fault (and replacement) occurs, but the list $\mathbf{q}(t)$ is updated to $\mathbf{q}(t+1)$, reflecting the new replacement priorities. It is clear that the sequence of states $\mathbf{q}(0), \mathbf{q}(1), \dots, \mathbf{q}(t), \dots$ forms a discrete-time homogeneous Markov chain with the state space consisting of $n!/(n-m)!$ permutations over $\{1, 2, \dots, n\}$. It is assumed that the main memory is preloaded initially with m pages. Since we will be studying the steady-state behavior of the Markov chain, the initial state has no effect on our results.

As an example, consider the LRU paging algorithm with $n = 3$ and $m = 2$. It is logical to let $\mathbf{q}(t)$ be ordered by the recency of usage, so that $\mathbf{q}(t) = (i, j)$ implies that the page indexed i was more recently used than page j , and, therefore, page j will be the candidate for replacement. The state space I is given by

$$I = \{(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2)\}.$$

Let the current state $\mathbf{q}(t) = (i, j)$. Then the next state $\mathbf{q}(t+1)$ takes one of the following values:

$$\mathbf{q}(t+1) = \begin{cases} (i, j), & \text{if } x_{t+1} = i, \text{ with associated probability } \beta_i, \\ (j, i), & \text{if } x_{t+1} = j, \text{ with associated probability } \beta_j, \\ (k, i), & \text{if } x_{t+1} = k, k \neq i, k \neq j, \text{ with associated probability } \beta_k. \end{cases}$$

Then the transition probability matrix P is given by

$$P = \begin{pmatrix} (1, 2) & (2, 1) & (1, 3) & (3, 1) & (2, 3) & (3, 2) \\ (1, 2) & \left[\begin{matrix} \beta_1 & \beta_2 & 0 & \beta_3 & 0 & 0 \\ \beta_1 & \beta_2 & 0 & 0 & 0 & \beta_3 \\ 0 & \beta_2 & \beta_1 & \beta_3 & 0 & 0 \\ 0 & 0 & \beta_1 & \beta_3 & \beta_2 & 0 \\ \beta_1 & 0 & 0 & 0 & \beta_2 & \beta_3 \\ 0 & 0 & \beta_1 & 0 & \beta_2 & \beta_3 \end{matrix} \right] \\ (2, 1) & \\ (1, 3) & \\ (3, 1) & \\ (2, 3) & \\ (3, 2) & \end{pmatrix}.$$

It can be verified that the above Markov chain is irreducible and aperiodic; hence a unique steady-state probability vector \mathbf{v} exists. This vector is obtained by solving the system of equations:

$$\mathbf{v} = \mathbf{v}P$$

and

$$\sum_{(i,j)} v_{(i,j)} = 1.$$

Solving this system of equations, we get (the student is urged to verify this):

$$v_{(i,j)} = \frac{\beta_i \beta_j}{1 - \beta_i}.$$

Note that a page fault occurs in state (i, j) , provided that a page other than i or j is referenced. The associated conditional probability of this event is $1 - \beta_i - \beta_j$; we hence assign reward $r_{(i,j)} = 1 - \beta_i - \beta_j$ to state (i, j) . The steady-state page fault probability is then given by

$$E[Z] = F(\text{LRU}) = \sum_{(i,j) \in I} (1 - \beta_i - \beta_j) \frac{\beta_i \beta_j}{1 - \beta_i}.$$

More generally, for arbitrary values of $n \geq 1$ and $1 \leq m \leq n$, it can be shown (see problem 3 at the end of this section) that

$$F(\text{LRU}) = \sum_{\substack{\text{over the} \\ \text{state space}}} D_1^2(\mathbf{q}) \prod_{i=1}^m \frac{\beta_{j_i}}{D_i(\mathbf{q})},$$

where $\mathbf{q} = (j_1, j_2, \dots, j_m)$ and

$$D_i(\mathbf{q}) = 1 - \sum_{k=1}^{m-i+1} \beta_{j_k}.$$

Similar results can be derived for several other paging algorithms (see problems 15 and 16 at the end of this section).

7.6.2.2 Performance Analysis of Cache Memories. Using the independent reference model, we can analyze the performance of different cache organizations: fully associative, direct mapped and set associative, as shown in Figure 7.12 [RAO 1978].

We assume that both the cache and the main memory are divided into equal-sized units called *blocks*. In fully associative cache, any block in the main memory can be mapped to any block of cache. In direct mapped cache, block i can be mapped to the cache block $(i \bmod m)$ only if we have m blocks in the cache. In set associate cache, the cache is divided into L sets with $s = m/L$ blocks per set. A block i in main memory can be in any block belonging to the set $(i \bmod L)$.

To calculate the cache miss ratio, let the blocks in the main memory be labeled $1, 2, \dots, n$. Let the memory block reference string be denoted by $x_1, x_2, \dots, x_t, \dots$. Let $[\beta_1, \beta_2, \dots, \beta_n]$ be the pmf of the block reference probabilities:

$$P(X_t = i) = \beta_i, \quad 1 \leq i \leq n, \quad \text{for all } t > 0.$$

The contents of the cache characterize the state of the cache at any time. Letting $\mathbf{q}(t)$ denote the t th state, the state sequence of $\mathbf{q}(t)$ forms a discrete-time homogeneous Markov chain. For fully associative caches, the IRM model discussed above can be also applied to the cache analysis.

We now consider the case of direct mapped caches. Let G_i denote the set of blocks in the main memory that can be in block frame i of the cache. Let $k = n/m$ be the cardinality of each of these sets. We refer to the pages in G_i by $1(i), \dots, k(i)$ with probabilities of reference $\beta_1(i), \dots, \beta_k(i)$.

Let $D_i = \sum_{j=1}^k \beta_j(i)$, $i = 1, 2, \dots, m$. Consider the two-state DTMC in Figure 7.13 for a particular cache block, where the state 1 is the state with block $j(i)$ in the cache, while 0 is the state with that block missing. When in state 1, a reference to any other block produces a miss, causing removal

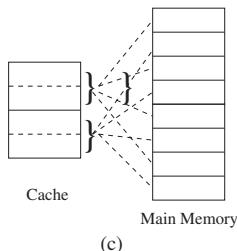
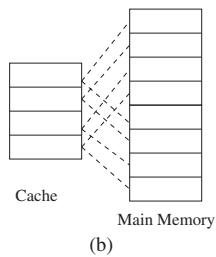
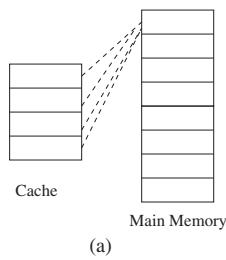


Figure 7.12. Different cache organizations: (a) fully associative; (b) direct-mapped; (c) set-associative

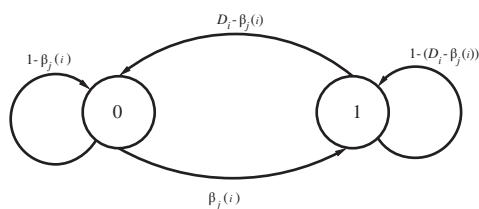


Figure 7.13. Markov chain for the directly mapped cache

of that block. Thus the transition probability from 1 to 0 is $D_i - \beta_j(i)$. Similarly, the transition probability from 0 to 1 is $\beta_j(i)$.

Using the conclusion from Example 7.10, we derive the solution of the steady-state probability of state 0 of this chain, which is the probability that block $j(i)$ is absent from the cache in steady state:

$$\begin{aligned} m_j(i) &= \frac{D_i - \beta_j(i)}{D_i - \beta_j(i) + \beta_j(i)} \\ &= 1 - \frac{\beta_j(i)}{D_i}. \end{aligned} \quad (7.33)$$

Since direct-mapped cache does not have replacement rule, we have

$$F(DM) = \sum_{i=1}^m \sum_{j=1}^k \beta_j(i) m_j(i) \quad (7.34)$$

where $m_j(i) = P(j(i)$ is not in the cache in steady state). Substituting for $m_j(i)$, we finally obtain

$$F(DM) = \sum_{i=1}^m \frac{(D_i^2 - \sum_{j=1}^k \beta_j^2(i))}{D_i} \quad (7.35)$$

as the expression for the limiting cache miss ratio for the direct-mapped cache.

Similarly, we can also calculate the miss ratio of set-associative caches.

7.6.2.3 The LRU Stack Model [Spir 1977]. Intuitively, we expect the probability of referencing a given page i at time t to depend on the pages referenced in the immediate past. Thus the independent reference model may be expected to be a poor model of practical reference strings. It has been observed that references to pages tend to cluster together, so that the probability of referencing a page is high for a more recently used page. The LRU stack model is able to reflect such a behavior of reference strings. Validation experiments have confirmed that this model fits real reference string behavior much better than does the IRM.

In the LRU stack model, we associate a sequence of LRU stacks $s_0 s_1 \cdots s_t$ with a reference string $w = x_1 x_2 \cdots x_t \cdots$. The stack s_t is the n -tuple (j_1, \dots, j_n) in which j_i is the i th most recently referenced page at time t . Let D_t be the position of the page x_t , in the stack s_{t-1} . Then, associated with the reference string, we have the distance string $D_1 D_2 \cdots D_t \cdots$.

The LRU stack model assumes that the distance string is a sequence of independent identically distributed random variables with the following pmf:

$$P(D_t = i) = a_i, \quad i = 1, 2, \dots, n, \quad t \geq 1, \quad \text{and} \quad \sum_{j=1}^n a_j = 1.$$

The distribution function is then

$$P(D_t \leq i) = A_i = \sum_{j=1}^i a_j, \quad i = 1, 2, \dots, n, \quad t \geq 1.$$

Without loss of generality, we assume the initial stack

$$\mathbf{s}_0 = (1, 2, \dots, n).$$

Note that IRM assumes that the reference string is a discrete independent process whereas the LRU stack model assumes that the distance string is a discrete independent process, and the corresponding reference string, a nonindependent stochastic process.

With this model, evaluation of the page-fault rate of the LRU paging algorithm is quite simple. Assume that the program has been allocated m page frames of main memory. Then a page fault will occur at time t provided $D_t > m$. Thus, the page fault probability is given by

$$F(\text{LRU}) = P(D_t > m) = 1 - P(D_t \leq m) = 1 - A_m.$$

Let us study the movement of a tagged page (say, y) through the LRU stack as time progresses. Define the random sequence $E_0 E_1 E_2 \cdots E_n \cdots$ such that $E_t = i$ if page y occupies the i th position in stack \mathbf{s}_t . Clearly, $1 \leq E_t \leq n$ for all $t \geq 1$. Thus the sequence described is a discrete-time, discrete-state stochastic process. By the stack updating procedure shown in Figure 7.14, the position of the page y in stack \mathbf{s}_{t+1} is determined by the next reference x_{t+1} and the position of page y in stack \mathbf{s}_t , but not its position in previous

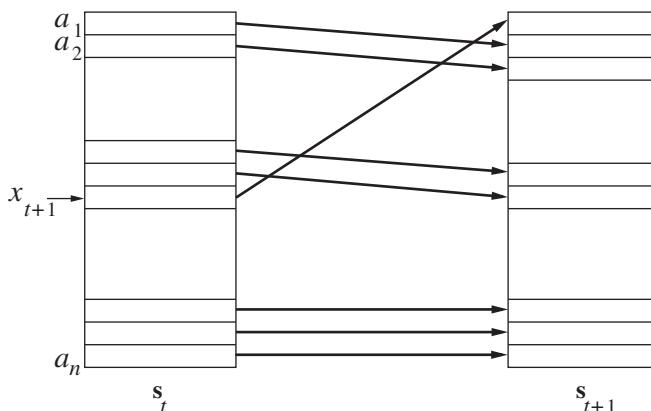


Figure 7.14. LRU stack updating procedure

stacks. Thus the sequence above is a discrete-time Markov chain. Furthermore, the chain is homogeneous.

We obtain the transition probabilities of the chain by observing the stack updating procedure shown in Figure 7.14.

Then

$$\begin{aligned} p_{i1} &= P(E_{t+1} = 1 | E_t = i) \\ &= P(x_{t+1} = y) = P(D_{t+1} = i) = a_i, \quad 1 \leq i \leq n, \\ p_{ii} &= P(E_{t+1} = i | E_t = i) \\ &= P(D_{t+1} < i) = A_{i-1}, \quad 2 \leq i \leq n, \\ p_{i,i+1} &= P(E_{t+1} = i+1 | E_t = i) \\ &= P(D_{t+1} > i) = 1 - A_i, \quad 1 \leq i \leq n-1, \end{aligned}$$

and

$$p_{i,j} = 0, \quad \text{otherwise.}$$

The state diagram is given in Figure 7.15. The transition probability matrix is given by

$$P = \begin{bmatrix} 1 & 2 & \cdots & i & i+1 & \cdots & n \\ \cdot & a_1 & 1-A_1 & 0 & 0 & \cdot & \cdots & 0 \\ \cdot & a_2 & A_1 & 1-A_2 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & 1-A_{i-1} & \cdot & \cdot & \cdot \\ \cdot & a_i & \cdot & \cdot & A_{i-1} & 1-A_i & \cdots & \cdot \\ \vdots & \vdots \\ n-1 & a_{n-1} & 0 & \cdot & 0 & \cdot & 1-A_{n-1} & \cdot \\ n & a_n & 0 & \cdot & 0 & \cdot & \cdot & A_{n-1} \end{bmatrix}.$$

Clearly, the chain is aperiodic and irreducible if we assume that $a_i > 0$ for all i . Then the steady-state probability vector $\mathbf{v} = [v_1, v_2, \dots, v_n]$ is obtained from the following system of equations:

$$v_1 = \sum_{i=1}^n v_i a_i, \tag{7.36}$$

$$v_i = v_{i-1}(1 - A_{i-1}) + v_i A_{i-1}, \quad 2 \leq i \leq n, \tag{7.37}$$

$$\sum_{i=1}^n v_i = 1. \tag{7.38}$$

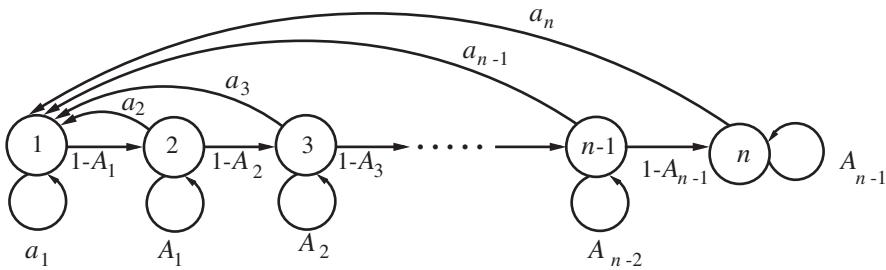


Figure 7.15. The state diagram for the LRU-stack example

From equation (7.37), we have $v_i = v_{i-1} = v_2, 2 \leq i \leq n$; from equation (7.36), we have

$$v_1 = v_1 a_1 + v_2 \sum_{i=2}^n a_i = v_1 a_1 + v_2 (1 - a_1)$$

and $v_1 = v_2$. Then from equation (7.38) we conclude

$$v_i = \frac{1}{n}, \quad i = 1, 2, \dots, n.$$

Thus, the position of the tagged page, in the steady state, is independent of its initial position and it is equally likely to be in any stack position. This implies that each page is equally likely to be referenced in the long run. Therefore, the LRU stack model is not able to cater to the nonuniform page referencing behavior of real programs, although it does reflect the clustering effect. This may be due to the assumption that the distance string is a sequence of independent identically distributed random variables. A logical generalization is to let the distance string be a Markov-dependent sequence, as discussed by Shedler and Tung [SHED 1972].

Problems

1. * Using the independent reference model of program behavior, show that the steady-state page-fault rate of the FIFO (first in, first out) paging algorithm is given by

$$F(\text{FIFO}) = G^{-1} \sum_{\mathbf{q}} D_1(\mathbf{q}) \prod_{i=1}^m \beta_{j_i},$$

where $\mathbf{q} = (j_1, j_2, \dots, j_m)$ and

$$G = \sum_{\mathbf{q}} \prod_{i=1}^m \beta_{j_i}.$$

2. Consider a Markov dependent reference string so that

$$\begin{aligned} P(x_t = i \mid x_{t-1} = j) &= q_{ij}, & 1 \leq i, j \leq n, t > 1, \\ P(x_1 = i) &= \beta_i, & 1 \leq i \leq n. \end{aligned}$$

Study the steady-state behavior of the page replacement algorithm that selects the page in memory with the smallest probability of being referenced at time $t + 1$ conditioned on x_t . As a special case, consider:

$$\begin{aligned} n &= 3, \quad m = 2, \quad q_{11} = 0, \quad q_{12} = \epsilon, \quad q_{13} = 1 - \epsilon, \\ q_{21} &= \frac{1}{2} - \delta, & q_{22} &= 0, & q_{23} &= \frac{1}{2} + \delta, \\ q_{31} &= 0, & q_{32} &= 1, & q_{33} &= 0. \end{aligned}$$

Describe the states and state transitions of the paging algorithm, compute steady-state probabilities and steady-state average page-fault rate.

3. * Generalize the result derived for the steady-state page-fault probability of the LRU paging algorithm to the case $n \geq 1$ and $1 \leq m \leq n$.

7.6.3 Slotted Aloha Model

The Aloha network was developed to provide radio-based data communication on the University of Hawaii campus [ABRA 1970]. The behavior of slotted Aloha can be captured by a DTMC. Consider m users, n of which are currently backlogged. Each of the $m - n$ unbacklogged users is assumed to transmit independently in each slot with probability a , while each backlogged user transmits independently in each slot with probability b . Let the number of backlogged users, n , denote the state of the DTMC. Let $A(i, n)$ denote the probability that i unbacklogged users attempt to transmit in a slot when the DTMC is in state n and let $B(i, n)$ be the corresponding probability for backlogged users. Then

$$\begin{aligned} A(i, n) &= \binom{m-n}{i} (1-a)^{m-n-i} a^i, \quad 0 \leq i \leq m-n, \\ B(i, n) &= \binom{n}{i} (1-b)^{n-i} b^i, \quad 0 \leq i \leq n. \end{aligned}$$

A packet is successfully transmitted in a slot provided either (1) exactly one unbacklogged user and no backlogged user transmits or (2) no unbacklogged user and exactly one backlogged user transmits. In all other cases, all transmitting unbacklogged users are added to the set of backlogged users. Thus, transition probabilities of the DTMC are given by Bertsekas and Gallager [BERT 1992]:

$$p_{n,n+i} = \begin{cases} A(i, n), & 2 \leq i \leq m-n, \\ A(1, n)[1 - B(0, n)], & i = 1, \\ A(1, n)B(0, n) + A(0, n)[1 - B(1, n)], & i = 0, \\ A(0, n)B(1, n), & i = -1. \end{cases} \quad (7.39)$$

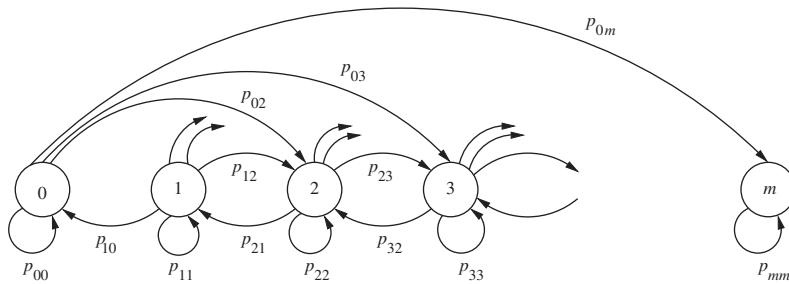


Figure 7.16. The state diagram of the DTMC for slotted Aloha

The state diagram of the DTMC is shown in Figure 7.16.

In state n , a successful transmission occurs with probability $A(1, n)B(0, n) + A(0, n)B(1, n)$. By attaching reward $r_n = A(1, n)B(0, n) + A(0, n)B(1, n)$ to state n , we can compute the expected steady-state reward $E[T]$ (the probabilities of successful transmission in a slot) to be

$$E[T] = \sum_{n=0}^m r_n v_n,$$

where v_n is the steady-state probability of state n .

Problems

- Derive explicit expressions for $E[T]$ for the slotted Aloha system with $m = 1, 2$ and 3 .

7.6.4 Performance Analysis of an ATM Multiplexer

In an ATM (asynchronous transfer mode) network, information is transferred in fixed-size packets called *cells*. The time used to transmit a cell is called a *slot*. Consider an ATM multiplexer, an equipment to aggregate traffic from multiple input links to an output link. There are n input links and 1 output link. There is a buffer at the output port that can accommodate an infinite number of cells (see Figure 7.17). Assume that the cell arrival process at each input is a Bernoulli process with success probability c . The cells arrive at the beginning of each slot. At the end of each slot, one cell is sent out from the buffer.

We define the random variable A as the number of cell arrivals at the buffer during a given slot. It follows that A has the binomial pmf

$$a_i = P[A = i] = \binom{n}{i} c^i (1 - c)^{(n-i)}$$

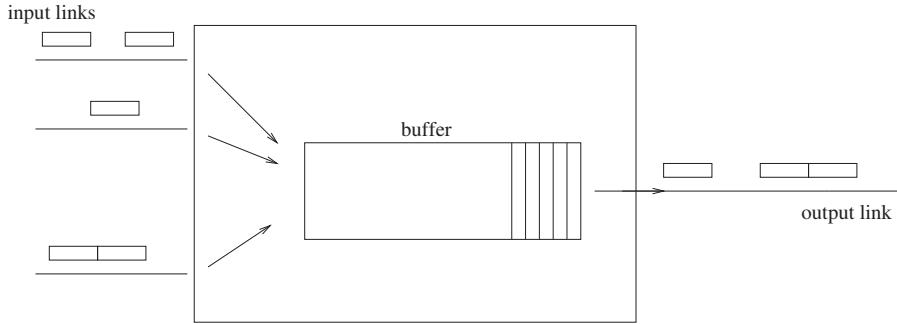


Figure 7.17. an ATM multiplexer

with probability generating function

$$G_A(z) = \sum_{i=0}^n a_i z^i = (1 - c + cz)^n. \quad (7.40)$$

Letting N_m denote the number of cells in the buffer at the end of the m th slot, and A_m denote the number of cells arriving during the m th time slot, we have

$$N_m = \max(0, N_{m-1} + A_m - 1). \quad (7.41)$$

The underlying stochastic process $\{N_m \mid m = 0, 1, 2, \dots\}$ is a DTMC, of which the state diagram is shown in Figure 7.18. If $nc < 1$, the steady-state of the number of cells in the buffer N exists; thus, we have

$$N = \max(0, N + A - 1)$$

with $q_i = P(N = i)$. Its probability generating function is given by

$$\begin{aligned} G_N(z) &= \sum_{k=0}^{\infty} P(N = k)z^k \\ &= \sum_{k=0}^{\infty} q_k z^k \\ &= a_0 q_0 + \sum_{k=1}^{\infty} P(N + A - 1 = k)z^k \\ &= a_0 q_0 + \frac{\sum_{k=1}^{\infty} P(N + A = k)z^k}{z} \\ &= a_0 q_0 + \frac{G_N(z)G_A(z) - a_0 q_0}{z}. \end{aligned}$$

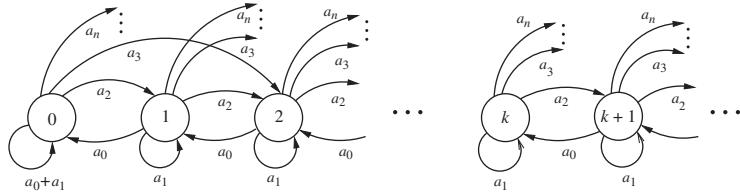


Figure 7.18. DTMC for the queue length of an ATM multiplexer

Then

$$G_N(z) = \frac{a_0 q_0 (1 - z)}{G_A(z) - z}. \quad (7.42)$$

In (7.42), $G_N(1) = 1$ should hold. Applying l'Hôpital's rule, we have

$$1 = \frac{-a_0 q_0}{G'_A(1) - 1} = \frac{-a_0 q_0}{nc - 1}$$

which means

$$a_0 q_0 = 1 - nc.$$

Then (7.42) becomes

$$G_N(z) = \frac{(1 - nc)(1 - z)}{G_A(z) - z}. \quad (7.43)$$

Finally, using (7.40) in (7.43), we obtain

$$G_N(z) = \frac{(1 - nc)(1 - z)}{(1 - c + cz)^n - z}. \quad (7.44)$$

Now, differentiating (7.44) with respect to z and taking the limit as $z \rightarrow 1$, we obtain the mean steady-state queue size

$$E[N] = \frac{n(n-1)c^2}{2(1-nc)} = \frac{(n-1)}{n} \frac{(nc)^2}{2(1-nc)}.$$

7.7 * THE $M/G/1$ QUEUING SYSTEM

We consider a single-server queuing system whose arrival process is Poisson with the average arrival rate λ . The job service times are independent and identically distributed with the distribution function F_B and pdf f_B . Jobs are scheduled for service in their order of arrival; that is, the scheduling discipline is FCFS. As a special case of the $M/G/1$ system, if we let F_B be the exponential distribution with parameter μ , then we obtain the $M/M/1$ queuing system. If the service times are assumed to be a constant, then we get the $M/D/1$ queuing system.

Let $N(t)$ denote the number of jobs in the system (those in the queue plus any in service) at time t . If $N(t) \geq 1$, then a job is in service, and since the general service time distribution need not be memoryless, besides $N(t)$, we also require knowledge of time spent by the job in service in order to predict the future behavior of the system. It follows that the stochastic process $\{N(t) \mid t \geq 0\}$ is *not* a Markov chain.

To simplify the state description, we take a snapshot of the system at times of departure of jobs. These epochs of departure, called **regeneration points**, are used to specify the index set of a new stochastic process. Let $t_n (n = 1, 2, \dots)$ be the time of departure (immediately following service) of the n th job, and let X_n be the number of jobs in the system at time t_n so that

$$X_n = N(t_n), \quad n = 1, 2, \dots \quad (7.45)$$

The stochastic process $\{X_n, n = 1, 2, \dots\}$ will be shown to be a homogeneous, discrete-time Markov chain, known as the **embedded Markov chain** of the continuous-parameter stochastic process $\{N(t) \mid t \geq 0\}$.

The method of the embedded Markov chain allows us to simplify analysis, since it converts a non-Markovian problem into a Markovian one. We can then use the limiting distribution of the embedded Markov chain as a measure of the original process $N(t)$, for it can be shown that the limiting distribution of the number of jobs $N(t)$ observed at an arbitrary point in time is identical to the distribution of the number of jobs observed at the departure epochs:

$$\lim_{t \rightarrow \infty} P(N(t) = k) = \lim_{n \rightarrow \infty} P(X_n = k). \quad (7.46)$$

This property of queuing systems with Poisson arrivals is known as the *PASTA* (Poisson arrivals see time averages) *theorem* [WOLF 1982]. For $n = 1, 2, \dots$, let Y_n be the number of jobs arriving during the service time of the n th job. Now the number of jobs immediately following the departure instant of $(n+1)$ st job can be written as:

$$X_{n+1} = \begin{cases} X_n - 1 + Y_{n+1}, & \text{if } X_n > 0, \\ Y_{n+1}, & \text{if } X_n = 0. \end{cases} \quad (7.47)$$

In other words, the number of jobs immediately following the departure of the $(n+1)$ st job depends on whether the $(n+1)$ st job was in the queue when the n th job departed. If $X_n = 0$, the next job to arrive is the $(n+1)$ st; during its service time Y_{n+1} jobs arrive, then the $(n+1)$ st job departs at time t_{n+1} , leaving Y_{n+1} jobs behind. If $X_n > 0$, then the number of jobs left behind by the $(n+1)$ st job equals $X_n - 1 + Y_{n+1}$. Since Y_{n+1} is independent of X_1, X_2, \dots, X_n , it follows that, given the value of X_n , we need not know the values of X_1, X_2, \dots, X_{n-1} , in order to determine the probabilistic behavior of X_{n+1} . Thus, $\{X_n, n = 1, 2, \dots\}$ is a Markov chain.

The transition probabilities of the Markov chain are obtained using equation (7.47):

$$\begin{aligned} p_{ij} &= P(X_{n+1} = j \mid X_n = i) \\ &= \begin{cases} P(Y_{n+1} = j - i + 1), & \text{if } i \neq 0, j \geq i - 1, \\ P(Y_{n+1} = j), & \text{if } i = 0, j \geq 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (7.48)$$

Since all jobs are statistically identical, we expect that the $\{Y_n\}$ terms are identically distributed with the pmf $P(Y_{n+1} = j) = a_j$ so that

$$\sum_{j=1}^{\infty} a_j = 1.$$

Then the (infinite-dimensional) transition probability matrix of $\{X_n\}$ is given by

$$P = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_0 & a_1 & a_2 & a_3 & \cdots \\ 0 & a_0 & a_1 & a_2 & \cdots \\ 0 & 0 & a_0 & a_1 & \cdots \\ 0 & 0 & 0 & a_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (7.49)$$

This Matrix structure is known as *upper Hessenberg*. Let the limiting probability of being in state j be denoted by v_j , so that

$$v_j = \lim_{n \rightarrow \infty} P(X_n = j). \quad (7.50)$$

Using equation (7.18), we get

$$v_j = v_0 a_j + \sum_{i=1}^{j+1} v_i a_{j-i+1}. \quad (7.51)$$

If we define the generating function $G(z) = \sum_{j=0}^{\infty} v_j z^j$, then, since

$$\sum_{j=0}^{\infty} v_j z^j = \sum_{j=0}^{\infty} v_0 a_j z^j + \sum_{j=0}^{\infty} \sum_{i=1}^{j+1} v_i a_{j-i+1} z^j,$$

we obtain

$$\begin{aligned} G(z) &= v_0 \sum_{j=0}^{\infty} a_j z^j + \sum_{i=1}^{\infty} \sum_{j=i-1}^{\infty} v_i a_{j-i+1} z^j \\ &\quad (\text{interchanging the order of summation}) \end{aligned}$$

$$\begin{aligned}
&= v_0 \sum_{j=0}^{\infty} a_j z^j + \sum_{i=1}^{\infty} \sum_{k=0}^{\infty} v_i a_k z^{k+i-1} \\
&= v_0 \sum_{j=0}^{\infty} a_j z^j + \frac{1}{z} \left[\sum_{i=1}^{\infty} v_i z^i \sum_{k=0}^{\infty} a_k z^k \right].
\end{aligned}$$

Defining $G_A(z) = \sum_{j=0}^{\infty} a_j z^j$, we have

$$G(z) = v_0 G_A(z) + \frac{1}{z} [G(z) - v_0] G_A(z)$$

or

$$G(z) = \frac{(z-1)v_0 G_A(z)}{z - G_A(z)}.$$

Since $G(1) = 1 = G_A(1)$, we can use l'Hôpital's rule to obtain

$$\begin{aligned}
G(1) = 1 &= \lim_{z \rightarrow 1} v_0 \frac{(z-1)G'_A(z) + G_A(z)}{1 - G'_A(z)} \\
&= \frac{v_0}{1 - G'_A(1)},
\end{aligned}$$

provided $G'_A(1)$ is finite and less than unity. [Note that $G'_A(1) = E[Y]$.] If we let $\rho = G'_A(1)$, it follows that

$$v_0 = 1 - \rho, \quad (7.52)$$

and, since v_0 is the probability that the server is idle, ρ is the server utilization in the limit (we assume throughout that $\rho < 1$). Also, we then have

$$G(z) = \frac{(1-\rho)(z-1)G_A(z)}{z - G_A(z)}. \quad (7.53)$$

Thus, if we knew the generating function $G_A(z)$, we could compute $G(z)$, from which we could compute the steady-state average number of jobs in the system by using

$$E[N] = \lim_{n \rightarrow \infty} E[X_n] = G'(1). \quad (7.54)$$

In order to evaluate $G_A(z)$, we first compute $a_j = P(Y_{n+1} = j)$. This is the probability that exactly j jobs arrive during the service time of the $(n+1)$ st job. Let the random variable B denote job service times. Now we obtain the conditional pmf of Y_{n+1}

$$P(Y_{n+1} = j | B = t) = e^{-\lambda t} \frac{(\lambda t)^j}{j!}$$

by the Poisson assumption. Using the (continuous version) theorem of total probability, we get

$$\begin{aligned} a_j &= \int_0^\infty P(Y_{n+1} = j \mid B = t) f_B(t) dt \\ &= \int_0^\infty e^{-\lambda t} \frac{(\lambda t)^j}{j!} f_B(t) dt. \end{aligned}$$

Therefore

$$\begin{aligned} G_A(z) &= \sum_{j=0}^{\infty} a_j z^j \\ &= \sum_{j=0}^{\infty} \int_0^\infty e^{-\lambda t} \frac{(\lambda t z)^j}{j!} f_B(t) dt \\ &= \int_0^\infty e^{-\lambda t} \left[\sum_{j=0}^{\infty} \frac{(\lambda t z)^j}{j!} \right] f_B(t) dt \\ &= \int_0^\infty e^{-\lambda t} e^{\lambda t z} f_B(t) dt \\ &= \int_0^\infty e^{-\lambda t(1-z)} f_B(t) dt \\ &= L_B[\lambda(1-z)], \end{aligned} \tag{7.55}$$

where $L_B[\lambda(1-z)]$ is the Laplace–Stieltjes transform of the service time distribution evaluated at $s = \lambda(1-z)$. Note that

$$\begin{aligned} \rho &= G'_A(1) = \left. \frac{dL_B[\lambda(1-z)]}{dz} \right|_{z=1} \\ &= \left. \frac{dL_B}{ds} \right|_{s=0} (-\lambda) \end{aligned}$$

by the chain rule, so

$$\rho = \lambda E[B] = \frac{\lambda}{\mu} \tag{7.56}$$

by the moment generating property of the Laplace transform. Here the reciprocal of the average service rate μ of the server equals the average service time $E[B]$.

Substituting (7.55) in (7.53), we get the well-known **Pollaczek–Khinchin (P–K) transform** equation:

$$G(z) = \frac{(1-\rho)(z-1)L_B[\lambda(1-z)]}{z - L_B[\lambda(1-z)]}. \tag{7.57}$$

The average number of jobs in the system, in the steady state, is determined by taking the derivative with respect to z and then taking the limit $z \rightarrow 1$:

$$E[N] = \lim_{n \rightarrow \infty} E[X_n] = \sum_{j=1}^{\infty} j v_j = \lim_{z \rightarrow 1} G'(z). \quad (7.58)$$

As an example, consider the $M/M/1$ queue with $f_B(x) = \mu e^{-\mu x}, x > 0$, and hence $L_B(s) = \mu/(s + \mu)$. It follows that

$$\begin{aligned} G(z) &= \frac{\mu - \lambda}{\mu - z\lambda} \\ &= \frac{1 - \rho}{1 - \rho z} \\ &= (1 - \rho) \sum_{j=0}^{\infty} (\rho z)^j. \end{aligned}$$

The coefficient of z^j in $G(z)$ gives the value of v_j :

$$v_j = (1 - \rho)\rho^j, \quad j = 0, 1, 2, \dots.$$

It follows that the number of jobs in the system has a modified geometric distribution with parameter $(1 - \rho)$. Therefore, the expected number of jobs in the system is given by

$$E[N] = \frac{\rho}{1 - \rho}. \quad (7.59)$$

This expression for $E[N]$ can also be obtained by taking the derivative of the generating function:

$$E[N] = G'(1) = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho}.$$

More generally, it can be shown [KLEI 1975] that

$$E[N] = \rho + \frac{\lambda^2 E[B^2]}{2(1 - \rho)} = \rho + \frac{\rho^2(1 + C_B^2)}{2(1 - \rho)}. \quad (7.60)$$

This is known as the **P–K mean-value formula**. Note that the average number of jobs in the system depends only on the first two moments of the service time distribution. In fact, $E[N]$ grows linearly with the squared coefficient of variation of the service time distribution. In particular, if we consider the $M/D/1$ system then $C_B^2 = 0$ and

$$E[N]_{M/D/1} = \rho + \frac{\rho^2}{2(1 - \rho)}. \quad (7.61)$$

Although we have assumed that the scheduling discipline is FCFS, all the results in this section hold under rather general scheduling disciplines provided we assume that

1. The server is not idle whenever a job is waiting for service.
2. The scheduling discipline does not base job sequencing on any a priori information on job execution times.
3. The scheduling is nonpreemptive; that is, once a job is scheduled for service, it is allowed to complete without interruption.

The method of embedded DTMC has been applied to other non-Markovian queuing systems such as $GI/M/1$ [KULK 1995] and $M/G/c$ queue with vacation [BOLC 1998], among others. For many other results on related queuing systems, see [GROS 1998].

Despite the enormous queuing theory literature, little work can be found regarding time-dependent (or transient) behavior, especially for non-Markovian queuing systems. A non-Markovian model can be Markovized using phase-type approximation. However, phase-type expansion increases the already large state space of a real system model. The problem becomes really severe when mixing non-exponential times with exponential ones. In the alternative approach, the process can be shown to be a Markov regenerative one (also known as a *semiregenerative process*), and therefore Markov renewal theory can be applied for its long-run as well as time-dependent behavior. Recently, several researchers have begun work on transient analysis of Markov regenerative process [CHOI 1994, GERM 2000] as well as on the automated generation of such processes starting from non-Markovian stochastic Petri nets [CHOI 1994, GERM 1994], and several applications have been solved in performance/reliability analysis of computer/communication systems [GERM 2000, LOGO 1994, LOGO 1995].

Problems

1. Jobs submitted to a university departmental file server can be divided into three classes:

Type	Relative frequency	Mean execution time (in seconds)
Student jobs	0.8	1
Faculty jobs	0.1	20
Administrative jobs	0.1	5

Assuming that, within a class, execution times are one-stage, two-stage, and three-stage Erlang, respectively, compute the average number of jobs in the server assuming a Poisson overall arrival stream of jobs with average rate of 0.1 jobs per second. Assume that all classes are treated equally by the scheduler.

2. For the $M/G/1$ queue, plot the average number in the system $E[N]$ as a function of server utilization ρ for several different service time distributions:
- Deterministic
 - Exponential
 - k -stage Erlang, $k = 2, 5$
 - k -stage hyperexponential, $k = 2; \alpha_1 = 0.5, \alpha_2 = 0.5; \mu_1 = 1, \mu_2 = 10$
3. Consider a computer system with a CPU and one disk drive. After a burst at the CPU the job completes execution with probability 0.1 and requests a disk I/O with probability 0.9. The time of a single CPU burst is exponentially distributed with mean 0.01 s. The disk service time is broken up into three phases: exponentially distributed seek time with mean 0.03 s, uniformly distributed latency time with mean 0.01 s, and a constant transfer time equal to 0.01 s. After a service completion at the disk, the job always requires a CPU burst. The average arrival rate of jobs is 0.8 job/s and the system does not have enough main memory to support multiprogramming. Solve for the average response time using the $M/G/1$ model. In order to compute the mean and the variance of the service time distribution, you may need the results of the section on random sums in Chapter 5.
4. Starting with the Pollaczek–Khinchin transform equation (7.57), derive expressions for the average number in the system $E[N]$ for an $M/G/1$ queue, assuming
- Deterministic service times ($M/D/1$)
 - Two-stage Erlang service time distribution ($M/E_2/1$)
 - Two-stage hyperexponential service time distribution ($M/H_2/1$)
5. * Consider a modification of the $M/G/1$ queue with FCFS scheduling so that after the completion of a service burst, the job returns to the queue with probability q and completes execution with probability p (see Figure 7.P.1). We wish to obtain the queue-length pmf in the steady-state as seen by a completer and as seen by a deporter. First consider the deporter's distribution. Using the notion

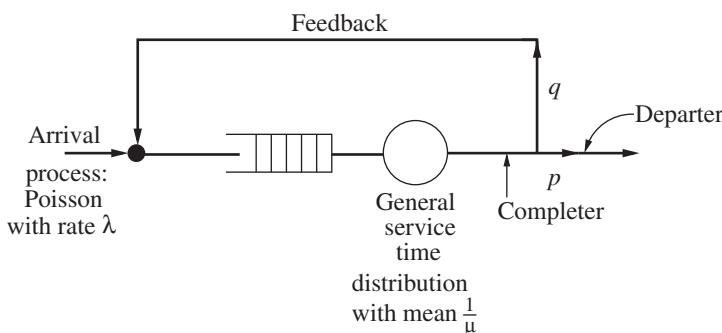


Figure 7.P.1. The $M/G/1$ queue with Bernoulli feedback

of random sums, first derive the Laplace–Stieltjes transform for the total service time T of a job as

$$L_T(s) = \frac{pL_B(s)}{1 - qL_B(s)},$$

where B is the random variable denoting the length of a single service burst. Now show that the generating function of N_d , the number of jobs in the system (in the steady state) as seen by a departer, defined by

$$G_{N_d}(z) = \sum_{k=0}^{\infty} p_{N_d}(k) z^k,$$

is given by

$$G_{N_d}(z) = \left(1 - \frac{\lambda}{\mu p}\right) \frac{p(1-z)L_B[\lambda(1-z)]}{(p+qz)L_B[\lambda(1-z)] - z}.$$

Find the average number of jobs $E[N_d]$ as seen by the departer. Specializing to the case in which the service time distribution is exponential, obtain the pmf of N_d .

Next consider the embedded Markov chain, where the completion of a service burst is defined to be an epoch. Show that the transition probability matrix of this Markov chain is as follows:

$$\begin{bmatrix} pa_0 & pa_1 + qa_0 & pa_2 + qa_1 & \dots \\ pa_0 & pa_1 + qa_0 & pa_2 + qa_1 & \dots \\ 0 & pa_0 & pa_1 + qa_0 & \dots \\ 0 & 0 & pa_0 & \dots \\ 0 & 0 & 0 & \dots \\ \cdot & \cdot & \cdot & \dots \\ \cdot & \cdot & \cdot & \dots \\ \cdot & \cdot & \cdot & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix}$$

where

$$\begin{aligned} a_i &= P(Y_{n+1} = i) \\ &= P[\text{"jobs arrive during the } (n+1)\text{st service burst"}]. \end{aligned}$$

Now show that the generating function of the steady-state number of jobs in the system as seen by a completer, defined by

$$G_{N_c}(z) = \sum_{k=0}^{\infty} p_{N_c}(k) z^k,$$

is given by

$$G_{N_c}(z) = p \left(1 - \frac{\lambda}{\mu p}\right) \frac{(p+qz)(1-z)L_B[\lambda(1-z)]}{(p+qz)L_B[\lambda(1-z)] - z}.$$

Find the average number in the system $E[N_c]$ as seen by a completer.

7.8 DISCRETE-TIME BIRTH-DEATH PROCESSES

We consider a special type of discrete-time Markov chain with all one-step transitions to nearest neighbors only. The transition probability matrix P is a tridiagonal matrix. To simplify notation, we let

$$\begin{aligned} b_i &= p_{i,i+1}, \quad i \geq 0 \quad \{\text{the probability of a birth in state } i\}, \\ d_i &= p_{i,i-1}, \quad i \geq 1 \quad \{\text{the probability of a death in state } i\}, \\ a_i &= p_{i,i}, \quad i \geq 0. \quad \{\text{the probability being in state } i\}. \end{aligned}$$

Note that $(a_i + b_i + d_i) = 1$ for all i . Thus the (infinite-dimensional) matrix P is given by

$$P = \begin{bmatrix} a_0 & b_0 & 0 & \cdot & \cdot & \cdot & & \\ d_1 & a_1 & b_1 & 0 & \cdot & \cdot & & \cdots \\ 0 & d_2 & a_2 & b_2 & 0 & \cdot & & \cdots \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & & \cdots \\ \vdots & \vdots & \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{i-1} & b_{i-1} & 0 & \cdot & \cdot & \cdots \\ 0 & 0 & 0 & 0 & \cdots & d_i & a_i & b_i & 0 & \cdot & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & d_{i+1} & a_{i+1} & b_{i+1} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & & & & & & & \vdots \end{bmatrix}$$

and the state diagram is shown in Figure 7.19. If we assume that $b_i > 0$ and $d_i = 0$ for all i , then all the states of the Markov chain are transient. Similarly, if we let $b_i = 0$ and $d_i > 0$ for all i , then all the states are transient except the state labeled 0, which will be an absorbing state. We will assume that $0 < b_i, d_i < 1$ for all $i \geq 1$ and $b_0 > 0$; hence the Markov chain is irreducible and aperiodic, which implies by Theorem 7.3 that the limiting probabilities exist and are independent of the initial probability vector. To compute the steady-state probability vector \mathbf{v} (if it exists), we use

$$\mathbf{v} = \mathbf{v}P$$

and get

$$v_0 = a_0 v_0 + d_1 v_1, \tag{7.62}$$

$$v_i = b_{i-1} v_{i-1} + a_i v_i + d_{i+1} v_{i+1}, \quad i \geq 1. \tag{7.63}$$

Since

$$1 - a_i = b_i + d_i,$$

from (7.63) we get

$$b_i v_i - d_{i+1} v_{i+1} = b_{i-1} v_{i-1} - d_i v_i,$$

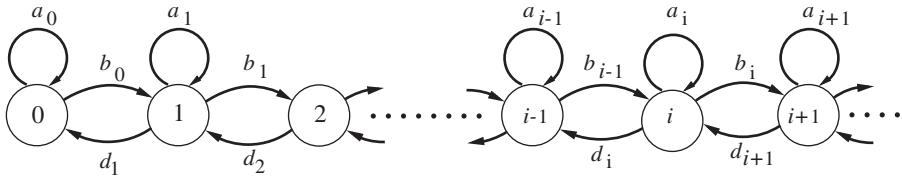


Figure 7.19. The state diagram of the discrete-time birth-death process

so, $b_i v_i - d_{i+1} v_{i+1}$ is independent of i , but $b_0 v_0 - d_1 v_1 = (1 - a_0)v_0 - d_1 v_1 = 0$ from (7.62). Therefore the solution to the above system of equations is given by

$$v_i = \frac{b_{i-1}}{d_i} v_{i-1} = \prod_{j=1}^i \frac{b_{j-1}}{d_j} v_0. \quad (7.64)$$

Now, using the condition $\sum_{i \geq 0} v_i = 1$, we get

$$v_0 = \frac{1}{\sum_{i \geq 0} \prod_{j=1}^i \frac{b_{j-1}}{d_j}}, \quad (7.65)$$

provided the series converges. If the series in the denominator diverges, then we can conclude that all states of the Markov chain are recurrent null. We will assume that the series is convergent, that is, that all states are recurrent nonnull; hence (7.64) and (7.65) give the unique steady-state probabilities.

Example 7.15 (Analysis of a Data Structure)

Consider a data structure (such as a linear list) being manipulated in a program. Suppose that we are interested only in the amount of memory consumed by the data structure. If the current amount of memory in use is i nodes, then we say that the state of the structure is s_i . Let probabilities associated with the next operation on the data structure be given by

$$\begin{aligned} b_i &= P(\text{"next operation is an insert"} \mid \text{"current state is } s_i\text{"}), \\ d_i &= P(\text{"next operation is a delete"} \mid \text{"current state is } s_i\text{"}), \\ a_i &= P(\text{"next operation is an access"} \mid \text{"current state is } s_i\text{"}). \end{aligned}$$

Then the steady-state pmf of the number of nodes in use is given by equations (7.64) and (7.65) above.

As a special case, we let $b_i = b$ ($i \geq 0$) and $d_i = d$ ($i \geq 1$) for all i . Then, assuming $b < d$ (for the chain to have recurrent nonnull states), we have

$$v_i = \frac{b}{d} v_{i-1} \quad \text{and} \quad v_0 = \frac{1}{\sum_{i \geq 0} \left(\frac{b}{d}\right)^i} = 1 - \frac{b}{d}$$

or

$$v_i = \left(1 - \frac{b}{d}\right) \left(\frac{b}{d}\right)^i.$$

Thus the steady-state pmf is modified geometric with parameter $(1 - b/d)$. The expected number of nodes in use in the steady state is given by $(b/d)/[1 - (b/d)] = b/(d - b)$. These formulas are valid under the assumption that

$$\sum_{i \geq 0} \left(\frac{b}{d}\right)^i \text{ is finite.} \quad (7.66)$$

This assumption is satisfied provided $b/d < 1$, or the probability of insertion is strictly less than the probability of deletion. If this condition is not satisfied, then the data structure will tend to grow continually, resulting in a memory overflow. ‡

Example 7.16

In Example 7.15, we assumed that a potentially infinite number of nodes are available for allocation. Next assume that a limited number $m \geq 1$ of nodes are available for allocation. Then, if m nodes are in use, an insertion operation will give rise to an overflow. We assume that such an operation is simply ignored, leaving the system in state s_m . The state diagram is given in Figure 7.20. The steady-state solution to this system is given by

$$v_i = \left(\frac{b}{d}\right)^i v_0, i = 0, 1, \dots, m$$

and

$$v_0 = \frac{1}{\sum_{i=0}^m \left(\frac{b}{d}\right)^i} = \frac{1 - \left(\frac{b}{d}\right)}{1 - \left(\frac{b}{d}\right)^{m+1}}.$$

Now the probability of an overflow is computed by

$$\begin{aligned} P_{ov} &= bv_m = b \left(\frac{b}{d}\right)^m \frac{1 - \left(\frac{b}{d}\right)}{1 - \left(\frac{b}{d}\right)^{m+1}} \\ &= \frac{b^{m+1}(d - b)}{d^{m+1} - b^{m+1}}. \end{aligned} \quad (7.67)$$

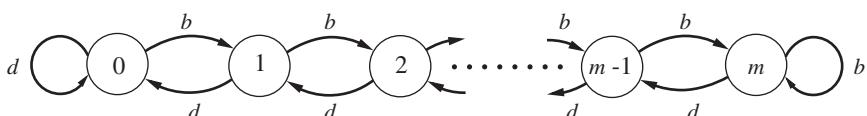


Figure 7.20. The state diagram for Example 7.16

Similarly, the probability of underflow is given by

$$\begin{aligned}
 P_{uf} = dv_0 &= d \cdot \frac{1 - \frac{b}{d}}{1 - \left(\frac{b}{d}\right)^{m+1}} \\
 &= \frac{d^{m+1}(d-b)}{d^{m+1} - b^{m+1}}. \tag{7.68}
 \end{aligned}$$

‡

The notion of the birth-death process can be generalized to multidimensional birth-death processes. We will introduce such processes through examples.

Example 7.17

Consider a program that uses two stacks, sharing an area of memory containing m locations. The stacks grow toward each other from the two opposite ends (see Figure 7.21). Clearly, an overflow will occur on an insertion into either stack when $i + j = m$. Let the state of the system be denoted by the pair (i, j) . Then the state space is $I = \{(i, j) \mid i, j \geq 0, i + j \leq m\}$. We assume that an overflow is simply ignored, leaving the state of the system unchanged. Underflow also does not change the system state as before.

At each instant of time, an operation on one of the stacks takes place with respective probabilities as shown in the probability tree of Figure 7.22. Thus, for $i = 1, 2$,

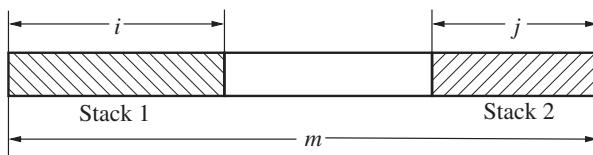


Figure 7.21. Two stacks sharing an area of memory

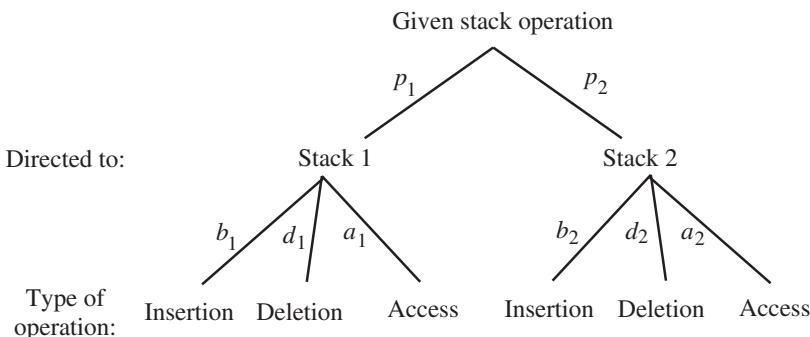


Figure 7.22. Tree diagram for Example 7.17

p_i is the probability that a given operation is directed to stack i , and b_i , d_i , and a_i , respectively denote the probabilities that this operation is an insertion, deletion, or access. The system behavior corresponds to a two-dimensional birth-death process with the state diagram shown in Figure 7.23. The steady-state probability vector

$$\mathbf{v} = [v_{(0,0)}, v_{(0,1)}, \dots, v_{(0,m)}, \dots, v_{(i,j)}, \dots, v_{(m,0)}]$$

may be obtained by solving

$$\mathbf{v} = \mathbf{v}P$$

$$a = p_1 a_1 + p_2 a_2$$

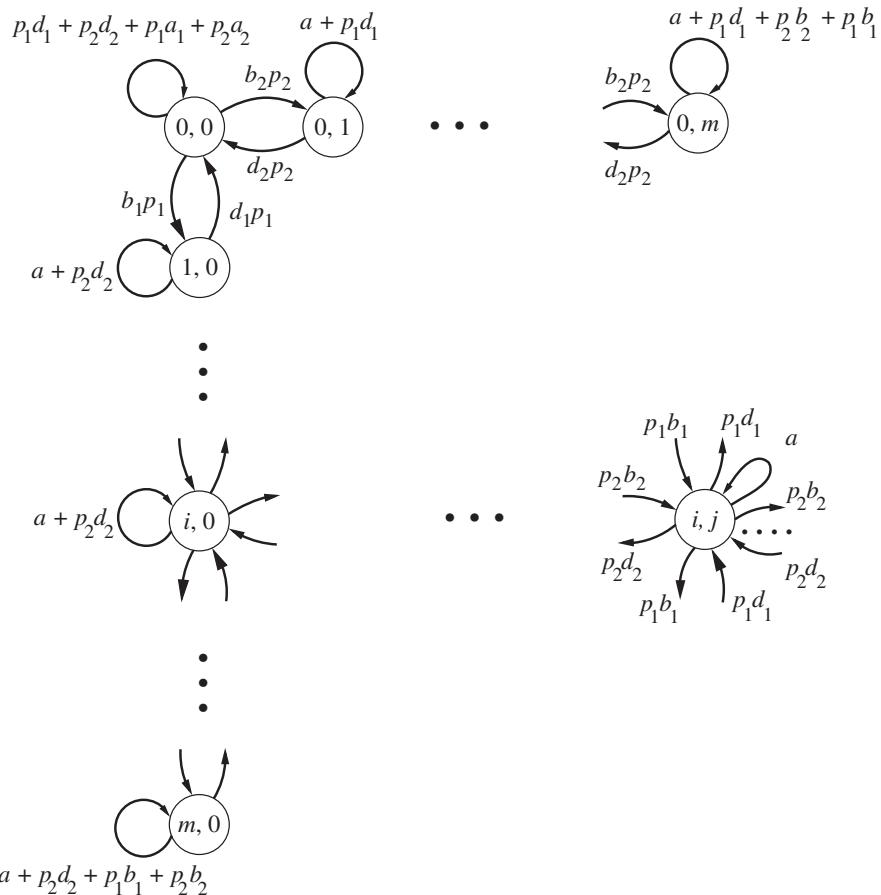


Figure 7.23. The state diagram for the two-stacks example

(and using the identities $a_1 + b_1 + d_1 = 1$, $a_2 + b_2 + d_2 = 1$, and $p_1 + p_2 = 1$). Thus

$$\begin{aligned}
 v_{(i,j)} &= b_2 p_2 v_{(i,j-1)} + d_2 p_2 v_{(i,j+1)} + (a_1 p_1 + a_2 p_2) v_{(i,j)} \\
 &\quad + b_1 p_1 v_{(i-1,j)} + d_1 p_1 v_{(i+1,j)}, & 1 \leq i, j; i+j < m, \\
 v_{(i,0)} &= (a_1 p_1 + a_2 p_2 + d_2 p_2) v_{(i,0)} + d_2 p_2 v_{(i,1)} \\
 &\quad + b_1 p_1 v_{(i-1,0)} + d_1 p_1 v_{(i+1,0)}, & 1 \leq i \leq m-1, \\
 v_{(0,j)} &= (a_1 p_1 + a_2 p_2 + d_1 p_1) v_{(0,j)} + d_1 p_1 v_{(1,j)} \\
 &\quad + b_2 p_2 v_{(0,j-1)} + d_2 p_2 v_{(0,j+1)}, & 1 \leq j \leq m-1, \\
 v_{(0,0)} &= (d_1 p_1 + d_2 p_2 + a_1 p_1 + a_2 p_2) v_{(0,0)} + d_1 p_1 v_{(1,0)} \\
 &\quad + d_2 p_2 v_{(0,1)}, \\
 v_{(0,m)} &= (a_1 p_1 + a_2 p_2 + d_1 p_1 + b_1 p_1 + b_2 p_2) v_{(0,m)} + b_2 p_2 v_{(0,m-1)}, \\
 v_{(m,0)} &= (a_1 p_1 + a_2 p_2 + d_2 p_2 + b_1 p_1 + b_2 p_2) v_{(m,0)} + b_1 p_1 v_{(m-1,0)}.
 \end{aligned}$$

It may be verified by direct substitution that

$$v_{(i,j)} = v_{(0,0)} \left(\frac{b_1}{d_1} \right)^i \left(\frac{b_2}{d_2} \right)^j, \quad i, j \geq 0, i+j \leq m.$$

We will use the abbreviation $r_1 = b_1/d_1$ and $r_2 = b_2/d_2$. Then

$$v_{(i,j)} = v_{(0,0)} r_1^i r_2^j.$$

The normalization requirement yields

$$\begin{aligned}
 1 &= \sum_{i=0}^m \sum_{j=0}^{m-i} v_{(i,j)} \\
 &= v_{(0,0)} \sum_{i=0}^m \left[\sum_{j=0}^{m-i} (r_2^j) \right] r_1^i \\
 &= v_{(0,0)} \sum_{i=0}^m \frac{1 - r_2^{m-i+1}}{1 - r_2} r_1^i \\
 &= \frac{v_{(0,0)}}{1 - r_2} \sum_{i=0}^m \left[r_1^i - r_2^{m+1} \left(\frac{r_1}{r_2} \right)^i \right] \\
 &= \begin{cases} \frac{v_{(0,0)}}{1 - r_2} \left[\frac{1 - r_1^{m+1}}{1 - r_1} - r_2^{m+1} \frac{1 - (r_1/r_2)^{m+1}}{1 - (r_1/r_2)} \right], & \text{if } r_1 \neq r_2; r_1 \neq 1, r_2 \neq 1 \\ \frac{v_{(0,0)}}{1 - r_1} \left[\frac{1 - r_1^{m+1}}{1 - r_1} - (m+1)r_1^{m+1} \right], & \text{if } r_1 = r_2 \neq 1. \end{cases}
 \end{aligned}$$

Simplifying, we obtain

$$v_{(0,0)} = \begin{cases} \frac{(1-r_1)(1-r_2)}{1 - \frac{1}{r_2-r_1}\{r_2^{m+2}(1-r_1) - r_1^{m+2}(1-r_2)\}}, & r_1 \neq r_2, \\ \frac{(1-r_1)^2}{1 - (m+2)r_1^{m+1} + (m+1)r_1^{m+2}}, & r_1 = r_2 \neq 1 \\ \frac{2}{(m+1)(m+2)}, & r_1 = r_2 = 1. \end{cases}$$

The probability of overflow is given by

$$\begin{aligned} P_{ov} &= \sum_{i+j=m} v_{(i,j)}(b_1 p_1 + b_2 p_2) \\ &= v_{(0,0)} \sum_{i=0}^m r_1^i r_2^{m-i} (b_1 p_1 + b_2 p_2) \\ &= \begin{cases} (b_1 p_1 + b_2 p_2) v_{(0,0)} r_2^m \frac{1 - (r_1/r_2)^{m+1}}{1 - (r_1/r_2)}, & r_1 \neq r_2, \\ (m+1)(b_1 p_1 + b_2 p_2) v_{(0,0)} r_1^m, & r_1 = r_2. \end{cases} \end{aligned} \quad (7.69)$$

¶

Example 7.18

We want to implement two stacks within $2k$ memory locations. The first solution is to divide the given area into two equal areas and preallocate the two areas to the two stacks. Assume $p_1 = p_2 = \frac{1}{2}$, $b_1 = b_2 = b$, $d_1 = d_2 = d$, and hence $r_1 = r_2 = r = b/d$.

Under the first scheme, the overflow in each stack occurs with probability [using equation (7.67)]

$$\frac{br^k(1-r)}{1 - r^{k+1}}$$

where the total overflow probability is twice as much.

Under the second scheme, where the two stacks grow toward each other, we have

$$v_{(0,0)} = \frac{(1-r)^2}{1 - (2k+2)r^{2k+1} + (2k+1)r^{2k+2}}$$

and the overflow probability is given by [using equation (7.69)]

$$P_{ov} = (2k+1)(b) \frac{(1-r)^2 r^{2k}}{1 - (2k+2)r^{2k+1} + (2k+1)r^{2k+2}}.$$

Then the condition under which the second scheme is better than the first one is given by

$$b(2k+1) \frac{(1-r)^2 r^{2k}}{1 - (2k+2)r^{2k+1} + (2k+1)r^{2k+2}} \leq 2br^k \frac{1-r}{1 - r^{k+1}}.$$

Assuming for simplicity that $r < 1$, this condition is rewritten as

$$\frac{\frac{(2k+1)r^k}{1-r^{2k+1}} - (2k+1)r^{2k+1}}{1-r} \leq \frac{2}{1-r^{k+1}}$$

and hence as

$$\begin{aligned} (2k+1)r^k + (2k+1)r^{2k+1} &\leq 2 \frac{1-r^{2k+1}}{1-r} \\ &= \left[2\left(\sum_{i=0}^{k-1} r^i\right) + r^k \right] + \left\{ r^k + 2 \left(\sum_{i=k+1}^{2k} r^i \right) \right\}. \end{aligned} \quad (7.70)$$

In order to show that inequality (7.70) holds, observe that there are $(2k+1)$ terms in the expression within the square brackets and each term is greater than or equal to r^k (since $r < 1$). Similarly, each of $(2k+1)$ term of the expression within braces is greater than or equal to r^{2k+1} .

Thus we conclude that the second scheme of sharing the $2k$ locations between the two stacks is superior to the first scheme of preallocating half the available area to each stack.

#

7.9 FINITE MARKOV CHAINS WITH ABSORBING STATES

In Chapter 5 we discussed the analysis of properly nested programs. Many programs, however, are not properly nested in that they contain **goto** statements. Program graphs associated with such a program can be treated as the state diagram of a discrete-time Markov chain, with appropriate assumptions. Since the program is eventually expected to terminate, it will contain certain “final” or “stopping” states. In the terminology of Markov chains, such states are called *absorbing states*. Also, since the number of statements in the program will be finite, the corresponding chain will have a finite number of states.

Consider a program with its associated directed graph as shown in Figure 7.24. Such a control flow graph of the program is sometimes known as its architecture [GOKH 1998]. Each vertex s_j in the figure represents a group of statements with a single entry point and a single exit point. The last statement in the group is a multiway branch. Vertex s_1 is the start vertex. Vertex s_5 has no outgoing edges and thus is a stop vertex. The weight p_{ij} of edge (s_i, s_j) is interpreted as the conditional probability that the program will next execute statement group s_j , given that it has just completed the execution of statement group s_i . We have assumed that this probability depends only on the current statement group and not on the previous history of the program. Therefore, the corresponding Markov chain will be homogeneous. Herein lies a serious deficiency of the model. In actual

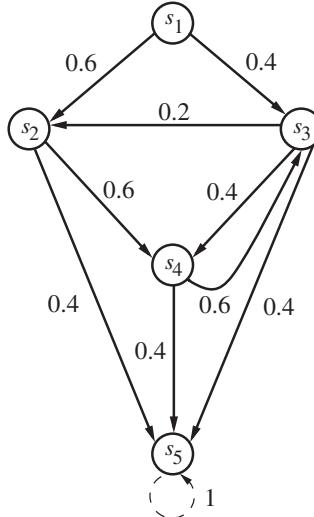


Figure 7.24. A program flow graph

programs such probabilities are not likely to be independent of previous history, and a more accurate model involving the use of nonhomogeneous Markov chains is desirable. We will, however, continue with the simplified, albeit inaccurate, model using homogeneous Markov chains. It is possible to automatically derive the control flow graphs of programs using information collected from the testing of the programs [GOKH 1998].

While interpreting the program flow graph of Figure 7.24 as the state diagram of a finite, discrete-time, homogeneous Markov chain, we encounter one difficulty. From the absorbing state s_5 there are no outgoing edges. Then $p_{5j} = 0$ for all j . But the assumption of the Markov chain requires that $\sum_j p_{ij} = 1$ for each i . To avoid this difficulty, we imagine a “dummy” edge forming a self-loop on the absorbing state s_5 . With this modification, the transition probability matrix of the Markov chain is given by

$$P = \begin{matrix} & s_1 & s_2 & s_3 & s_4 & s_5 \\ s_1 & 0 & 0.6 & 0.4 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.6 & 0.4 \\ s_3 & 0 & 0.2 & 0 & 0.4 & 0.4 \\ s_4 & 0 & 0 & 0.6 & 0 & 0.4 \\ s_5 & 0 & 0 & 0 & 0 & 1 \end{matrix}.$$

Note that states s_1 through s_4 are transient states while state s_5 is an absorbing state.

In general, we consider a Markov chain with n states, s_1, s_2, \dots, s_n , where s_n is the absorbing state, and the remaining states are transient. The

transition probability matrix of such a chain may be partitioned so that

$$P = \left[\begin{array}{c|c} Q & \mathbf{C} \\ \hline \mathbf{0} & 1 \end{array} \right]. \quad (7.71)$$

where Q is an $(n - 1)$ by $(n - 1)$ substochastic matrix (with at least one row sum less than 1) describing the probabilities of transition only among the transient states. \mathbf{C} is a column vector and $\mathbf{0}$ is a row vector of $(n - 1)$ zeros.

Now the k -step transition probability matrix P^k has the form

$$P^k = \left[\begin{array}{c|c} Q^k & \mathbf{C}' \\ \hline \mathbf{0} & 1 \end{array} \right], \quad (7.72)$$

where \mathbf{C}' is a column vector whose elements will be of no further use and hence need not be computed. The (i, j) entry of matrix Q^k denotes the probability of arriving in (transient) state s_j after exactly k steps starting from (transient) state s_i . It can be shown that $\sum_{k=0}^t Q^k$ converges as t approaches infinity [PARZ 1962]. This implies that the inverse matrix $(I - Q)^{-1}$, called the **fundamental matrix**, M , exists and is given by

$$M = (I - Q)^{-1} = I + Q + Q^2 + \cdots = \sum_{k=0}^{\infty} Q^k.$$

The fundamental matrix M is a rich source of information on the Markov chain, as seen below. Let X_{ij} ($1 \leq i, j < n$) be the random variable denoting the number of times the program visits state s_j before entering the absorbing state, given that it started in state s_i . Let $\mu_{ij} = E[X_{ij}]$.

THEOREM 7.5. For $1 \leq i, j < n$, $E[X_{ij}] = m_{ij}$, the (i, j) th element of the fundamental matrix M .

Proof [BHAT 1984]: Initially the process is in the transient state s_i . In one step it may enter the absorbing state s_n with probability p_{in} . The corresponding number of visits to state s_j is equal to zero unless $j = i$. Thus, $X_{ij} = \delta_{ij}$ with probability p_{in} , where δ_{ij} is the Kronecker δ function ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise). Alternately, the process may go to transient state s_k at the first step (with probability p_{ik}). The subsequent number of visits to state s_j is given by X_{kj} . If $i = j$, the total number of visits, X_{ij} , to state s_j will be $X_{kj} + 1$, otherwise it will be X_{kj} . Therefore

$$X_{ij} = \begin{cases} \delta_{ij} & \text{with probability } p_{in}, \\ X_{kj} + \delta_{ij} & \text{with probability } p_{ik}, 1 \leq k < n. \end{cases}$$

If the random variable Y denotes the state of the process at the second step (given that the initial state is i), we can summarize as follows:

$$\begin{aligned} E[X_{ij} | Y = n] &= \delta_{ij}, \\ E[X_{ij} | Y = k] &= E[X_{kj} + \delta_{ij}] = E[X_{kj}] + E[\delta_{ij}] = E[X_{kj}] + \delta_{ij}. \end{aligned}$$

Now since the pmf of Y is easily derived as $P(Y = k) = p_{ik}$, $1 \leq k \leq n$, we can use the theorem of total expectation to obtain

$$\begin{aligned} \mu_{ij} &= E[X_{ij}] = \sum_k E[X_{ij} | Y = k] P(Y = k) \\ &= p_{in}\delta_{ij} + \sum_{k=1}^{n-1} p_{ik}(E[X_{kj}] + \delta_{ij}) \\ &= \sum_{k=1}^n p_{ik}\delta_{ij} + \sum_{k=1}^{n-1} p_{ik}E[X_{kj}] \\ &= \delta_{ij} + \sum_{k=1}^{n-1} p_{ik}\mu_{kj}. \end{aligned} \tag{7.73}$$

Forming the $(n - 1) \times (n - 1)$ matrix consisting of elements μ_{ij} , we have

$$[\mu_{ij}] = I + Q[\mu_{ij}]$$

or

$$[\mu_{ij}] = (I - Q)^{-1} = M. \tag{7.74}$$

For DTMC with absorbing states we will be interested in the expected accumulated reward till absorption, that is $\lim_{k \rightarrow \infty} E[Y_k]$. Let V_j denote the average number of times the statement group s_j is executed in a typical run of the program. Then, since s_1 is the starting state of the program, $V_j = m_{1j}$, the element in the first row and the j th column of the fundamental matrix, M . Now if r_j denotes the reward attached to statement group s_j (per visit), then the expected accumulated reward till absorption due to the execution of the program is given by

$$\lim_{k \rightarrow \infty} E[Y_k] = \left(\sum_{j=1}^{n-1} V_j r_j \right) + r_n = \left(\sum_{j=1}^{n-1} m_{1j} r_j \right) + r_n,$$

since the number of visits, V_n , to the stop vertex s_n is one. In case we are interested in the mean execution (completion) time of the program, the reward attached to statement group s_j will be the average execution time t_j of the statement group. If we are interested in the cache misses for the whole program, then we will assign the reward rate r_j to group s_j to be the average

number of cache misses for that group of statements s_j . An alternative form of (7.73) can be used to simplify computations of the visit counts as follows. From (7.74) we have

$$M(I - Q) = I \quad \text{or} \quad M = I + MQ.$$

Therefore, the (i, j) element of matrix M can be computed using the formula

$$m_{ij} = \delta_{ij} + \sum_{k=1}^{n-1} m_{ik} p_{kj}. \quad (7.75)$$

Recalling that $m_{1j} = V_j$, we have

$$V_j = \delta_{1j} + \sum_{k=1}^{n-1} V_k p_{kj}, \quad j = 1, 2, \dots, n-1. \quad (7.76)$$

Thus, the visit counts are obtained by solving a system of $(n - 1)$ linear equations (7.76).

Example 7.19

Returning to the program discussed earlier in this section, the matrix Q is given by

$$Q = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.6 \\ 0 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0.6 & 0 \end{bmatrix}.$$

The fundamental matrix M is computed to be

$$M = (I - Q)^{-1} = \begin{bmatrix} 1 & 0.7791 & 0.8953 & 0.8256 \\ 0 & 1.1047 & 0.5233 & 0.8721 \\ 0 & 0.2907 & 1.4535 & 0.7558 \\ 0 & 0.1744 & 0.8721 & 1.4535 \end{bmatrix}.$$

Thus the vertices s_1 , s_2 , s_3 , and s_4 are respectively executed 1, 0.7791, 0.8953, and 0.8254 times on the average. If t_j is the average execution time of statement group s_j , then the average execution time of the program is equal to

$$t_1 + 0.7791 \cdot t_2 + 0.8953 \cdot t_3 + 0.8256 \cdot t_4 + t_5 \text{ time units.}$$

Note that with the addition of execution time for each statement group, we are actually dealing with a semi-Markov process. More material on semi-Markov process is covered in Chapter 8. If n_j is the average number of cache misses for the statement groups s_j , then the overall number of cache misses for the whole program is

$$n_1 + 0.7791 \cdot n_2 + 0.8953 \cdot n_3 + 0.8256 \cdot n_4 + n_5.$$

Finally, if let R_j be the reliability of statement group s_j , then the overall reliability of the program can be approximated by

$$R = \prod_j R_j^{V_j}$$

or

$$\ln R = \sum_j V_j \ln R_j.$$

Hence, by assigning reward $r_j = \ln R_j$ to state s_j we can obtain $\ln R = \ln R_1 + 0.7791 \cdot \ln R_2 + 0.8953 \cdot \ln R_3 + 0.8256 \cdot \ln R_4 + \ln R_5$. ‡

Example 7.20

Often we are not interested in the details of computation states of a program but only wish to distinguish between the computation state and one of the m I/O states. Thus the program may appear as shown in Figure 7.25. Vertex (labeled 0) COMP is the start vertex. The transition probability matrix P is given by

$$\begin{array}{ccccccc} & \text{COMP} & \text{I/O1} & \cdots & \text{I/O}_i & \cdots & \text{I/O}_m & \text{STOP} \\ \text{COMP} & \left[\begin{matrix} 0 & p_1 & \cdots & p_i & \cdots & p_m & p_0 \\ 1 & 0 & & 0 & & 0 & 0 \\ 1 & 0 & & 0 & & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 1 & 0 & & 0 & & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 1 & 0 & & 0 & & 0 & 0 \\ 0 & 0 & & 0 & & 0 & 1 \end{matrix} \right] \end{array}$$

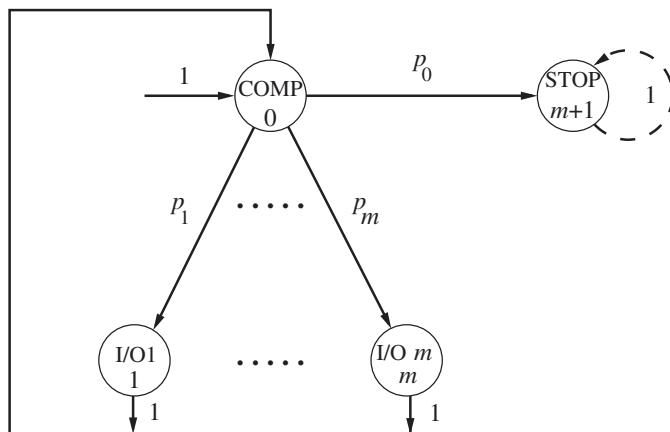


Figure 7.25. A program flow graph

The portion of the transition probability matrix for the transitions among the transient states is given by

$$Q = \begin{bmatrix} 0 & p_1 & \cdots & p_m \\ 1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

Then

$$(I - Q) = \begin{bmatrix} 1 & -p_1 & \cdots & -p_m \\ -1 & 1 & 0 & 0 \\ \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

and the fundamental matrix

$$M = (I - Q)^{-1} = \begin{bmatrix} \frac{1}{p_0} & \frac{p_1}{p_0} & \frac{p_2}{p_0} & \cdots & \frac{p_m}{p_0} \\ \frac{1}{p_0} & 1 + \frac{p_1}{p_0} & \frac{p_2}{p_0} & \cdots & \frac{p_m}{p_0} \\ \vdots & & & & \\ \frac{1}{p_0} & \frac{p_1}{p_0} & \frac{p_2}{p_0} & \cdots & 1 + \frac{p_m}{p_0} \end{bmatrix}.$$

Thus $V_0 = m_{00} = 1/p_0$ is the average number of times the COMP state is visited, and $V_j = m_{0j} = p_j/p_0$ is the average number of times the state I/O j is visited.

This result can be derived directly using equation (7.76):

$$\begin{aligned} V_0 &= \delta_{00} + \sum_{k=0}^m p_{k0} V_k, j = 0, \\ &= 1 + \sum_{k=1}^m V_k \end{aligned}$$

and

$$\begin{aligned} V_j &= \delta_{0j} + \sum_{k=0}^m p_{kj} V_k, j = 1, 2, \dots, m, \\ &= p_j V_0. \end{aligned}$$

Solving, we get

$$V_j = \begin{cases} \frac{1}{p_0}, & j = 0, \\ \frac{p_j}{p_0}, & j = 1, 2, \dots, m. \end{cases} \quad (7.77)$$

#

So far, we have assumed that there is a unique entry point (START state) to the program. The results above are easily generalized to the case with multiple entry points. Suppose that the program starts from vertex s_j with probability q_j ($1 \leq j \leq n$) so that $\sum_{j=1}^n q_j = 1$. Since m_{ij} denotes the

average number of times node j is visited given that the process started in node i , the average number, V_j , of times node j is visited without conditioning on the START state, is given by the theorem of total expectation as

$$V_j = \sum_{i=1}^n m_{ij} q_i. \quad (7.78)$$

Substituting expression (7.75) for m_{ij} , we have

$$\begin{aligned} V_j &= \sum_{i=1}^n q_i \left[\delta_{ij} + \sum_{k=1}^{n-1} m_{ik} p_{kj} \right] \\ &= \sum_{i=1}^n q_i \delta_{ij} + \sum_{i=1}^n q_i \sum_{k=1}^{n-1} m_{ik} p_{kj} \\ &= q_j + \sum_{k=1}^{n-1} p_{kj} \sum_{i=1}^n m_{ik} q_i, \end{aligned}$$

interchanging the order of summation. Now, using (7.78), we have

$$V_j = q_j + \sum_{k=1}^{n-1} p_{kj} V_k, \quad j = 1, 2, \dots, n-1. \quad (7.79)$$

Clearly, for the STOP state s_n , the VISIT count still remains at 1. Equation (7.79) will be used again in Chapter 9.

So far we have assumed a single absorbing state. We next consider multiple absorbing states on a homogeneous DTMC. Assume that there are $n-m$ transient states and m absorbing states. We organize the transition probability matrix P as in equation (7.71):

$$P = \left[\begin{array}{c|c} Q & C \\ \hline 0 & I \end{array} \right],$$

where Q is an $(n-m)$ by $(n-m)$ substochastic matrix, I is an m by m identity matrix and C is a rectangular matrix that is $(n-m)$ by m . Define the matrix $A = [a_{ik}]$ so that a_{ik} denotes the probability that the DTMC starting with a transient state i eventually gets absorbed in an absorbing state k . Then it can be shown that $A = (I - Q)^{-1}C$ [MEDH 1994].

Example 7.21

We extend the program graph in the beginning of this section to include the possibility of failure for each statement group. Two absorbing states, S and F , are

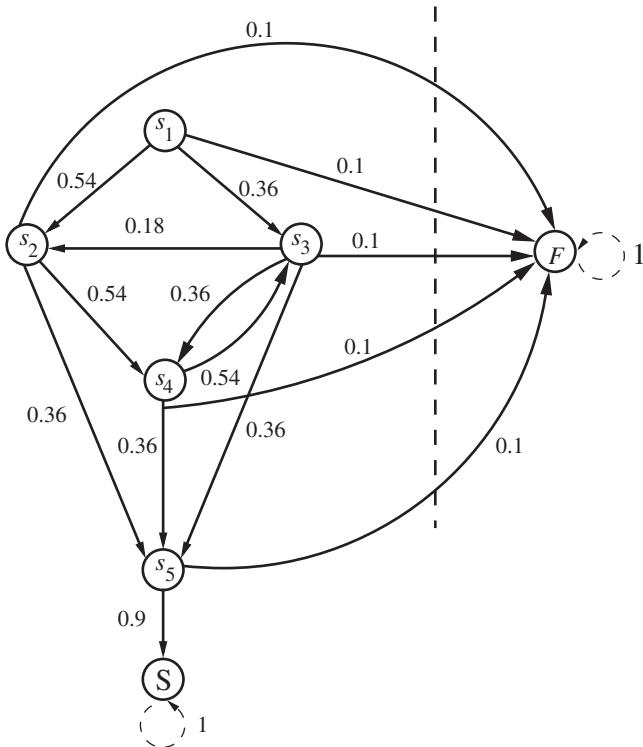


Figure 7.26. A program flow graph with failure state

added (see Figure 7.26), representing correct output and failure, respectively. The failure of a statement group s_j is considered by creating a directed edge to failure state F with transition probability $1 - R_j$ where R_j is the reliability of statement group s_j . We also need to modify transition probability from state s_j to state s_k to $R_j p_{jk}$, which represents the probability that statement group s_j produces the correct result and the control is transferred to statement group s_k . From the exit state s_5 , a directed edge to state S is created with transition probability R_5 to represent correct execution. In our numerical example, we assume that $R_j = 0.9$ for all j . With this modification, the transition probability matrix of the Markov chain is given by

$$P = \begin{bmatrix} & s_1 & s_2 & s_3 & s_4 & s_5 & S & F \\ s_1 & 0 & 0.54 & 0.36 & 0 & 0 & 0 & 0.1 \\ s_2 & 0 & 0 & 0 & 0.54 & 0.36 & 0 & 0.1 \\ s_3 & 0 & 0.18 & 0 & 0.36 & 0.36 & 0 & 0.1 \\ s_4 & 0 & 0 & 0.54 & 0 & 0.36 & 0 & 0.1 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0.1 \\ S & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now the matrix Q is

$$Q = \begin{bmatrix} 0 & 0.54 & 0.36 & 0 & 0 \\ 0 & 0 & 0 & 0.54 & 0.36 \\ 0 & 0.18 & 0 & 0.36 & 0.36 \\ 0 & 0 & 0.54 & 0 & 0.36 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The fundamental matrix M is computed to be

$$M = (I - Q)^{-1} = \begin{bmatrix} 1 & 0.6637 & 0.6871 & 0.6057 & 0.7043 \\ 0 & 1.0697 & 0.3872 & 0.7170 & 0.7826 \\ 0 & 0.2390 & 1.3278 & 0.6071 & 0.7826 \\ 0 & 0.1291 & 0.7170 & 1.3278 & 0.7826 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and matrix A is computed as

$$A = MC = \begin{bmatrix} 0.6339 & 0.3661 \\ 0.7043 & 0.2957 \\ 0.7043 & 0.2957 \\ 0.7043 & 0.2957 \\ 0.9 & 0.1 \end{bmatrix}.$$

Thus, if the program started in state s_1 , it will complete successfully with probability 0.6339 and it will fail with probability 0.3661.

Interestingly, the same result can also be obtained by applying the power method [equation (7.20)]. Suppose that we are interested in finding out the probability that the program completes successfully or fails starting from state s_1 . The initial probability vector $\mathbf{v}^{(0)} = [1, 0, 0, 0, 0, 0]$. We iterate using the following:

$$\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} P, \quad k = 1, 2, \dots,$$

until convergence is reached, that is, $|\mathbf{v}^{(k)} - \mathbf{v}^{(k-1)}| \leq \epsilon$, where ϵ is a very small positive real number. A simple Matlab or Mathematica script can give us the result after $k = 25$ iterations for $\epsilon = 10^{-6}$:

$$\mathbf{v}^{(k)} = [0, 0, 0, 0, 0, 0.6339, 0.3661].$$

Same procedure can be used for different starting states. To get the result in one single run, we may start the iteration with an initial probability matrix instead of a vector:

$$\mathbf{V}^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Applying the same iteration, after $k = 26$ iterations for the same ϵ , we have

$$\mathbf{V}^{(k)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.6339 & 0.3661 \\ 0 & 0 & 0 & 0 & 0 & 0.7043 & 0.2957 \\ 0 & 0 & 0 & 0 & 0 & 0.7043 & 0.2957 \\ 0 & 0 & 0 & 0 & 0 & 0.7043 & 0.2957 \\ 0 & 0 & 0 & 0 & 0 & 0.9000 & 0.1000 \end{bmatrix}.$$

The last two columns are the probabilities of interest, which are the same as those obtained using the fundamental matrix method.

‡

Problems

- Refer to Knuth [KNUT 1997]. Given the stochastic program flow graph shown in Figure 7.P.2, compute the average number of times each vertex s_i is visited, and assuming that the execution time of s_i is given by $t_i = 2i + 1$ time units, find the average total execution time τ of the program.

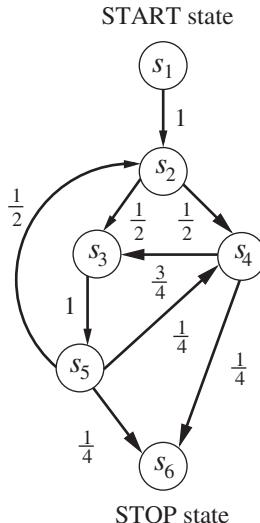


Figure 7.P.2. Another program flow graph

Review Problems

- * Refer to McCabe [MCCA 1965]. In Example 4.2 we stated that the average time to sequentially search a linear list is minimized if the keys are arranged in decreasing order of request probabilities. If the request probabilities are not known in advance, near-optimal behavior can still be achieved by using a self-organized list.

One such technique, known as a “move to front” heuristic, moves the requested key, if located, to the front of the list. Show that the average number of key comparisons needed for a successful search is given by

$$E[X] = 1 + \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n \frac{\alpha_i \alpha_j}{\alpha_i + \alpha_j},$$

where n is the number of distinct keys in the list and $0 < \alpha_i < 1$ is the probability of accessing the key labeled i (which may be located in any one of the n positions). You may proceed to show this result first in case $n = 3$ and then attempt to generalize it.

2. * Reconsider Example 7.16 and assume that an overflow forces the program to abort, but an underflow is simply ignored. Compute the average number of operations until an abort occurs. Simplify the problem by first considering a data structure with three nodes and then attempt to generalize your result.

REFERENCES

- [ABRA 1970] N. Abramson, “The Aloha system—another alternative for computer communications,” *Proc. Fall Joint Computer Conf.*, AFIPS Conf., 1970.
- [AJMO 1986] M. Ajmone-Marsan, G. Balbo, and G. Conte, *Performance Models of Multiprocessor Systems*, MIT Press, Cambridge, MA, 1986.
- [ASH 1970] R. B. Ash, *Basic Probability Theory*, Wiley, New York, 1970.
- [BERT 1992] D. Bertsekas and R. Gallager, *Data Networks*, 2 Ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [BHAT 1984] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2 Ed., Wiley, New York, 1984.
- [BOLC 1998] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains*, Wiley, New York, 1998
- [CHOI 1993] H. Choi, V. G. Kulkarni, and K. S. Trivedi, “Transient analysis of deterministic and stochastic Petri nets,” *Proc. 14th Int. Conf. Application and Theory of Petri Nets*, Chicago, IL, 1993, pp. 166–185.
- [CHOI 1994] H. Choi, V. G. Kulkarni, and K. S. Trivedi, “Markov regenerative stochastic Petri nets,” *Performance Evaluation*, **20**(1–3), 337–357 (1994).
- [CLAR 1970] A. B. Clarke and R. L. Disney, *Probability and Random Processes for Engineers and Scientists*, Wiley, New York, 1970.
- [FELL 1968] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vols. I, II, Wiley, New York, 1968.
- [GERM 2000] R. German, *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*, Wiley, New York, 2000.

- [GERM 1994] R. German and C. Lindemann, "Analysis of deterministic and stochastic Petri nets by the method of supplementary variables," *Proc. MASCOTS '95*, Durham, NC, 1994.
- [GOKH 1998] S. Gokhale, W. E. Wong, J. R. Horgan, and K. S. Trivedi, "An analytical approach to architecture-based software reliability prediction," *IEEE International Computer Performance and Dependability Symp.*, Durham, NC, Sept. 1998.
- [GOSE 2000] K. Goševa-Popstojanova and K. S. Trivedi, "Failure correlation in software reliability models," *IEEE Trans. Reliability*, **49**(1), 37–48 (2000).
- [GROS 1998] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 3rd ed., Wiley, New York, 1998.
- [KLEI 1975] L. Kleinrock, *Queueing Systems*, Vol. I, *Theory*, Wiley, New York, 1975.
- [KNUT 1997] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*, 3rd ed., Addison-Wesley, Reading, MA, 1997.
- [KULK 1995] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, Chapman & Hall, London, 1995.
- [LOGO 1994] D. Logothetis and K. S. Trivedi, "Transient analysis of the leaky bucket rate control scheme under Poisson and on-off sources", *Proc. Infocom '94*, Toronto, Canada, 1994.
- [LOGO 1995] D. Logothetis and K. S. Trivedi, "Time-dependent behavior of redundant systems with deterministic repair," in W. J. Stewart (ed.), *Proc. 2nd Int. Workshop Numerical Solution of Markov Chains*, Kluwer Academic Publishers, Amsterdam, 1995, pp. 135–150.
- [MCCA 1965] J. McCabe, "On Serial files with relocatable records," *Oper. Res.*, **13**, 609–618 (1965).
- [MEDH 1994] J. Medhi, *Stochastic Processes*, Wiley Eastern Limited, New Delhi, India, 1994.
- [MOLL 1989] M. K. Molloy, *Fundamentals of Performance Modeling*, Macmillan, New York, 1989.
- [ONNU 1993] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issue*, Artech House, Boston, 1993.
- [PARZ 1962] E. Parzen, *Stochastic Processes*, Holden-Day, San Francisco, CA, 1962.
- [RAO 1978] G. S. Rao, "Performance of cache memories," *J. ACM*, **25**(3), 378–385 (1978).
- [SHED 1972] G. S. Shedler and C. Tung, "Locality in page reference string," *SIAM J. Comput.*, **1**(3), 218–241 (1972).
- [SPIR 1977] J. R. Spirn, *Program Behavior: Models and Measurements*, Elsevier, New York, 1977.
- [STEW 1994] W. J. Stewart, *Introduction to Numerical Solution of Markov Chains*, Princeton Univ. Press, Princeton, NJ, 1994.
- [WOLF 1982] R. Wolff, "Poisson arrivals see time averages," *Oper. Res.*, **30**, 223–231 (1982).

Chapter 8

Continuous-Time Markov Chains

8.1 INTRODUCTION

The analysis of continuous-time Markov chains (CTMCs) is similar to that of the discrete-time case, except that the transitions from a given state to another state can take place at any instant of time. As in the last chapter, we confine our attention to **discrete-state** processes. This implies that, although the parameter t has a continuous range of values, the set of values of $X(t)$ is discrete. Let $I = \{0, 1, 2, \dots\}$ denote the state space of the process, and $T = [0, \infty)$ be its parameter space. Recalling from Chapter 6, a discrete-state continuous-time stochastic process $\{X(t) | t \geq 0\}$ is called a Markov chain if for $t_0 < t_1 < t_2 < \dots < t_n < t$, with t and $t_r \geq 0$ ($r = 0, 1, \dots, n$), its conditional pmf satisfies the relation

$$\begin{aligned} P(X(t) = x | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0) \\ = P(X(t) = x | X(t_n) = x_n). \end{aligned} \quad (8.1)$$

The behavior of the process is characterized by (1) the initial state probability vector of the CTMC given by the pmf of $X(t_0)$, i.e., $P(X(t_0) = k)$, $k = 0, 1, 2, \dots$, and (2) the transition probabilities:

$$p_{ij}(v, t) = P(X(t) = j | X(v) = i) \quad (8.2)$$

for $0 \leq v \leq t$ and $i, j = 0, 1, 2, \dots$, where

$$\sum_{j \in I} p_{ij}(v, t) = 1 \quad \text{for all } i; \quad 0 \leq v \leq t. \quad (8.3)$$

and where we define

$$p_{ij}(t, t) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

The Markov chain $\{X(t) | t \geq 0\}$ is said to be **(time-)homogeneous** (or is said to have **stationary transition probabilities**) if $p_{ij}(v, t)$ depends only on the time difference $(t - v)$. In this case, we abbreviate the notation for the transition probabilities as $p_{ij}(t - v)$. This conditional probability is written as:

$$p_{ij}(t) = P(X(t + v) = j | X(v) = i) \quad \text{for any } v \geq 0. \quad (8.4)$$

Let us denote the pmf of $X(t)$ (or the **state probabilities** at time t) by

$$\pi_j(t) = P(X(t) = j), \quad j = 0, 1, 2, \dots; t \geq 0. \quad (8.5)$$

It is clear that

$$\sum_{j \in I} \pi_j(t) = 1$$

for any $t \geq 0$, since at any given time the process must be in *some* state.

By using the theorem of total probability, for given $t > v$, we can express the pmf of $X(t)$ in terms of the transition probabilities $p_{ij}(v, t)$ and the pmf of $X(v)$:

$$\begin{aligned} \pi_j(t) &= P(X(t) = j) \\ &= \sum_{i \in I} P(X(t) = j | X(v) = i) P(X(v) = i) \\ &= \sum_{i \in I} p_{ij}(v, t) \pi_i(v). \end{aligned} \quad (8.6)$$

If we let $v = 0$ in (8.6), then

$$\pi_j(t) = \sum_{i \in I} p_{ij}(0, t) \pi_i(0). \quad (8.7)$$

Hence, the probabilistic behavior of a CTMC is completely determined once the transition probabilities $p_{ij}(v, t)$ and the initial probability vector $\boldsymbol{\pi}(0) = [\pi_0(0), \pi_1(0), \dots]$ are specified.

The transition probabilities of a CTMC $\{X(t) | t \geq 0\}$ satisfy the **Chapman–Kolmogorov equation** which states that for all $i, j \in I$,

$$p_{ij}(v, t) = \sum_{k \in I} p_{ik}(v, u) p_{kj}(u, t), \quad 0 \leq v < u < t. \quad (8.8)$$

To prove (8.8), we use the theorem of total probability:

$$\begin{aligned} P(X(t) = j \mid X(v) = i) &= \sum_{k \in I} P(X(t) = j \mid X(u) = k, X(v) = i) \\ &\quad \cdot P(X(u) = k \mid X(v) = i). \end{aligned}$$

The subsequent application of the Markov property (8.1) yields (8.8).

The direct use of (8.8) is difficult. Usually we obtain the transition probabilities by solving a system of differential equations that we derive next. For this purpose, under certain regularity conditions, we can show that for each j there is a nonnegative continuous function $q_j(t)$ defined by

$$\begin{aligned} q_j(t) &= -\frac{\partial}{\partial t} p_{jj}(v, t)|_{v=t} \\ &= \lim_{h \rightarrow 0} \frac{p_{jj}(t, t) - p_{jj}(t, t+h)}{h} = \lim_{h \rightarrow 0} \frac{1 - p_{jj}(t, t+h)}{h}. \end{aligned} \quad (8.9)$$

Similarly, for each i and j ($\neq i$) there is a nonnegative continuous function $q_{ij}(t)$ (known as the transition rate from state i to state j at time t) defined by

$$\begin{aligned} q_{ij}(t) &= \frac{\partial}{\partial t} p_{ij}(v, t)|_{v=t} \\ &= \lim_{h \rightarrow 0} \frac{p_{ij}(t, t) - p_{ij}(t, t+h)}{-h} = \lim_{h \rightarrow 0} \frac{p_{ij}(t, t+h)}{h}. \end{aligned} \quad (8.10)$$

Then the transition probabilities and the transition rates are related by¹

$$p_{ij}(t, t+h) = q_{ij}(t) \cdot h + o(h), \quad i \neq j,$$

and

$$p_{jj}(t, t+h) = 1 - q_j(t) \cdot h + o(h), \quad i = j.$$

Substituting $t+h$ for t in equation (8.8), we get

$$p_{ij}(v, t+h) = \sum_k p_{ik}(v, u)p_{kj}(u, t+h)$$

which implies

$$p_{ij}(v, t+h) - p_{ij}(v, t) = \sum_k p_{ik}(v, u)[p_{kj}(u, t+h) - p_{kj}(u, t)].$$

¹ $o(h)$ is any function of h that approaches zero faster than h :

$$\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$$

Dividing both sides by h and taking the limit $h \rightarrow 0$ and $u \rightarrow t$, we get the differential equation known as **Kolmogorov's forward equation**. For $0 \leq v < t$ and $i, j \in I$

$$\frac{\partial p_{ij}(v, t)}{\partial t} = \left[\sum_{k \neq j} p_{ik}(v, t) q_{kj}(t) \right] - p_{ij}(v, t) q_j(t). \quad (8.11)$$

In a similar fashion we can also derive **Kolmogorov's backward equation**:

$$\frac{\partial p_{ij}(v, t)}{\partial v} = \left[\sum_{k \neq i} p_{kj}(v, t) q_{ik}(v) \right] - p_{ij}(v, t) q_i(v). \quad (8.12)$$

Define the infinitesimal generator matrix $Q(t) = [q_{ij}(t)]$ with the diagonal entries $q_{ii}(t) = -q_i(t)$. It is easy to see that $\sum_j q_{ij}(t) = 0$ for all i . Now if we define the matrix $P(v, t) = [p_{ij}(v, t)]$, we can write these equations in matrix form:

$$\frac{\partial P(v, t)}{\partial t} = P(v, t) Q(t),$$

$$\frac{\partial P(v, t)}{\partial v} = Q(v) P(v, t).$$

Using (8.6) and (8.11) we can also derive a differential equation for the unconditional probability $\pi_j(t)$ as

$$\frac{d\pi_j(t)}{dt} = \left[\sum_{i \neq j} \pi_i(t) q_{ij}(t) \right] - \pi_j(t) q_j(t). \quad (8.13)$$

We use (8.11) when we want specifically to show the initial state, while we use (8.13) when the initial state (or initial probability vector) is implied. If we let the $\boldsymbol{\pi}(t) = [\pi_0(t), \pi_1(t), \dots]$, then in matrix form we have

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t) Q(t). \quad (8.14)$$

In many important applications the transition probabilities $p_{ii}(t, t+h)$ do not depend on the initial time t but only on the elapsed time h (i.e., the resulting Markov chain is **time-homogeneous** or simply homogeneous). This implies that the transition rates $q_{ij}(t)$ and $q_j(t)$ are independent of t . However, when $q_j(t)$ are time dependent, then the resulting Markov chain is said to be **nonhomogeneous**. Unless otherwise stated, we will be concerned only with

time-homogeneous situations. In this case the transition rates² are denoted by q_{ij} and the transition probabilities $p_{ij}(t, t + h)$ by $p_{ij}(h)$. Equations (8.11) and (8.13) are rewritten as

$$\frac{dp_{ij}(t)}{dt} = \left[\sum_{k \neq j} p_{ik}(t)q_{kj} \right] - p_{ij}(t)q_j, \quad (8.15)$$

$$\frac{d\pi_j(t)}{dt} = \sum_{i \neq j} \pi_i(t)q_{ij} - \pi_j(t)q_j. \quad (8.16)$$

In matrix form the equations are

$$\frac{dP(t)}{dt} = P(t)Q, \quad (8.17)$$

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)Q. \quad (8.18)$$

The infinitesimal generator matrix Q of a homogeneous CTMC has time-independent entries. Matrix Q will sometimes be called the *generator matrix* of a CTMC for short.

Define $L_j(t) = \int_0^t \pi_j(x)dx$. It can be shown that $L_j(t)$ is the time spent by the CTMC in state j during the interval $(0, t]$. Let the vector $\mathbf{L}(t) = [L_j(t)]$. Then by integration of (8.18), we get

$$\frac{d\mathbf{L}(t)}{dt} = \mathbf{L}(t)Q + \boldsymbol{\pi}(0). \quad (8.19)$$

Even in this simpler case of a time-homogeneous Markov chain, solution of equation (8.18) to obtain the time-dependent probabilities $\pi_j(t)$ in closed form is quite difficult. We will consider several special cases where closed-form solution is possible and we will also consider methods of numerical solution for the more general case. Nevertheless, in many interesting situations a further reduction is possible in that the probabilities $\pi_j(t)$ approach a limit π_j as t approaches infinity. We wish to explore the conditions under which such a limiting probability vector exists.

A classification of states for a CTMC is similar to the discrete-time case except that there is no notion of periodic/aperiodic state in CTMC. A state i is said to be an **absorbing state** provided that $q_{ij} = 0$ for all $j \neq i$, so that, once entered, the process is destined to remain in that state. For a CTMC

²It is to be noted that $q_{ij}, i \neq j$ is always finite. While $q_j (\geq 0)$ always exists and is finite when I is finite, q_j may be infinite when I is denumerably infinite.

with two or more absorbing states, the limiting probabilities $\lim_{t \rightarrow \infty} p_{ij}(t)$ may well depend on the initial state.

A state j is said to be **reachable** from state i if for some $t > 0$, $p_{ij}(t) > 0$. A CTMC is said to be **irreducible** if every state is reachable from every other state.

THEOREM 8.1. For an irreducible continuous-time Markov chain, the limits

$$\pi_j = \lim_{t \rightarrow \infty} p_{ij}(t) = \lim_{t \rightarrow \infty} \pi_j(t) \quad i, j \in I \quad (8.20)$$

always exist and are independent of the initial state i .

If the limiting probabilities π_j exist, then

$$\lim_{t \rightarrow \infty} \frac{d\pi_j(t)}{dt} = 0, \quad (8.21)$$

and, substituting into equation (8.18), we get the following system of linear homogeneous equations (one for each state j):

$$0 = \sum_{i \neq j} \pi_i q_{ij} - \pi_j q_j. \quad (8.22)$$

If we define the steady-state probability vector $\boldsymbol{\pi} = [\pi_0, \pi_1, \dots]$, then in vector-matrix form equation (8.22) becomes

$$\boldsymbol{\pi}Q = 0. \quad (8.23)$$

This is the continuous analog of equation (7.18).

For the homogeneous system of equations, one possible solution is that $\pi_j = 0$ for all j . If another solution exists, then an infinite number of solutions can be obtained by multiplying by scalars. To determine a nonzero unique solution, we use the following condition:

$$\sum_j \pi_j = 1. \quad (8.24)$$

Irreducible Markov chains that yield positive limiting probabilities $\{\pi_j\}$ in this way are called **recurrent nonnull** or **positive recurrent**, and the probabilities $\{\pi_j\}$, satisfying (8.22) and (8.24) are also known as **steady-state probabilities**. It is clear that a finite irreducible Markov chain must be positive recurrent; hence we can obtain its unique limiting probabilities by solving the finite system of equations (8.22) under the condition (8.24). More

generally, the states of finite Markov chain can be partitioned into subsets C_1, C_2, \dots, C_k such that C_k consists of all transient states and for $i = 1, 2, \dots, k-1$, C_i is a closed set of recurrent nonnull states. If C_i contains only one state then that state is an absorbing state.

Once state probabilities, π_j and $\pi_j(t)$ (or integrals $L_j(t)$) have been computed, measures of interest are usually obtained as weighted averages of these quantities. Assume a weight or a reward rate r_j is attached to state j . Let $Z(t) = r_{X(t)}$ be the reward rate of the CTMC at time t . Then the expected instantaneous reward rate at time t is

$$E[Z(t)] = \sum_j r_j \pi_j(t). \quad (8.25)$$

For an irreducible CTMC, the expected steady-state reward rate is

$$E[Z] = \lim_{t \rightarrow \infty} E[Z(t)] = \sum_j r_j \pi_j. \quad (8.26)$$

Define $Y(t) = \int_0^t Z(\tau) d\tau$ as the accumulated reward in the interval $(0, t]$. Then the expected accumulated reward in the interval $(0, t]$ is given by

$$E[Y(t)] = \sum_j r_j \int_0^t \pi_j(\tau) d\tau = \sum_j r_j L_j(t). \quad (8.27)$$

We have seen in Chapter 6 that the distribution of times that a homogeneous CTMC spends in a given state must be memoryless. This implies that holding times in a state of a CTMC of the homogeneous type are exponentially distributed. In the next section we study the limiting probability vector of a special type of Markov chain, called the *birth-death process*. In Section 8.4, we study limiting probability vector of several non-birth-death processes.

The study of transient behavior $[\pi_j(t), t \geq 0]$ is quite complex for a general Markov chain. In Sections 8.3 and 8.5 we consider special cases where it is possible to obtain an explicit solution for $\pi_j(t)$, while in Section 8.6, we will briefly review solution techniques for homogeneous CTMCs. The automated generation of CTMCs is discussed in Section 8.7.

Problems

1. * Show that the solution to the matrix equation (8.17) with the initial condition $P(0) = I$ can be written as the matrix exponential:

$$P(t) = e^{Qt} = I + \sum_{n=1}^{\infty} Q^n \frac{t^n}{n!},$$

assuming that matrix series converges. Generalize this result to the case of a nonhomogeneous CTMC.

2. Show that the solution to the matrix–vector equation (8.18) can be written as

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)e^{Qt}.$$

3. Show that the solution to the matrix–vector equation (8.19) can be written as

$$\mathbf{L}(t)Q = \boldsymbol{\pi}(0)[e^{Qt} - I].$$

4. Show that the solution to equation (8.14) for a nonhomogeneous CTMC,

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)e^{\int_0^t Q(x)dx}.$$

5. * For a homogeneous CTMC show that the Laplace transform of the transition probability matrix $P(t)$, denoted by $\overline{P}(s)$, is given by

$$\overline{P}(s) = (sI - Q)^{-1}.$$

6. * Show that the integral (convolution) form of the Kolmogorov forward equation is given by

$$p_{ij}(v, t) = \delta_{ij} e^{-\int_v^t q_{ii}(\tau)d\tau} + \int_v^t \sum_k p_{ik}(v, x)q_{kj}(x)e^{-\int_x^t q_{jj}(\tau)d\tau} dx,$$

where δ_{ij} is the Kronecker delta function defined by $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Specialize this result to the case of a homogeneous CTMC.

7. * Show that $\gamma_0 = 0$ is an eigenvalue of the generator matrix Q .

8.2 THE BIRTH–DEATH PROCESS

A continuous-time homogeneous Markov chain $\{X(t) \mid t \geq 0\}$ with the state space $\{0, 1, 2, \dots\}$ is known as a **birth–death process** if there exist constants λ_i ($i = 0, 1, \dots$) and μ_i ($i = 1, 2, \dots$) such that the transition rates are given by

$$q_{i,i+1} = \lambda_i,$$

$$q_{i,i-1} = \mu_i,$$

$$q_i = \lambda_i + \mu_i,$$

$$q_{ij} = 0 \quad \text{for } |i - j| > 1.$$

The **birth rate** λ_i (≥ 0) is the rate at which births occur in state i , and the **death rate** μ_i (≥ 0) is the rate at which deaths occur in state i . These rates are assumed to depend only on state i and are independent of time. Note that only “nearest-neighbor transitions” are allowed. In a given state, births and

deaths occur independently of each other. Such a process is a useful model of many situations in queuing theory and reliability theory.

The CTMC will be in state k at time $t + h$ if one of the following mutually exclusive and collectively exhaustive events occurs:

1. The CTMC is in state k at time t , and no changes of state occur in the interval $(t, t + h]$; the associated conditional probability is

$$p_{k,k}(t, t + h) = 1 - q_k \cdot h + o(h) = 1 - (\lambda_k + \mu_k) \cdot h + o(h).$$

2. The CTMC is in state $k - 1$ at time t , and one birth occurs in the interval $(t, t + h]$; the associated conditional probability is

$$p_{k-1,k}(t, t + h) = q_{k-1,k} \cdot h + o(h) = \lambda_{k-1} \cdot h + o(h).$$

3. The CTMC is in state $k + 1$ at time t , and one death occurs in the interval $(t, t + h]$; the associated conditional probability is

$$p_{k+1,k}(t, t + h) = q_{k+1,k} \cdot h + o(h) = \mu_{k+1} \cdot h + o(h).$$

4. Two or more transitions occur in the interval $(t, t + h]$, resulting in $X(t + h) = k$, with associated conditional probability $o(h)$.

Then, by the theorem of total probability, we have

$$\begin{aligned} P(X(t + h) = k) &= \pi_k(t + h) \\ &= \pi_k(t)p_{k,k}(t, t + h) + \pi_{k-1}(t)p_{k-1,k}(t, t + h) \\ &\quad + \pi_{k+1}(t)p_{k+1,k}(t, t + h) + o(h). \end{aligned}$$

After rearranging, dividing by h , and taking the limit as $h \rightarrow 0$, we get

$$\begin{aligned} \frac{d\pi_k(t)}{dt} &= -(\lambda_k + \mu_k)\pi_k(t) + \lambda_{k-1}\pi_{k-1}(t) + \mu_{k+1}\pi_{k+1}(t), \quad k \geq 1, \\ \frac{d\pi_0(t)}{dt} &= -\lambda_0\pi_0(t) + \mu_1\pi_1(t), \quad k = 0, \end{aligned} \tag{8.28}$$

where the special equation for $k = 0$ is required because the state space of the process is assumed to be $\{0, 1, 2, \dots\}$. Equation (8.28) is a special case of equation (8.13), with $q_{k-1,k} = \lambda_{k-1}$, $q_{k+1,k} = \mu_{k+1}$, and $q_k = (\lambda_k + \mu_k)$. The corresponding generator matrix Q is tridiagonal.

The solution of this system of differential-difference equations is a formidable task. However, if we are not interested in the transient behavior, then we can set the derivative $d\pi_k(t)/dt$ equal to zero, and the resulting

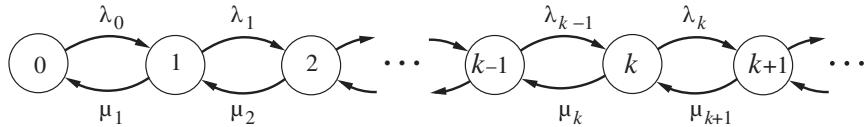


Figure 8.1. The state diagram of the birth–death process

set of difference equations provide the steady-state solution of the CTMC. Let π_k denote the steady-state probability that the chain is in state k , that is, $\pi_k = \lim_{t \rightarrow \infty} \pi_k(t)$ (assuming that it exists). Then the above differential–difference equations reduce to [a special case of equation (8.22)]

$$0 = -(\lambda_k + \mu_k)\pi_k + \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1}, \quad k \geq 1, \quad (8.29)$$

$$0 = -\lambda_0\pi_0 + \mu_1\pi_1. \quad (8.30)$$

These are known as the **balance equations**, and we can obtain them directly from the state diagram, shown in Figure 8.1, by equating the rates of flow into and out of each state. From the state diagram we have the rate of transition into state k as $\lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1}$ and the rate of transition out of state k as $(\lambda_k + \mu_k)\pi_k$. In the steady state no buildup occurs in state k ; hence these two rates must be equal.

We should note the difference between this state diagram (of a continuous-time Markov chain) and the state diagram of a discrete-time Markov chain (Chapter 7). In the latter, the arcs are labeled with conditional probabilities; in the former they are labeled with state transition rates (hence the term *transition-rate diagram* is sometimes used).

By rearranging equation (8.29), we get

$$\lambda_k\pi_k - \mu_{k+1}\pi_{k+1} = \lambda_{k-1}\pi_{k-1} - \mu_k\pi_k = \cdots = \lambda_0\pi_0 - \mu_1\pi_1.$$

But from equation (8.30) we have $\lambda_0\pi_0 - \mu_1\pi_1 = 0$. It follows that

$$\lambda_{k-1}\pi_{k-1} - \mu_k\pi_k = 0$$

and hence

$$\pi_k = \frac{\lambda_{k-1}}{\mu_k}\pi_{k-1}, \quad k \geq 1.$$

Therefore

$$\pi_k = \frac{\lambda_0\lambda_1 \cdots \lambda_{k-1}}{\mu_1\mu_2 \cdots \mu_k} \pi_0 = \pi_0 \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right), \quad k \geq 1. \quad (8.31)$$

Since $\sum_{k \geq 0} \pi_k = 1$, we have

$$\pi_0 = \frac{1}{1 + \sum_{k \geq 1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right)}. \quad (8.32)$$

Thus, the limiting state probability vector $[\pi_0, \pi_1, \dots]$ is now completely determined. Note that the limiting probabilities are nonzero, provided that the series

$$\sum_{k \geq 1} \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right)$$

converges (in which case, all the states of the Markov chain are recurrent nonnull).

When the state space of CTMC is finite (i.e., $I = \{0, 1, \dots, n\}$), the corresponding difference equations become

$$\begin{aligned} 0 &= -\lambda_0 \pi_0 + \mu_1 \pi_1 \\ 0 &= -(\lambda_k + \mu_k) \pi_k + \lambda_{k-1} \pi_{k-1} + \mu_{k+1} \pi_{k+1}, \quad 1 \leq k \leq n-1 \\ 0 &= -\mu_n \pi_n + \lambda_{n-1} \pi_{n-1}. \end{aligned}$$

By a method similar to the one we used in the case of infinite state space, we have the limiting probabilities

$$\pi_k = \pi_0 \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right), \quad k = 1, 2, \dots, n \quad (8.33)$$

where

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^n \prod_{i=0}^{k-1} \left(\frac{\lambda_i}{\mu_{i+1}} \right)}.$$

We note that there is no additional condition for convergence in the case of a finite CTMC as a finite state irreducible CTMC is always positive recurrent.

Next we consider several special cases of the birth-death process.

8.2.1 The $M/M/1$ Queue

We consider a single-server Markovian queue shown in Figure 8.2. Customer arrivals form a Poisson process with rate λ . Equivalently the customer interarrival times are exponentially distributed with mean $1/\lambda$. Service times of customers are independent identically distributed random variables, the common

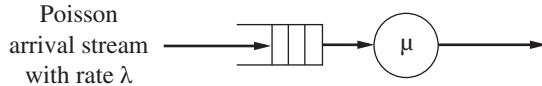


Figure 8.2. The $M/M/1$ queuing system

distribution being exponential with mean $1/\mu$. Assume that customers are served in their order of arrival (FCFS scheduling). If the “customer” denotes a job arriving into a computer system, then the server represents the computer system. [Since most computer systems consist of a set of interacting resources (and hence a network of queues), such a simple representation may be acceptable for “small” systems with little concurrency.] In another interpretation of the $M/M/1$ queue, the customer may represent a message and the server a communication channel.

Let $N(t)$ denote the number of customers in the system (those queued plus the one in service) at time t . [We change the notation from $X(t)$ to $N(t)$ to conform to standard practice.] Then $\{N(t) \mid t \geq 0\}$ is a birth–death process with

$$\lambda_k = \lambda, \quad k \geq 0; \quad \mu_k = \mu, \quad k \geq 1.$$

The ratio $\rho = \lambda/\mu$ = mean service time/mean interarrival time, is an important parameter, called the **traffic intensity** of the system. The traffic intensity is a dimensionless quantity but in teletraffic theory, it is often quoted in units known as **Erlangs**. Equations (8.31) and (8.32) in this case reduce to

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0 = \rho^k \pi_0$$

and

$$\pi_0 = \frac{1}{\sum_{k \geq 0} \rho^k} = 1 - \rho,$$

provided $\rho < 1$, that is, when the traffic intensity is less than unity. In the case that the arrival rate λ exceeds the service rate μ (i.e., $\rho \geq 1$), the geometric series in the denominator of the expression for π_0 diverges. In the case that $\rho > 1$, all the states of the CTMC are transient and if $\rho = 1$, all the states of the CTMC are null recurrent [KLEI 1975]. Hence in both these cases, the number of customers in the system tends to increase without bound. Such a system is called **unstable**. For a stable system ($\rho < 1$), the steady-state probabilities form a modified geometric pmf with parameter $1 - \rho$:

$$\pi_k = (1 - \rho)\rho^k, \quad k \geq 0. \tag{8.34}$$

The server utilization, $U_0 = 1 - \pi_0 = \rho$, is interpreted as the proportion of time the server is busy.

The mean and variance of the number of customers in the system are obtained using the properties of the modified geometric distribution as

$$E[N] = \sum_{k=0}^{\infty} k\pi_k = \frac{\rho}{1-\rho} \quad (8.35)$$

and

$$\text{Var}[N] = \sum_{k=0}^{\infty} k^2\pi_k - (E[N])^2 = \sum_{k=0}^{\infty} (k^2 - (E[N])^2)\pi_k = \frac{\rho}{(1-\rho)^2}. \quad (8.36)$$

Note that both these measures are expressed as weighted averages of steady-state probabilities. By attaching a suitably chosen set of weights $\{r_k\}$ to the states of the CTMC we can get most measures of interest expressed as the weighted average $\sum_{k=0}^{\infty} r_k\pi_k$ [see equation (8.26)]. The resulting CTMC will be known as a *Markov reward model* (MRM). The above measure of interest will then be known as the *expected reward rate in the steady state*.

Let the random variable R denote the response time (defined as the time elapsed from the instant of job arrival until its completion) in the steady state. In order to compute the average response time $E[R]$ we use the well-known **Little's formula**, which states that the mean number of jobs in a queuing system in the steady state is equal to the product of the arrival rate and the mean response time. When applied to the present case, Little's formula gives us

$$E[N] = \lambda E[R];$$

hence

$$E[R] = \frac{E[N]}{\lambda} = \sum_{k=0}^{\infty} \frac{k}{\lambda} \pi_k. \quad (8.37)$$

Little's formula holds for a broad variety of queuing systems. For a proof see [STID 1974], and for its limitations see [BEUT 1980].

Using (8.35) and applying Little's formula to the present case, we have

$$E[R] = \lambda^{-1} \frac{\rho}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{\text{average service time}}{\text{probability that the server is idle}}. \quad (8.38)$$

Note that the congestion in the system, and hence the delay, build rapidly as the traffic intensity increases (see Figures 8.3 and 8.4).

We may often employ a scheduling discipline other than FCFS. We distinguish between preemptive and nonpreemptive scheduling disciplines. A **nonpreemptive discipline** such as FCFS allows a job to complete execution once scheduled, whereas a **preemptive discipline** may interrupt the currently executing job in order to give preferential service to another job.

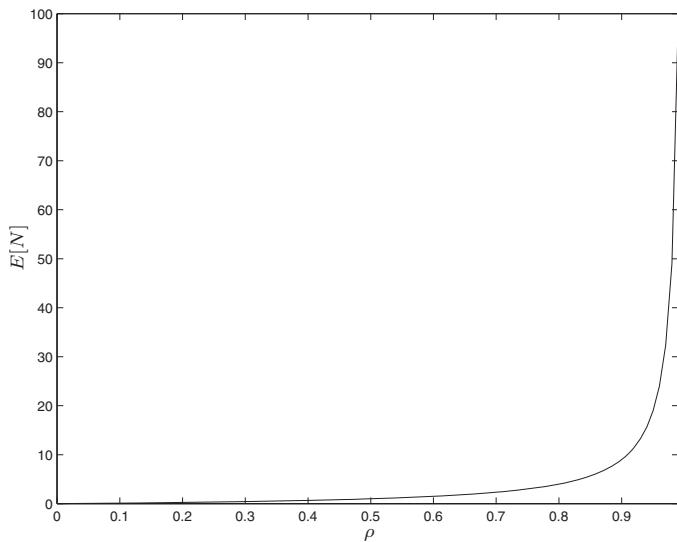


Figure 8.3. Expected number of jobs in system versus traffic intensity

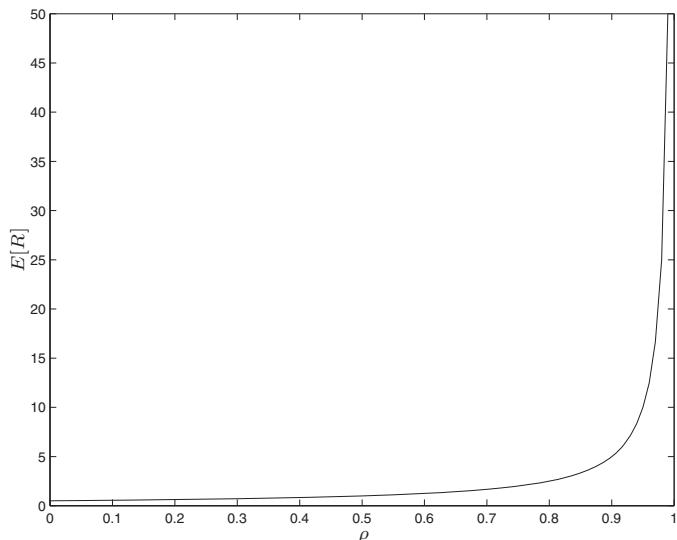


Figure 8.4. Average response time versus traffic intensity

A common example of a preemptive discipline is RR (round robin), which permits a job to remain in service for an interval of time referred to as its **quantum**. If the job does not finish execution within the quantum, it has to return to the end of the queue, awaiting further service. This gives preferential treatment to short jobs at the expense of long jobs. When the time quantum approaches zero, the RR discipline is known as the PS (processor sharing) discipline.

Although we assumed FCFS scheduling discipline in deriving formula (8.34) for the queue-length pmf and formula (8.38) for the average response time, they hold for any scheduling discipline that satisfies the following conditions [KOBA 1978]:

1. The server is not idle when there are jobs waiting for service.
2. The scheduler is not allowed to use any deterministic a priori information about job service times. Thus, for instance, if all job service times are known in advance, the use of a discipline known as SRPT (shortest remaining processing time first) is known to reduce $E[R]$ below that given by (8.38).
3. The service time distribution is not affected by the scheduling discipline.

Formulas (8.34) and (8.38) also apply for preemptive scheduling disciplines such as RR and PS, provided the overhead of preemption can be neglected (otherwise condition 3 above will be violated). We have also assumed in the above that a job is not allowed to leave the system before completion (see problems 3 and 4 below for exceptions).

Although the expression for the average response time (8.38) holds under a large class of scheduling disciplines, the distribution of the response time does depend on the scheduling discipline. We shall derive the distribution function of the response time R in the steady-state assuming the FCFS scheduling discipline. If an arriving job finds n jobs in the system, then the response time is the sum of $n + 1$ random variables, $S + S'_1 + S_2 + \dots + S_n$. Here S is the service time of the tagged job, S'_1 is the remaining service time of the job undergoing service, and S_2, \dots, S_n denote the service times of $(n - 1)$ jobs waiting in the queue. By our assumptions and the memoryless property of the exponential distribution, these $(n + 1)$ random variables are independent and exponentially distributed with parameter μ . Thus, the conditional Laplace–Stieltjes transform of R given $N = n$ is the convolution:

$$L_{R|N}(s | n) = \left(\frac{\mu}{s + \mu} \right)^{n+1}. \quad (8.39)$$

Wolff [WOLF 1982] shows that the pmf of the number of jobs in the system as seen by an arriving job is the same as that given by (8.34). Then, applying

the theorem of total Laplace transform, we obtain

$$\begin{aligned}
 L_R(s) &= \sum_{n=0}^{\infty} \left(\frac{\mu}{s+\mu}\right)^{n+1} (1-\rho)\rho^n \\
 &= \frac{\mu(1-\rho)}{s+\mu} \frac{1}{1 - \frac{\mu\rho}{s+\mu}} \\
 &= \frac{\mu(1-\rho)}{s+\mu(1-\rho)}. \tag{8.40}
 \end{aligned}$$

It follows that the steady-state response time R is exponentially distributed with parameter $\mu(1-\rho)$. Note that $L_R(s)$ is also expressed as the expected steady-state reward rate with $r_n = [\mu/(s+\mu)]^{n+1}$.

Other measures of system performance are easily obtained. Let the random variable W denote the waiting time in the queue; that is, let

$$W = R - S. \tag{8.41}$$

Then

$$E[W] = E[R] - E[S] = E[R] - \frac{1}{\mu}.$$

It follows that the average waiting time is given by

$$E[W] = \frac{1}{\mu(1-\rho)} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}. \tag{8.42}$$

If we now let the random variable Q denote the number of jobs waiting in the queue (excluding those, if any, in service), then, to determine the average number of jobs $E[Q]$ in the queue, we apply Little's formula to the queue excluding the server to obtain

$$E[Q] = \lambda E[W] = \frac{\rho^2}{1-\rho}. \tag{8.43}$$

Note that the average number of jobs found in the server is

$$E[N] - E[Q] = \rho. \tag{8.44}$$

Example 8.1

The capacity of a wireless communication channel is 20,000 bits per second (bps). This channel is used to transmit 8-bit characters, so the maximum rate is 2500 characters per second (cps). The application calls for traffic from many devices to

be sent on the channel with a total volume of 120,000 characters per minute. In this case

$$\lambda = \frac{120,000}{60} = 2,000 \text{ cps}, \quad \mu = 2,500 \text{ cps},$$

and channel utilization $\rho = \lambda/\mu = 4/5 = 0.8$.

The average number of characters waiting to be transmitted is $E[Q] = (0.8 \times 0.8)/(1 - 0.8) = 3.2$, and the average transmission (including queuing delay) time per character is $E[N]/\lambda = 4/2000\text{s} = 2 \text{ ms}$. ‡

Example 8.2

We wish to determine the maximum call rate that can be supported by one telephone booth. Assume that the mean duration of a telephone conversation is 3 min, and that no more than a 3-min (average) wait for the phone may be tolerated; what is the largest amount of incoming traffic that can be supported?

1. $\mu = \frac{1}{3}$ calls per minute; therefore, λ must be less than $\frac{1}{3}$ calls per minute, for the system to be stable.
2. The average waiting time $E[W]$ should be no more than 3 min; that is:

$$E[W] = \frac{\rho}{\mu(1-\rho)} \leq 3,$$

and since $\mu = \frac{1}{3}$, we get

$$1 - \rho \geq \rho$$

or

$$\rho \leq \frac{1}{2}.$$

Therefore, the call arrival rate is given by

$$\lambda \leq \frac{1}{6} \text{ calls per minute.}$$
‡

Problems

1. Consider an $M/M/1$ queue with an arrival rate λ and the service rate μ . We have derived the distribution function of the response time R . Now we are interested in deriving the distribution function of the waiting time W . The waiting time W is the response time minus the service time. To get started, first compute the conditional distribution of W conditioned on the number of jobs in the system, and later compute the unconditional distribution function. Note that W is a mixed random variable since its distribution function has a jump equal to $P(W = 0)$ at the origin.

2. A group of telephone subscribers is observed continuously during a 80-min busy-hour period. During this time they make 30 calls, and the total conversation time is 4200 s. Estimate the call arrival rate and the traffic intensity.
3. Consider an $M/M/1$ queuing system in which the total number of jobs is limited to n owing to a limitation on queue size.
 - (a) Find the steady state probability that an arriving request is rejected because the queue is full.
 - (b) Find the steady-state probability that the processor is idle.
 - (c) Find the throughput of the system in the steady state.
 - (d) Given that a request has been accepted, find its average response time.
4. The arrival of large jobs at a server forms a Poisson process with rate two per hour. The service times of such jobs are exponentially distributed with mean 20 min. Only four large jobs can be accommodated in the system at a time. Assuming that the fraction of computing power utilized by smaller jobs is negligible, determine the probability that a large job will be turned away because of lack of storage space.
5. Let the random variable T_k denote the holding time in state k of the $M/M/1$ queue. Starting from the given assumptions on the interarrival time and the service time distributions, show that the distribution of T_k is exponential (for each k).
6. Derive an expression for the frequency of entering state 0 (server idle) in an $M/M/1$ queue. This quantity is useful in estimating the overhead of scheduling. Plot this frequency as a function of ρ for a fixed μ .
7. Define the perceived mean queue length [GEIS 1983] $N^* = E[N^2]/E[N]$ and the perceived mean response time by $R^* = E[R^2]/E[R]$. Derive formulae for N^* and R^* in the $M/M/1$ queue.

8.2.2 The $M/M/m$ Queue

Consider a queuing system with arrival rate λ as before, but where $m \geq 1$ servers, each with a service rate μ , share a common queue (see Figure 8.5).

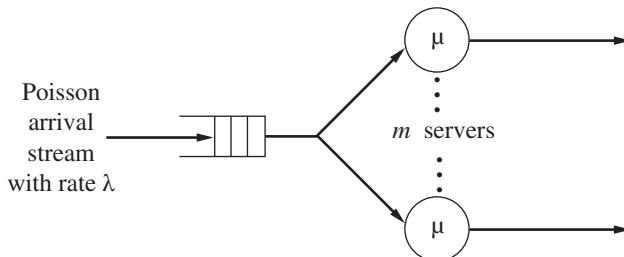


Figure 8.5. The $M/M/m$ queuing system

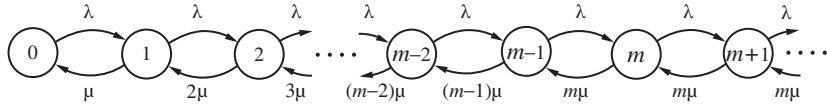


Figure 8.6. The state diagram of the $M/M/m$ queue

This gives rise to a birth-death model with the following rates:

$$\lambda_k = \lambda, \quad k = 0, 1, 2, \dots,$$

$$\mu_k = \begin{cases} k\mu, & 0 < k < m, \\ m\mu, & k \geq m. \end{cases}$$

The state diagram of this system is shown in Figure 8.6. The steady-state probabilities are given by [using equation (8.31)]

$$\begin{aligned} \pi_k &= \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} \\ &= \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}, \quad k < m, \\ \pi_k &= \pi_0 \prod_{i=0}^{m-1} \frac{\lambda}{(i+1)\mu} \prod_{j=m}^{k-1} \frac{\lambda}{m\mu} \\ &= \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{m!m^{k-m}}, \quad k \geq m. \end{aligned} \tag{8.45}$$

Defining $\rho = \lambda/(m\mu)$, the condition for stability is given by $\rho < 1$. The expression for π_0 is obtained using (8.45) and the fact that $\sum_{k=0}^{\infty} \pi_k = 1$:

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}. \tag{8.46}$$

The expression for the average number of jobs in the system is (see problem 5 at the end of this section) obtained as the expected reward rate in the steady state after assigning reward rate $r_k = k$ to state k :

$$E[N] = \sum_{k \geq 0} k\pi_k = m\rho + \rho \frac{(m\rho)^m}{m!} \frac{\pi_0}{(1-\rho)^2}. \tag{8.47}$$

Let the random variable M denote the number of busy servers; then

$$P(M = k) = \begin{cases} P(N = k) = \pi_k, & 0 \leq k \leq m - 1, \\ P(N \geq m) = \sum_{k=m}^{\infty} \pi_k = \frac{\pi_m}{1 - \rho}, & k = m. \end{cases}$$

The average number of busy servers is then

$$E[M] = \sum_{k=0}^{m-1} k\pi_k + \frac{m\pi_m}{1 - \rho},$$

which can be seen as the expected reward rate in the steady state with the reward rate assignment, $r_k = k$ for $k < m$ and $r_k = m$ for $k \geq m$. This formula can be simplified (see problem 5 at the end of this section):

$$E[M] = m\rho = \frac{\lambda}{\mu}. \quad (8.48)$$

Thus, the utilization of any individual server is $\rho = \lambda/(m\mu)$, while the average number of busy servers is equal to the traffic intensity λ/μ .

The probability that an arriving customer is required to join the queue is derived as

$$\begin{aligned} P(\text{queuing}) &= \sum_{k=m}^{\infty} \pi_k = \frac{\pi_m}{1 - \rho} \\ &= \frac{(m\rho)^m}{m!} \cdot \frac{\pi_0}{1 - \rho}, \end{aligned} \quad (8.49)$$

where π_0 is given in (8.46). Formula (8.49) finds application in telephone traffic theory and gives the probability that no trunk is available for an arriving call in an exchange with m trunks, assuming that blocked called are queued. This formula is referred to as *Erlang's C formula* (or *Erlang's delayed-call formula*).

Example 8.3

While designing a multiprocessor operating system, we wish to compare two different queuing schemes shown in Figure 8.7. The criterion for comparison will be the average response times $E[R_s]$ and $E[R_c]$. It is clear that the first organization corresponds to two independent $M/M/1$ queues, with $\rho = \lambda/(2\mu)$. Therefore, using equation (8.38), we have

$$E[R_s] = \frac{\frac{1}{\mu}}{1 - \frac{\lambda}{2\mu}} = \frac{2}{2\mu - \lambda}.$$

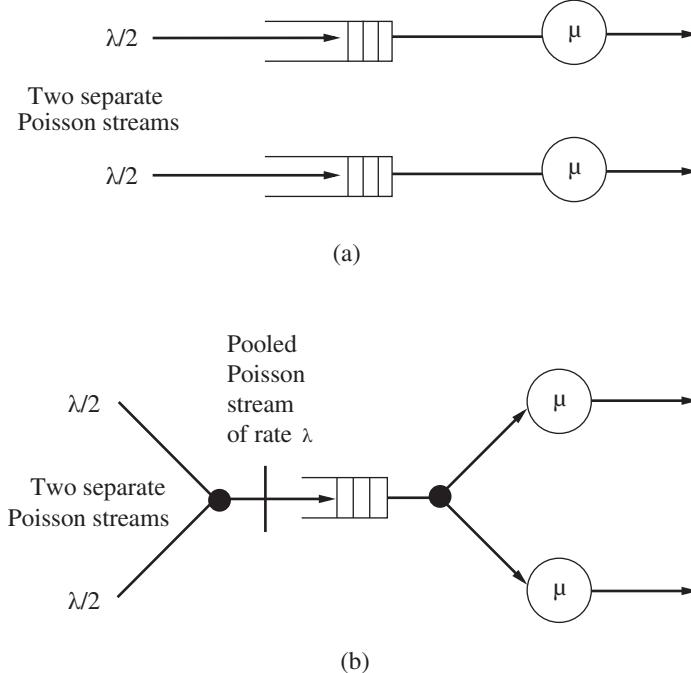


Figure 8.7. Queuing schemes: (a) separate queues; (b) common queue

On the other hand, the common queue organization corresponds to an $M/M/2$ system. To obtain $E[R_c]$, we first obtain $E[N_c]$ [using equation (8.47)] as

$$E[N_c] = 2\rho + \frac{\rho(2\rho)^2}{2!} \frac{\pi_0}{(1-\rho)^2} \quad \text{where } \rho = \frac{\lambda}{2\mu},$$

and using equation (8.46), we have

$$\begin{aligned} \pi_0 &= [1 + 2\rho + \frac{(2\rho)^2}{2!} \frac{1}{1-\rho}]^{-1} \\ &= \frac{1-\rho}{(1-\rho)(1+2\rho)+2\rho^2} = \frac{1-\rho}{1+\rho}. \end{aligned}$$

Thus

$$E[N_c] = 2\rho + 2\rho^3 \frac{1-\rho}{(1+\rho)(1-\rho)^2} = \frac{2\rho(1-\rho^2+\rho^2)}{1-\rho^2} = \frac{2\rho}{1-\rho^2}$$

and, using Little's formula, we have

$$\begin{aligned} E[R_c] &= \frac{E[N_c]}{\lambda} = \frac{\frac{1}{2\mu}}{1 - (\frac{\lambda}{2\mu})^2} \\ &= \frac{1}{\mu(1 - \rho^2)} = \frac{4\mu}{4\mu^2 - \lambda^2}. \end{aligned} \quad (8.50)$$

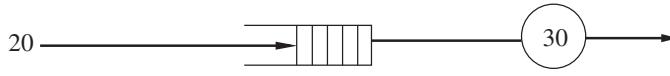
Now

$$E[R_s] = \frac{2}{2\mu - \lambda} = \frac{4\mu + 2\lambda}{4\mu^2 - \lambda^2} > E[R_c].$$

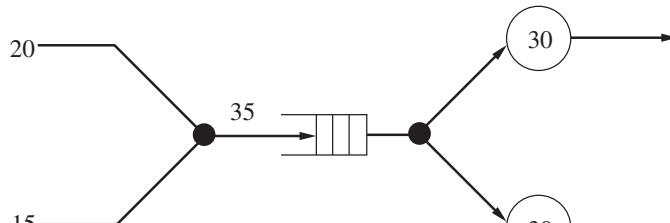
This implies that a common-queue organization is better than a separate-queue organization. This result generalizes to the case of m servers [KLEI 1976]. ‡

Example 8.4

Once again consider the problem of designing a system with two identical processors. We have two independent job streams with respective arrival rates $\lambda_1 = 20$ and $\lambda_2 = 15$ per hour. The average service time for both job types is $1/\mu = 2$ min = $\frac{1}{30}$



(a)



(b)

Figure 8.8. Queuing schemes for Example 8.4: (a) separate queues; (b) common queue

hours. Should we dedicate a processor per job stream, or should we pool the job streams and processors together (see Figure 8.8)? Let $E[R_{s1}]$ and $E[R_{s2}]$ be the average response times of the two job streams in the separate-queue organization and let $E[R_c]$ be the response time in the common-queue situation. Let $\rho_1 = \lambda_1/\mu = \frac{20}{30}$, $\rho_2 = \lambda_2/\mu = \frac{15}{30}$, $\rho = (\lambda_1 + \lambda_2)/(2\mu) = \frac{35}{60}$. Then

$$\begin{aligned} E[R_{s1}] &= \frac{\frac{1}{\mu}}{1 - \rho_1} = \frac{\frac{1}{\mu}}{1 - \frac{20}{30}}, \quad \text{using formula (8.38)} \\ &= \frac{1}{30 - 20} = \frac{1}{10} \text{ h} = 6 \text{ min.} \\ E[R_{s2}] &= \frac{\frac{1}{\mu}}{1 - \rho_2} = \frac{\frac{1}{\mu}}{1 - \frac{15}{30}}, \quad \text{using formula (8.38)} \\ &= \frac{1}{15} \text{ h} = 4 \text{ min.} \\ E[R_c] &= \frac{\frac{1}{\mu}}{1 - \rho^2}, \quad \text{using formula (8.50)} \\ &= \frac{\frac{1}{\mu}}{1 - (\frac{35}{60})^2} = 3.03 \text{ min.} \end{aligned}$$

Clearly, it is much better to form a common pool of jobs.

‡

Problems

- Consider a telephone switching system consisting of n trunks with an infinite caller population. The arrival stream is Poisson with rate λ and call holding times are exponentially distributed with average $1/\mu$. The traffic offered, A (in Erlangs), is defined to be the average number of call arrivals per holding time. Thus, $A = \lambda/\mu = \rho$, the traffic intensity. We assume that an arriving call is lost if all trunks are busy. This is known as BCC (blocked calls cleared) scheduling discipline. Draw the state diagram and derive an expression for π_i , the steady-state probability that i trunks are busy. Show that this pmf approaches the Poisson pmf in the limit $n \rightarrow \infty$ (i.e., ample-trunks case). Therefore, for finite n , the above pmf is known as the “truncated Poisson pmf”. Define the **call congestion**, $B(n)$, as the proportion of lost calls in the long run. Then show that

$$B(n) = \frac{\frac{\rho^n}{n!}}{\sum_{i=0}^n \frac{\rho^i}{i!}}. \quad (8.51)$$

This is known as Erlang's B formula. Define traffic carried, C (in Erlangs), to be the average number of calls completed in a time interval $1/\mu$. Then

$$C(n) = \sum_{i=0}^n i\pi_i.$$

Verify that:

$$B(n) = 1 - \frac{C(n)}{A}.$$

Show that the efficient formula to compute the loss probability holds [AKIM 1993]:

$$B(k) = \frac{\frac{\rho}{k} B(k-1)}{1 + \frac{\rho}{k} B(k-1)}, \quad k = 1, 2, \dots, n,$$

where $B(0) = 1$.

2. Derive the steady-state distribution of the waiting time W for an $M/M/2$ queuing system as follows:

- (a) First show that

$$P(W = 0) = \pi_0 + \pi_1.$$

- (b) Now, assuming that $n \geq 2$ jobs being present in the system at the time of arrival of the tagged job, argue that the distribution of W is $(n-1)$ -stage Erlang with parameter 2μ .

Compute the distribution function and hence compute the expected value of W .

3. * Show that the response time distribution in an $M/M/m$ -FCFS queue is given by [GROS 1998]:

$$F(t) = W_m(1 - e^{-\mu t}) + (1 - W_m) \left[\frac{m\mu - \lambda}{(m-1)\mu - \lambda} (1 - e^{-\mu t}) - \frac{\mu}{(m-1)\mu - \lambda} (1 - e^{-(m\mu - \lambda)t}) \right],$$

where $W_m = \sum_{j=0}^{m-1} \pi_j$ is the probability that waiting time $W = 0$.

4. $[M/M/\infty$ Queuing System] Suppose $\pi_n(t)$ is the probability that n telephone lines are busy at time t . Assume that infinitely many lines are available and that the call arrival rate is λ while average call duration is $1/\mu$. Derive the differential equation for $\pi_n(t)$. Solve the equation for $\pi_n(t)$ as $t \rightarrow \infty$. Let $E[N(t)]$ denote the average number of busy lines. Derive the differential equation for $E[N(t)]$. Obtain an expression for average length of queue $E[N]$ in the steady state. Also find the mean as well as the distribution of response time for this queuing system.
5. Show that the average number of busy servers for an $M/M/m$ queue in the steady-state is given by

$$E[M] = \frac{\lambda}{\mu}.$$

Also verify formula (8.47) for the average number in the system.

8.2.3 Finite State Space

We consider a special case of the birth-death process having a finite state space $\{0, 1, \dots, n\}$, with constant birth rates $\lambda_i = \lambda$, $0 \leq i \leq n - 1$, and constant death rates $\mu_i = \mu$, $1 \leq i \leq n$. Also let $\rho = \lambda/\mu$, as before. The generator matrix of the CTMC is given by

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \cdots & n \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ n \end{matrix} & \begin{bmatrix} -\lambda & \lambda & & & & \\ \mu & -(\lambda + \mu) & \lambda & & & \\ & \mu & -(\lambda + \mu) & \lambda & & \\ & & & \ddots & \ddots & \ddots \\ & & & & \mu & -\mu \end{bmatrix} \end{matrix}.$$

The state diagram is given by Figure 8.9, and the steady-state probabilities are

$$\pi_i = \rho^i \pi_0, \quad 0 \leq i \leq n,$$

$$\pi_0 = \frac{1}{\sum_{i=0}^n \rho^i} = \begin{cases} \frac{1-\rho}{1-\rho^{n+1}}, & \rho \neq 1 \\ \frac{1}{n+1}, & \rho = 1. \end{cases} \quad (8.52)$$

The same result can also be obtained from equation (8.33) by substituting the values of λ_i and μ_i . Note that such a system with a finite customer population will always be stable, irrespective of the value of ρ . Thus, (8.52) gives the steady-state probabilities for all finite values of ρ . The transient solution of the above CTMC is also known [MORS 1958]:

$$p_{mk}(t) = \pi_k + \frac{2\rho^{(k-m)/2}}{n+1} \sum_{i=1}^n \frac{1}{x_i} \left[\sin\left(\frac{im\pi}{n+1}\right) - \sqrt{\rho} \sin\left(\frac{i(m+1)\pi}{n+1}\right) \right] \times \left[\sin\left(\frac{ik\pi}{n+1}\right) - \sqrt{\rho} \sin\left(\frac{i(k+1)\pi}{n+1}\right) \right] e^{-\gamma_i t}. \quad (8.53)$$

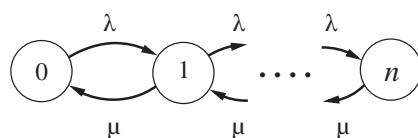


Figure 8.9. State diagram of a birth-death process with a finite state space

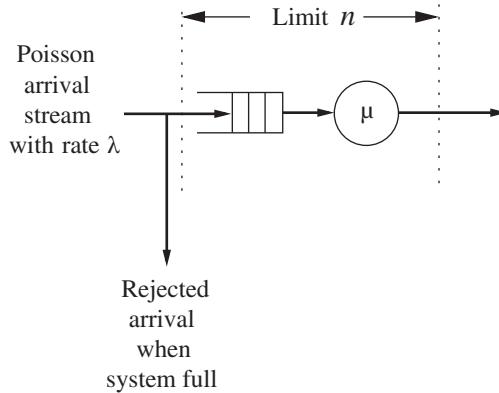


Figure 8.10. $M/M/1/n$ queuing system

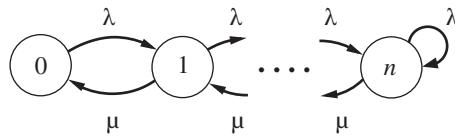


Figure 8.11. $M/M/1/n$ state diagram

Here k denotes the number of jobs in the system at time t and m denotes the initial number of jobs in the system. The $\{-\gamma_i\}$ are the eigenvalues of the generator matrix; they are given by

$$\gamma_i = \lambda + \mu - 2\sqrt{\lambda\mu} \cos\left(\frac{i\pi}{n+1}\right) = \mu x_i, \quad i = 1, 2, \dots, n. \quad (8.54)$$

Example 8.5 ($M/M/1/n$ Queue)

Consider an $M/M/1$ queuing system with a limited buffer space so that at most n jobs can be in the system at a time. In the Kendall notation, such a system will be denoted as $M/M/1/n$. Figure 8.10 shows the queuing system, and Figure 8.11 shows the homogeneous CTMC state diagram of the system.

The state diagram of the $M/M/1/n$ system is very similar to the one in Figure 8.9 except for the self-loop on state n . If a job arrives while the system is in state n , the job is rejected and the system remains in state n . It can be shown that the transient and the steady-state equations for a CTMC with such a self-loop are the same as those without a self-loop. Hence, the steady-state solution [equation (8.52)] and the transient solution [equation (8.53)] given above apply to the $M/M/1/n$ queue.

By assigning different reward rates, we can get different measures for this system as shown in Table 8.1.

TABLE 8.1. Measures for the $M/M/1/n$ system

Measure	Reward rate assignment	Expected steady-state reward rate
Mean number in system	$r_j = j$	$\frac{\rho}{1 - \rho} - \frac{n + 1}{1 - \rho^{n+1}} \rho^{n+1}$
Loss probability	$r_n = 1, r_j = 0 \ (j \neq n)$	$\pi_n = \frac{1 - \rho}{1 - \rho^{n+1}} \rho^n$
Throughput	$r_j = \mu \ (j \neq 0), r_0 = 0$ or $r_j = \lambda \ (j \neq n), r_n = 0$	$\mu(1 - \pi_0) = \lambda(1 - \pi_n)$
URTD ^a	$r_j = 1 - \sum_{i=0}^j \frac{(\mu t)^i e^{-\mu t}}{i!}; r_n = 0$	Equation (8.55)
CRTD ^b	$r_j = \frac{1 - \sum_{i=0}^j \frac{(\mu t)^i e^{-\mu t}}{i!}}{1 - \pi_n}; r_n = 0$	Equation (8.56)

^a Unconditional response time distribution.

^b Conditional response time distribution.

The mean response time (conditioned on the job being accepted) is obtained by dividing the mean number in system (row 1 of Table 8.1) by the throughput (row 3 of Table 8.1). The response time distribution can be obtained by an argument very similar to the one used for the $M/M/1$ queue:

$$F(t) = \sum_{j=0}^{n-1} \pi_j \left[1 - \sum_{i=0}^j \frac{(\mu t)^i e^{-\mu t}}{i!} \right]. \quad (8.55)$$

Note that this distribution is defective (for the definition of defective distribution see Section 3.4.9) with a mass at infinity equal to $1 - \sum_{j=0}^{n-1} \pi_j = \pi_n$; this is the probability that the job is not accepted and hence takes infinite amount of time to complete. In order to get the response time distribution conditional on the job being accepted, we divide expression (8.55) by $1 - \pi_n$:

$$F_c(t) = \sum_{j=0}^{n-1} \frac{\pi_j}{1 - \pi_n} \left[1 - \sum_{i=0}^j \frac{(\mu t)^i e^{-\mu t}}{i!} \right]. \quad (8.56)$$

#

Example 8.6 (Machine Breakdown)

Consider a component with a constant failure rate λ . On a failure, it is repaired with an exponential repair time distribution of parameter μ . Thus, the MTTF is $1/\lambda$ and

the MTTR is $1/\mu$. This is an example of the Markov chain of Figure 8.9 with $n = 1$ (a two-state system with state 0 as the up state and state 1 as the down state) with the generator matrix:

$$Q = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}.$$

Hence

$$\pi_0 = \frac{1 - \rho}{1 - \rho^2} = \frac{1}{1 + \rho}$$

and

$$\pi_1 = \frac{\rho}{1 + \rho}.$$

The steady-state availability is the steady-state probability that the system is in state 0, the state with the system functioning properly. Thus, from equation (8.52),

$$\begin{aligned} \text{Steady-state availability, } A &= \pi_0 = \frac{1}{1 + \rho} = \frac{1}{1 + \frac{\lambda}{\mu}} = \frac{\mu}{\lambda + \mu} \\ &= \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}. \end{aligned} \quad (8.57)$$

Note that a system with a low reliability will have a small MTTF, but if the repairs can be made fast enough (implying a low MTTR), the system may possess a high availability. In order to obtain the transient solution of the two-state availability model, we can use equation (8.53). There is only one eigenvalue (other than zero) of the Q matrix, say, $-\gamma_1$. From equation (8.54), we have

$$\gamma_1 = \lambda + \mu \text{ and } x_1 = \frac{\lambda + \mu}{\mu}.$$

Then using equation (8.53), we have,

$$\begin{aligned} p_{00}(t) &= \pi_0 + \frac{2}{2} \left[\frac{1}{x_1} (\sin 0 - \sqrt{\rho} \sin \frac{\pi}{2}) (\sin 0 - \sqrt{\rho} \sin \frac{\pi}{2}) e^{-(\lambda+\mu)t} \right] \\ &= \pi_0 + \frac{\rho\mu}{\lambda + \mu} e^{-(\lambda+\mu)t} = \pi_0 + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \\ &= \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}. \end{aligned}$$

If we assume that the system is in state 0 to begin with, then the instantaneous availability $A(t)$ is given by

$$A(t) = p_{00}(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}. \quad (8.58)$$

By integration and subsequent division by t , we obtain the expected interval availability $A_I(t)$ as:

$$A_I(t) = \frac{\int_0^t A(x) dx}{t} = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{(\lambda + \mu)^2 t} (1 - e^{-(\lambda + \mu)t}). \quad (8.59)$$

Note that the limiting values of $A_I(t)$ and $A(t)$ are equal to the steady-state availability A . ‡

Example 8.7 (Example 8.6 Continued)

We now consider task-oriented measures for the two-state availability model. Consider a task that needs x amount of time to execute in absence of failures. Let $T(x)$ be the completion time of the task. First consider $\lambda = 0$ so that there are no failures. In this case $T_1(x) = x$, and hence the distribution function of $T_1(x)$ is given by (see Figure 8.12)

$$F_{T_1(x)}(t) = u(t - x),$$

where $u(t - x)$ is the unit-step function at $t = x$. Next consider a nonzero value of λ but set $\mu = 0$. In this case, if we assume that the server is up when the task arrives, the task will complete at time x provided the server does not fail in the interval $(0, x)$. Otherwise, the task will never complete. Then

$$F_{T_2(x)}(t) = e^{-\lambda x} u(t - x).$$

In Figure 8.12, note that $T_2(x)$ is a defective random variable (for the definition of defective distribution see Section 3.4.9) with a defect at infinity equal to $1 - e^{-\lambda x}$, the probability that a task will never finish.

The third case where the server can fail and get repaired is quite complex. If a server failure occurs before the task is completed, we need to consider two separate cases. If the work done so far is not lost so that when the server repair is completed, the task resumes from where it was interrupted, we have the *preemptive resume*

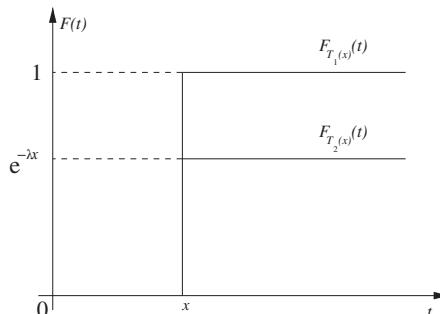


Figure 8.12. Task completion time distribution

(*prs*) case. Otherwise we have the *preemptive repeat* (*prt*) case. We simply quote the result here, the interested reader may consult Chimento and Trivedi [CHIM 1993].

The LST of the completion time distributions for the two cases are

$$L_{\text{prs}}(s) = \exp\left\{-\frac{s^2 + (\lambda + \mu)s}{s + \mu}x\right\} \quad (8.60)$$

and

$$L_{\text{prt}}(s) = \frac{e^{-(s+\lambda)x}}{1 - \frac{\lambda}{s+\lambda} \frac{\mu}{s+\mu}(1 - e^{-(s+\lambda)x})}. \quad (8.61)$$

The transforms need to be numerically inverted in order to get the distribution functions. Expectations can also be obtained by differentiating with respect to s , setting $s = 0$ and multiplying by -1 .

#

Availability models such as in Example 8.6 assume that all failures are recoverable. Consequently, the Markov chains of such systems are irreducible. If we assume that some failures are irrecoverable, then the system will have one or more absorbing states. In such cases, we study the distribution of time to reach an absorbing state (or failure state), and system reliability (see Section 8.5 for some examples).

Example 8.8 (Cyclic Queuing Model of a Multiprogramming System)

Consider the cyclic queuing model shown in Figure 8.13. Assume that the lengths of successive CPU execution bursts are independent exponentially distributed random variables with mean $1/\mu$ and that successive I/O burst times are also independent exponentially distributed variables with mean $1/\lambda$. At the end of a CPU burst, a program requests an I/O operation with probability q_1 ($0 \leq q_1 \leq 1$), and it completes execution with probability q_0 ($q_1 + q_0 = 1$). At the end of a program completion, another statistically identical program enters the system, leaving the number of programs in the system at a constant level n (known as the **degree of multiprogramming**).

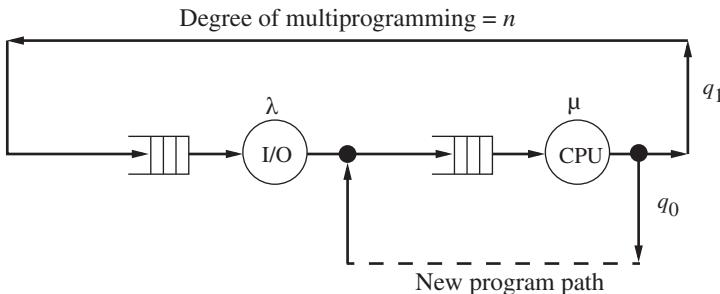


Figure 8.13. The cyclic queuing model of a multiprogramming system

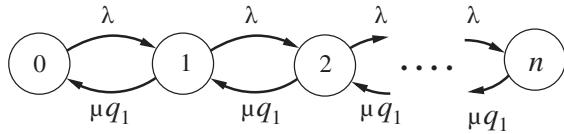


Figure 8.14. The state diagram for the cyclic queuing model

Let the number of programs in the CPU queue including any being served at the CPU denote the state of the system, i , where $0 \leq i \leq n$. Then the state diagram is given by Figure 8.14. Denoting $\lambda/(\mu q_1)$ by ρ , we see that the steady-state probabilities are given by

$$\pi_i = \left(\frac{\lambda}{\mu q_1} \right)^i \pi_0 = \rho^i \pi_0, \text{ and } \pi_0 = \frac{1}{\sum_{i=0}^n \rho^i},$$

so that

$$\pi_0 = \begin{cases} \frac{1-\rho}{1-\rho^{n+1}}, & \rho \neq 1, \\ \frac{1}{n+1}, & \rho = 1. \end{cases}$$

The CPU utilization is given by

$$U_0 = 1 - \pi_0 = \begin{cases} \frac{\rho - \rho^{n+1}}{1 - \rho^{n+1}}, & \rho \neq 1, \\ \frac{n}{n+1}, & \rho = 1. \end{cases} \quad (8.62)$$

Let $C(t)$ denote the number of jobs completed by time t . Then the (time) average $C(t)/t$ converges, under appropriate conditions, to a limit as t approaches ∞ [ROSS 1970]. This limit is the average system throughput in the steady state, and (with a slight abuse of notation) it is denoted here by $E[T]$. Whenever the CPU is busy, the rate at which CPU bursts are completed is μ , and a fraction q_0 of these will contribute to the throughput. Then

$$E[T] = \mu q_0 U_0. \quad (8.63)$$

For fixed values of μ and q_0 , $E[T]$ is proportional to the CPU utilization, U_0 .

Let the random variable B_0 denote the total CPU time requirement of a tagged program. Then $B_0 \sim EXP(\mu q_0)$. This is true because B_0 is the random sum of K CPU service bursts, which are independent $EXP(\mu)$ random variables. Here the random variable K is the number of visits to the CPU per program and hence is geometrically distributed with parameter q_0 . The required result is then obtained from our discussion on random sums in Chapter 5. Alternatively, the average number of visits V_0 to the CPU is $V_0 = 1/q_0$ (see Example 7.20), and thus $E[B_0] = V_0 E[S_0] = 1/(\mu q_0)$, where $E[S_0] = 1/\mu$ is the average CPU time per burst.

The average throughput can now be rewritten as

$$E[T] = \frac{U_0}{E[B_0]}. \quad (8.64)$$

If B_1 represents the total I/O service time per program, then as in the case of CPU:

$$E[B_1] = \frac{q_1}{q_0} \frac{1}{\lambda} = V_1 E[S_1],$$

where the average number of visits V_1 to the I/O device is given by $V_1 = q_1/q_0$ (by Example 7.20), and $E[S_1] = 1/\lambda$ is the average time per I/O operation. (Note that if U_1 denotes the utilization of the I/O device then, similar to (8.64), we have $E[T] = U_1/E[B_1]$.) Now the parameter ρ can be rewritten as follows:

$$\rho = \frac{\lambda}{\mu q_1} = \frac{q_0 \lambda}{q_1} \cdot \frac{1}{\mu q_0} = \frac{E[B_0]}{E[B_1]}. \quad (8.65)$$

Thus ρ indicates the relative measure of the CPU versus I/O requirements of a program. If the CPU requirement $E[B_0]$ is less than the I/O requirement $E[B_1]$ (i.e., $\rho < 1$), the program is said to be **I/O-bound**; if $\rho > 1$, then program is said to be **CPU-bound**; and otherwise it is called **balanced**.

In Figure 8.15 we have plotted U_0 as a function of the balance factor ρ and of the degree of multiprogramming n . When $\rho \ll 1$ or $\rho \gg 1$, U_0 is insensitive to n .

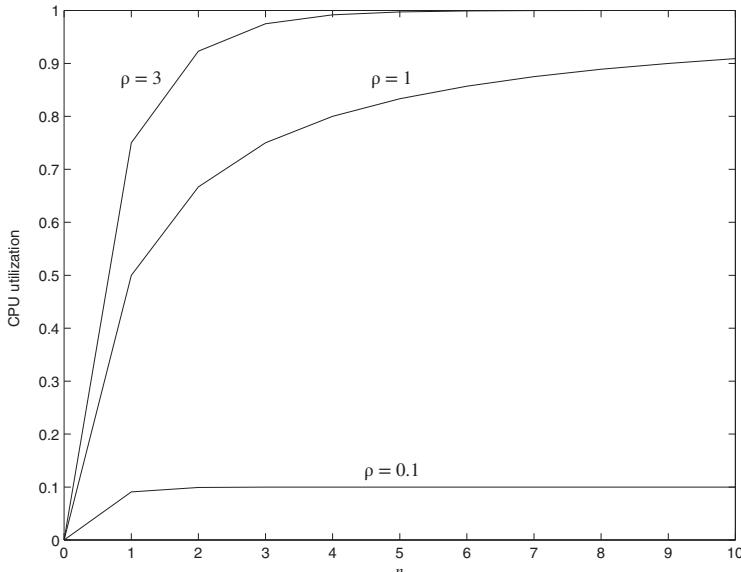


Figure 8.15. The CPU utilization as a function of the degree of multiprogramming

Thus, multiprogramming is capable of appreciably improving throughput only when the workload is nearly balanced (i.e., ρ is close to 1).

‡

Example 8.9

Let us return to the availability model of Example 8.6 and augment the system with $(n - 1)$ identical copies of the component, which are to be used in a standby spare mode. Assume that an unpowered spare does not fail and that switching a spare is a fault-free process. Then the system is in state k provided that $n - k$ units are in working order and k units are under repair. We picture this situation as the cyclic queuing model of Figure 8.13, with $q_0 = 0$ and $q_1 = 1$. The total number of components, n , is the analog of the degree of multiprogramming. The queue of available components is the analog of the I/O queue; the queue of components under repair is represented by the CPU queue; $\text{MTTR} = 1/\mu$, and $\text{MTTF} = 1/\lambda$. The steady-state availability is given by

$$\begin{aligned} A &= P(\text{"at least one copy is functioning properly"}) \\ &= \pi_0 + \pi_1 + \cdots + \pi_{n-1} = 1 - \pi_n \\ &= \begin{cases} 1 - \rho^n \cdot \frac{1 - \rho}{1 - \rho^{n+1}}, & \rho \neq 1, \\ \frac{n}{n+1}, & \rho = 1, \end{cases} \end{aligned}$$

or

$$A = \begin{cases} \frac{1 - \rho^n}{1 - \rho^{n+1}}, & \rho \neq 1, \\ \frac{n}{n+1}, & \rho = 1, \end{cases} \quad (8.66)$$

where $\rho = \text{MTTR}/\text{MTTF}$. For $n = 1$, we obtain

$$A = \frac{1}{1 + \rho} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

as in Example 8.6, and as $n \rightarrow \infty$, we have

$$A \rightarrow \min \left\{ 1, \frac{1}{\rho} \right\} = \min \left\{ 1, \frac{\text{MTTF}}{\text{MTTR}} \right\}.$$

In the usual case, $\text{MTTR} \ll \text{MTTF}$ and steady-state availability will approach unity as the number of spares increases. In Table 8.2 we have shown how the system availability A increases with the number of spares for $\rho = 0.01$. Thus, even though the single-component availability has only two 9s, the number of 9s increases by two

TABLE 8.2. Availability of a standby redundant system

<i>n</i>	<i>Availability</i>	<i>Number of spares</i>
1	0.99009900	0
2	0.99990099	1
3	0.99999990	2
4	0.99999999	3

with each additional spare in this case. We caution the reader that the increase in availability will not be as significant if the spare failure rate is nonzero or if detection and switching are imperfect. Also note that we can capture imperfect repair in the above model by setting $q_0 > 0$.

‡

Problems

1. Plot the response time distribution (both conditional and unconditional) of the $M/M/1/n$ queue assuming $\lambda = 0.9$ and $\mu = 1$. Vary $n = 10, 50, 100$. Also plot the rejection probability π_n as a function of n . Obtain the expression for the mean response time from the (conditional) response time distribution (equation (8.56)) and show that it is the same as that obtained using Little's result.
2. Derive the steady-state probabilities for $M/M/m/n$ queue. Then derive the expression for the loss probability. Finally, derive the response time distribution (both the conditional and unconditional) using the approach for $M/M/1/n$ queue.
3. Plot task completion time distributions for the two-state availability model for the two cases of $\lambda = 0$ and $\lambda = 0.1$. Assume $\mu = 0$ and $x = 10$. For extra credit, numerically invert the LSTs for the *prs* and *prt* cases and plot these two distribution functions as well on the same plot. Assume $\mu = 1$ and $\lambda = 0.1$.
4. From the LSTs of the completion time (equations (8.60) and (8.61)), derive expressions for the average completion time for the *prs* and *prt* cases in the two-state availability model.
5. Specify reward assignments to the CTMC of Figure 8.14 for computing $E[T]$, U_1 , and U_0 .
6. [FULL 1975] Consider a variation of the cyclic queuing network model of Example 8.8, in which the I/O device uses the SLTF (shortest latency time first) scheduling discipline. The I/O service rate, λ_k , is a function of the number of requests in its queue and is given by

$$\frac{1}{\lambda_k} = \frac{\tau}{k+1} + \frac{1}{r} = \frac{r\tau + k + 1}{r(k+1)},$$

where $1/r$ is the mean record transmission time and τ is the rotation time of the device. Obtain an expression for the CPU utilization and the average system throughput, assuming $q_0 = 0.05$, $r\tau = \frac{1}{3}$, and $\tau = 10$ ms. Plot the average system throughput as a function of the CPU service rate μ (ranging from $0.1r$ to $10r$) for various values of the degree of multiprogramming $n = 1, 2, 5, 10$. For $\mu = r$, compare the throughputs obtained by the SLTF scheduling and the FCFS scheduling algorithm (for the latter case, the average I/O service time is constant at $1/\lambda_1$).

7. Consider an application of the cyclic queuing model (CPU, paging device) to demonstrate a phenomenon called “thrashing” that could occur in paged virtual memory systems. Assume a fixed number, M , of main-memory page frames, equally divided among n active programs. Increasing n implies a smaller page allotment per program, which in turn implies increased paging activity—that is, an increased value of μ . Often it is assumed that

$$\mu(n) = \frac{1}{a} \left(\frac{M}{n} \right)^{-b}.$$

Assuming $\lambda = 0.0001$, $a = 0.2$, $b = 2.00$, and $M = 100$, plot the average system throughput as a function of the degree of multiprogramming. [Note that q_0 and q_1 are functions of n and that when $n = 1$, a job is assumed to have all the memory it needs, requiring only initial paging, so that, $q_0(1) = 0.9$.] Unlike the model of nonpaged systems, the average throughput here will not increase monotonically, but after a critical value of n , it will drop sharply. This is the phenomenon of thrashing [NUTT 1997].

8. For the availability model with $n - 1$ spares, obtain expressions for instantaneous and expected interval availabilities by specializing equation (8.53). For $n = 1, 2, 3$, numerically compute $A(t)$, $A_I(t)$ and compare your results with those obtained using a software package such as SHARPE [SAHN 1996] with $\lambda = 0.0001$ and $\mu = 1$.
9. Show that the number of nines in the steady-state availability A is given by $-\log_{10}(1 - A)$.
10. In this problem we wish to investigate the effect of a self-loop on a state of a CTMC (see Figure 8.11). For instance, consider the two CTMCs in Figure 8.P.1.
 - (a) First, write down the matrix of transition rates $R = [r_{ij}]$, where r_{ij} is the transition rate from state i to state j , and the infinitesimal generator matrix $Q = [q_{ij}]$, where

$$q_{ij} = r_{ij}, \quad i \neq j;$$

$$q_{ii} = - \sum_{j \neq i} r_{ij}.$$

Also show that Q matrices are identical despite different R matrices.

- (b) Define T as the total sojourn time (the time measured from the instance of entering a state until the instance of leaving the state), and X_i as the i th mini-sojourn time (the time between the occurrence of an incoming transition and an outgoing transition, including the self-loop). T is the

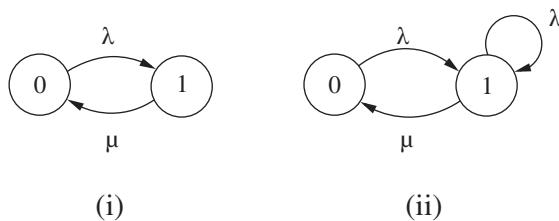


Figure 8.P.1. Two CTMCs

random sum (Section 5.5) of the $\{X_i\}$, that is, $T = \sum_{i=1}^N X_i$, where N is random. Show that the distribution of T is the same in the two CTMCs even though the distribution of X_i for state 1 in the two CTMCs are different. [Hint: In case (ii) in Figure 8.P.1, $X_i \sim EXP(\lambda + \mu)$ and $T \sim EXP(\mu)$.]

8.2.3.1 Machine Repairman Model. An interesting special case of the birth-death process occurs when the birth rate λ_j is of the form $(M - j)\lambda$, $j = 0, 1, \dots, M$, and the death rate $\mu_j = \mu$. Such a situation occurs in the modeling of a server where an individual client issues a request at the rate λ whenever it is in the “thinking state.” If j out of the total of M clients are currently waiting for a response to a pending request, the effective request rate is $(M - j)\lambda$. Here μ denotes the request completion rate. A similar situation arises when M machines share a repair facility. The failure rate of each machine is λ and the repair rate is μ .

The state diagram of such a finite-population system is given in Figure 8.16. The expressions for steady-state probabilities are obtained using equation (8.33) as

$$\pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda(M-i)}{\mu}, \quad 0 \leq k \leq M,$$

or

$$\pi_k = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{M!}{(M-k)!} = \pi_0 \rho^k \frac{M!}{(M-k)!}.$$

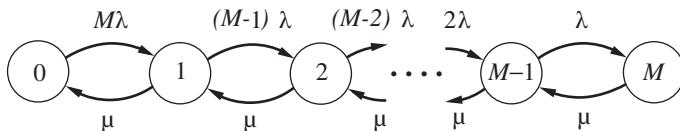


Figure 8.16. The state diagram of a finite-population queuing system

Hence

$$\pi_0 = \frac{1}{\sum_{k=0}^M \rho^k \frac{M!}{(M-k)!}}. \quad (8.67)$$

Example 8.10

Consider a parallel redundant system with M components, each with a constant failure rate λ . The system is unavailable for use whenever all M components have failed and are waiting for repairs. We wish to compare the following designs of the repair facility.

1. Each component has its own repair facility with repair rate μ . Then the availability of an individual component is given by formula (8.57) as

$$\frac{1}{1 + \frac{\lambda}{\mu}} = \frac{1}{1 + \rho}. \quad (8.68)$$

Now the system availability is computed by means of a reliability block diagram with M components in parallel each having the availability given by the expression (8.68). Hence system availability in this case is

$$A_1 = 1 - \left(\frac{\rho}{1 + \rho} \right)^M.$$

Note that in this scheme, no machine has to wait for a repair facility to be available.

2. We want to economize on the repair facilities and share a single repair facility of rate μ among all M machines. Then equation (8.67) applies, and noting that the system is down only when all the components are undergoing repair, we compute the steady-state availability by

$$A_2 = 1 - \pi_M = 1 - \frac{\rho^M M!}{\sum_{k=0}^M \rho^k \frac{M!}{(M-k)!}}.$$

Note that by assigning reward rate 1 to all the up states of the CTMC in Figure 8.16, steady-state availability can be obtained as the steady-state expected reward rate.

3. If we find that the availability A_2 is low, we may speed up the rate of the repair facility to $M\mu$, while retaining a single repair facility. Then, using equation (8.67), we have

$$A_3 = 1 - \frac{\left(\frac{\lambda}{M\mu}\right)^M M!}{\sum_{k=0}^M \left(\frac{\lambda}{M\mu}\right)^k \frac{M!}{(M-k)!}}.$$

TABLE 8.3. Availabilities for parallel redundant system

ρ	<i>M</i> number of components	<i>A</i> ₁ <i>Individual</i> <i>repair facility</i>	<i>Single repair</i> <i>facility of</i> <i>rate</i> μ	<i>Single repair</i> <i>facility of</i> <i>rate</i> $M\mu$
			<i>A</i> ₂	<i>A</i> ₃
0.1	1	0.909091	0.909091	0.909091
	2	0.991736	0.983607	0.995475
	3	0.999249	0.995608	0.999799
0.0001	1	0.999900	0.999900	0.999900
	2	0.999999	0.999999	0.999999
	3	0.999999	0.999999	0.999999

Table 8.3 shows the values of A_1 , A_2 and A_3 for various values of M , assuming that $\rho = 0.1$ and $\rho = 0.0001$. It is clear that

$$A_3 \geq A_1 \geq A_2.$$

#

Example 8.11 (Equivalent Failure and Repair Rates)

As was seen in Example 8.10, steady-state availability formulas for multistate systems can be quite complex. Yet in practice, engineers often wish to present steady-state system availability in the simple form that is known for a two-state system:

$$A = \frac{\text{MTTF}_{\text{eq}}}{\text{MTTF}_{\text{eq}} + \text{MTTR}_{\text{eq}}} = \frac{\mu_{\text{eq}}}{\lambda_{\text{eq}} + \mu_{\text{eq}}}.$$

We can then view the system behavior as an alternating renewal process discussed in Chapter 6.

For this purpose, we need to properly define the equivalent failure rate and repair rate for the system. We first partition the states into two classes of states, up states and down states. Transitions from up states to down states are called *red transitions*. Transitions from down states to up states are called *green transitions*. The equivalent failure rate λ_{eq} is equal to the summation over all the red transitions of the transition's failure rate times the conditional probability of being in the transition's source state, given that the source state is in the set of up states. A similar technique is applied to the green transitions to generate μ_{eq} .

Let I designate the set of all states, U the set of up states, D the set of down states, R the set of red transitions, G the set of green transitions and t_{ij} the transition from state i to state j . The following equation shows how to compute λ_{eq} :

$$\lambda_{\text{eq}} = \sum_{t_{ij} \in R} P(\text{system in state } i \mid \text{system is up}) \times q_{ij} = \frac{\sum_{t_{ij} \in R} \pi_i \times q_{ij}}{A},$$

where

$$A = \sum_{k \in U} \pi_k.$$

The computation of μ_{eq} is similar.

$$\mu_{\text{eq}} = \sum_{t_{ij} \in G} P(\text{system in state } i \mid \text{system is down}) \times q_{ij} = \frac{\sum_{t_{ij} \in G} \pi_i \times q_{ij}}{1 - A}.$$

For the availability model of Figure 8.16,

$$\lambda_{\text{eq}} = \frac{\lambda \pi_{M-1}}{\sum_{j=0}^{M-1} \pi_j} = \frac{\lambda \pi_{M-1}}{1 - \pi_M}$$

and

$$\mu_{\text{eq}} = \mu.$$

Now

$$A = \frac{\mu_{\text{eq}}}{\lambda_{\text{eq}} + \mu_{\text{eq}}} = \frac{\mu}{\frac{\lambda \pi_{M-1}}{1 - \pi_M} + \mu} = \frac{\mu(1 - \pi_M)}{\lambda \pi_{M-1} + \mu(1 - \pi_M)}$$

(noting that $\lambda \pi_{M-1} = \mu \pi_M$ from the balance equation for state M)

$$= \frac{\mu(1 - \pi_M)}{\mu \pi_M + \mu(1 - \pi_M)} = 1 - \pi_M.$$

‡

Example 8.12 (Slot Availability of a k -out-of- n System)

Consider a k -out-of- n system with k service slots and n components (or slot units). Initially k of these components occupy the respective service slots, and the remaining $(n - k)$ components form a shared pool of spares. The failure rate of each component is λ , and the repair rate is μ . We will assume that a single repair facility is shared by all components. A service slot fails if the component in that slot has failed and no working spare is available. We wish to obtain an expression for the expected number of failed service slots in the steady state. Assuming that the number of failed components denotes the system state, the CTMC for this model is shown in Figure 8.17.

The reward rate assigned to state j is such that it denotes the number of failed service slots in that state. Thus $r_j = 0$, $j = 0, \dots, n - k$ and $r_j = j - (n - k)$ for

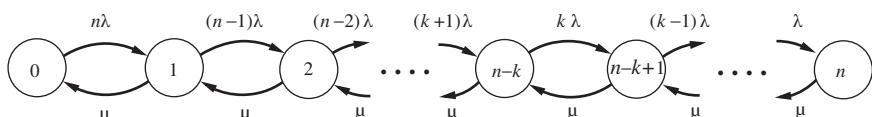


Figure 8.17. Service slot availability model

$j = n - k + 1, \dots, n$. The expected reward rate in the steady state then gives the expected number of failed service slots:

$$E[S_f] = \sum_{j=n-k+1}^n [j - (n-k)]\pi_j, \quad (8.69)$$

where

$$\pi_j = \frac{(\frac{\lambda}{\mu})^j \frac{n!}{(n-j)!}}{\sum_{j=0}^n (\frac{\lambda}{\mu})^j \frac{n!}{(n-j)!}}.$$

¶

Example 8.13 (Response Time in a Client–Server System)

Consider a client–server system with M clients in which individual think times are exponentially distributed with mean $1/\lambda$ seconds (see Figure 8.18). Assume that the service time per request, B_0 , is exponentially distributed with mean $E[B_0] = 1/\mu$ seconds. Then the steady-state probability that there are n requests executing or waiting on the CPU is given by [equation (8.67)]

$$\pi_n = \pi_0 \rho^n \frac{M!}{(M-n)!}, \quad n = 0, 1, 2, \dots, M,$$

and the probability that the CPU is idle is

$$\pi_0 = \frac{1}{\sum_{n=0}^M \rho^n \frac{M!}{(M-n)!}},$$

where $\rho = \lambda/\mu$.

The CPU utilization U_0 is $1 - \pi_0$, and the average rate of request completion is $E[T] = \mu(1 - \pi_0) = U_0/E[B_0]$. If $E[R]$ denotes the average response time, then on

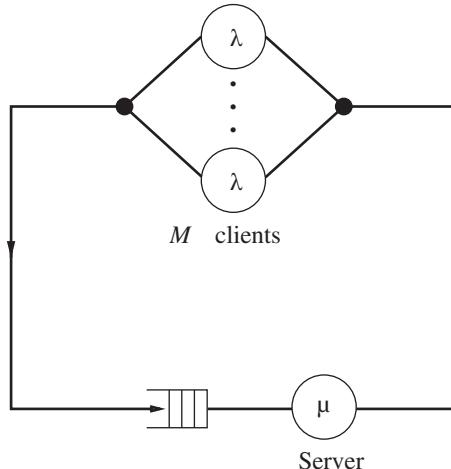


Figure 8.18. A client–server system

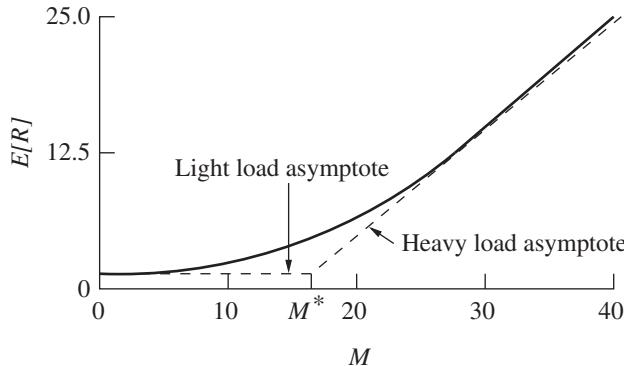


Figure 8.19. Average response time as a function of the number of clients

the average a request is generated by a given client in $E[R] + (1/\lambda)$ seconds. Thus the average request generation rate of the client subsystem is $M/[E[R] + (1/\lambda)]$. In the steady state, the request generation and completion rates must be equal. Therefore we have

$$\frac{M}{E[R] + \frac{1}{\lambda}} = \mu(1 - \pi_0)$$

or

$$\begin{aligned} E[R] &= \frac{M}{\mu(1 - \pi_0)} - \frac{1}{\lambda} = \frac{M \cdot E[B_0]}{U_0} - \frac{1}{\lambda} = \frac{M}{E[T]} - \frac{1}{\lambda} \\ &= \frac{\text{number of clients}}{\text{average throughput}} - \text{average think time}. \end{aligned} \quad (8.70)$$

The last expression for average response time can also be derived using Little's formula, and as such it is known to hold under rather general conditions [DENN 1978]. In Figure 8.19, $E[R]$ is plotted as a function of the number of clients, M , assuming $1/\lambda = 15$ s and $1/\mu = 1$ s.

When the number of clients $M = 1$, there is no queuing and the response time $E[R]$ equals the average service time $E[B_0]$. As the number of clients increases, there is increased congestion as the server utilization U_0 approaches unity. In the limit $M \rightarrow \infty$, $E[R]$ is a linear function [$ME[B_0] - (1/\lambda)$] of M . In this limit, the installation of an additional client increases every other client's response time by the new client's service time $E[B_0]$. This complete state of interference is to be generally avoided. The number of clients, M^* , for which the heavy-load asymptote $E[R] = ME[B_0] - (1/\lambda)$ intersects with the light-load asymptote $E[R] = E[B_0]$ is therefore called the **saturation number** [KLEI 1976] and is given by

$$M^* = \frac{E[B_0] + 1/\lambda}{E[B_0]} = 1 + \frac{\mu}{\lambda}. \quad (8.71)$$

For our example, the number of clients beyond which we call the system saturated is given by $M^* = 16$.



Problems

- For the availability model of parallel redundant system with shared repair person (part 2 of Example 8.10), write down the generator matrix Q and derive expressions for λ_{eq} and μ_{eq} for $M = 1, 2, 3$.
- Assuming an average think time of 10 s, design a client–server system to support 101 clients without saturation ($M \leq M^*$). We assume that on the average each request from any client requires the execution of 1,000,000 machine instructions. The result of the design process should be the minimum number of instructions that the server should be able to execute per unit time.
- Consider a model of a telephone switching system consisting of n trunks with a finite caller population of M callers. This is a variation of problem 1 in Section 8.2.2 where we had an infinite caller population. The average call rate of an idle caller (free source) is λ calls per unit time, and the average holding time of a call is $1/\mu$. If an arriving call finds all trunks busy, then it is lost (i.e., BCC scheduling discipline is used). Assuming that the call holding times and the intercall times of each caller are exponentially distributed, draw the state diagram and derive an expression for the steady-state probability $\pi_i = P(i \text{ trunks are busy})$, $i = 0, 1, \dots, n$. The resulting pmf is known as the Engset pmf. Show that the expected total traffic offered (in Erlangs) by the M sources per holding time is given by

$$\begin{aligned} A &= \sum_{i=0}^n \pi_i (M-i) \left(\frac{\lambda}{\mu} \right) \\ &= \frac{M \sum_{i=0}^n \binom{M-1}{i} \left(\frac{\lambda}{\mu} \right)^{i+1}}{\sum_{i=0}^n \binom{M}{i} \left(\frac{\lambda}{\mu} \right)^i}. \end{aligned}$$

Let $\rho = \lambda/\mu$.

Next obtain an expression for the traffic carried (in Erlangs) by the switching system per holding time:

$$C = \sum_{j=0}^n j \pi_j.$$

Now the probability, B , that a given call is lost is computed as

$$B = 1 - \frac{C}{A} = \frac{\binom{M-1}{n} \rho^n}{\sum_{i=0}^n \binom{M-1}{i} \rho^i}.$$

The quantity B is also known as call congestion.

8.2.3.2 Wireless Handoff Performance Model. As a variation of the $M/M/n$ loss system that was discussed in the telephone trunk problem (problem 1 at the end of Section 8.2.2), consider the performance model of a single cell in a cellular wireless communication network [HONG 1986]. New calls arrive in a Poisson stream at the rate λ_1 and handoff calls arrive in a Poisson stream at the rate λ_2 . An ongoing call (new or handoff) completes service at the rate μ_1 , and the mobile engaged in the call departs the cell at the rate μ_2 . There are a limited number of channels, n , in the channel pool. When a handoff call arrives and an idle channel is available in the channel pool, the call is accepted and a channel is assigned to it. Otherwise, the handoff call is dropped. When a new call arrives, it is accepted provided $g + 1$ or more channels are available in the channel pool; otherwise the new call is blocked. Here g is the number of guard channels used so as to give priority to handoff calls. We assume that $g < n$ in order not to block new calls altogether.

Figure 8.20 shows the finite state birth-death model where the number of busy channels is the state index. Let $\lambda = \lambda_1 + \lambda_2$, $\mu = \mu_1 + \mu_2$, $A = \rho = \lambda/\mu$ and $A_1 = \lambda_2/\mu$. Steady-state probabilities are then given by [using equation (8.33)]

$$\pi_k = \begin{cases} \pi_0 \frac{A^k}{k!}, & k \leq n - g \\ \pi_0 \frac{A^{n-g}}{k!} A_1^{k-(n-g)}, & k \geq n - g \end{cases}$$

where

$$\pi_0 = \frac{1}{\sum_{k=0}^{n-g-1} \frac{A^k}{k!} + \sum_{k=n-g}^n \frac{A^{n-g}}{k!} A_1^{k-(n-g)}}.$$

The probability of dropping a handoff call is given by

$$P_d(n, g) = \pi_n = \frac{\frac{A^{n-g}}{n!} A_1^g}{\sum_{k=0}^{n-g-1} \frac{A^k}{k!} + \sum_{k=n-g}^n \frac{A^{n-g}}{k!} A_1^{k-(n-g)}}. \quad (8.72)$$

The probability of blocking a new call is obtained by assigning reward rate 1 to states $n - g$ to n and reward rate zero to the remaining states:

$$P_b(n, g) = \sum_{k=n-g}^n \pi_k = \frac{\sum_{k=n-g}^n \frac{A^{n-g}}{k!} A_1^{k-(n-g)}}{\sum_{k=0}^{n-g-1} \frac{A^k}{k!} + \sum_{k=n-g}^n \frac{A^{n-g}}{k!} A_1^{k-(n-g)}}. \quad (8.73)$$

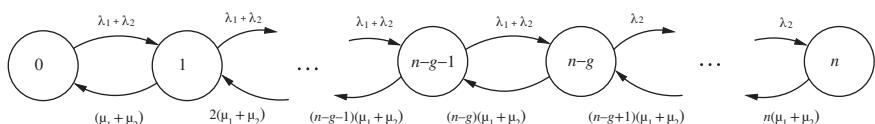


Figure 8.20. CTMC for the wireless handoff model

By setting $g = 0$, formulas (8.72) and (8.73) reduce to the Erlang B loss formula (8.51).

So far, we assumed that besides λ_1 , μ_1 , and μ_2 , λ_2 , the handoff arrival rate is an input parameter to the model above. However, λ_2 is determined by handoff out from neighboring cells and as such depends on the parameters λ_1 , μ_1 , μ_2 , n , and g . Assuming that all cells are statistically identical, handoff out throughput from a cell must equal handoff arrival rate in steady state.

Let $T(\lambda_2)$ denote the handoff out rate from the cell; then

$$T(\lambda_2) = \mu_2 \sum_{k=1}^n k \pi_k.$$

We set up a fixed-point equation

$$\lambda_2 = T(\lambda_2) \quad (8.74)$$

to determine the value of λ_2 . Starting with some initial value of λ_2 , we iterate using the equation (8.74) until convergence is reached. Haring *et al.* [HARI 2001], have shown that equation (8.74) has a unique solution.

Problems

1. Show that the dropping probability and the blocking probability satisfy the following computationally efficient recursive relationships [HARI 2001]:

for $k = 1, 2, \dots, g$,

$$P_d(n_1 + k, k) = \frac{P_d(n_1 + k - 1, k - 1)}{\frac{n}{\alpha A} + P_d(n_1 + k - 1, k - 1)},$$

and

$$P_b(n_1 + k, k) = \frac{\frac{n}{\alpha A} P_b(n_1 + k - 1, k - 1) + P_d(n_1 + k - 1, k - 1)}{\frac{n}{\alpha A} + P_d(n_1 + k - 1, k - 1)},$$

where $\alpha A = A_1$ and $n = n_1 + k$.

2. Compute and plot the loss probabilities $P_b(n, g)$ and $P_d(n, g)$ for $n = 1$ to 100 and $g = 0, 1, 2$, and 3. Assume $A = 70$ and $\alpha = 0.3$.
3. Determine the optimal number of guard channels, g , for $A = 80$ Erlangs, $\mu_1 + \mu_2 = 1$, $\alpha = 0.5$ so as to minimize the blocking probability of new calls subject to the constraint that the dropping probability of handoff calls is not to exceed 10^{-6} . Consider two cases of $n = 120$ and $n = 125$. For further details on this optimization problem see [HARI 2001].

Additional Problems

1. You are given a hybrid k -out-of- n system in which n units are active and m units are in a standby status so that the failure rate of an active unit is λ and a unit in standby mode does not fail. There is a single repairperson with repair rate μ . Give a queuing network that will model the behavior of this system. Draw the state diagram.
2. Components arrive at a repair facility with a constant rate λ , and the service time is exponentially distributed with mean $1/\mu$. The last step in the repair process is a quality-control inspection, and with probability p , the repair is considered inadequate, in which case the component will go back into the queue for repeated service. Determine the steady-state pmf of the number of components at the repair facility.
3. In the availability model of a standby redundant system of Example 8.9, we made an assumption that the failure rate of an unpowered spare is zero. Extend the model so that the failure rates of powered and unpowered units are λ_1 and λ_2 , respectively with $\lambda_1 \geq \lambda_2 \geq 0$. Obtain an expression for the steady-state availability. Verify that the expression derived yields the availability expression derived in Example 8.9 when $\lambda_2 = 0$. Similarly, verify that it gives the availability expression for a parallel redundant system when $\lambda_2 = \lambda_1 = \lambda$ as in Example 8.10 of Section 8.2.3.1.

8.3 OTHER SPECIAL CASES OF THE BIRTH-DEATH MODEL

We noted that the solution of the differential-difference equations (8.28) to obtain the probabilities $\pi_k(t)$ is a formidable task, in general. However, the calculation of the limiting probabilities $\pi_k = \lim_{t \rightarrow \infty} \pi_k(t)$ is relatively simple. In this section we consider several special cases of the birth-death model when the time-dependent probabilities $\pi_k(t)$ can be computed by simple techniques.

8.3.1 The Pure Birth Process

If the death rates $\mu_k = 0$ for all $k = 1, 2, \dots$, we have a **pure birth process**. If, in addition, we impose the condition of constant birth rates [i.e., $\lambda_k = \lambda$ ($k = 0, 1, 2, \dots$)], then we have the familiar Poisson process. The state diagram is shown in Figure 8.21. The equations (8.28) now reduce to

$$\begin{aligned}\frac{d\pi_0(t)}{dt} &= -\lambda\pi_0(t) & k = 0, \\ \frac{d\pi_k(t)}{dt} &= \lambda\pi_{k-1}(t) - \lambda\pi_k(t) & k \geq 1,\end{aligned}\tag{8.75}$$

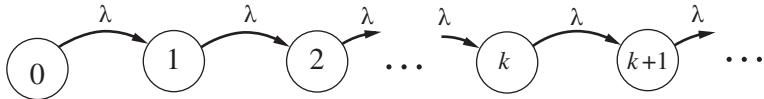


Figure 8.21. State diagram of the Poisson process

where we have assumed that the initial state $N(0) = 0$, so that

$$\pi_0(0) = 1, \quad \pi_k(0) = 0 \quad \text{for } k \geq 1. \quad (8.76)$$

One method of solving such differential equations is to use the Laplace transform, which simplifies the system of differential equations to a system of algebraic equations. The Laplace transform of $\pi_k(t)$, denoted by $\bar{\pi}_k(s)$, is defined in the usual way, namely:

$$\bar{\pi}_k(s) = \int_0^\infty e^{-st} \pi_k(t) dt,$$

and the Laplace transform of the derivative $d\pi_k/dt$ is given by (see Appendix D):

$$s\bar{\pi}_k(s) - \pi_k(0).$$

Now, taking Laplace transforms on both sides of equations (8.75), we get

$$s\bar{\pi}_0(s) - \pi_0(0) = -\lambda\bar{\pi}_0(s),$$

$$s\bar{\pi}_k(s) - \pi_k(0) = \lambda\bar{\pi}_{k-1}(s) - \lambda\bar{\pi}_k(s), \quad k \geq 1.$$

Using (8.76) and rearranging, we get

$$\bar{\pi}_0(s) = \frac{1}{s + \lambda}$$

and

$$\bar{\pi}_k(s) = \frac{\lambda}{s + \lambda} \bar{\pi}_{k-1}(s),$$

from which we have

$$\bar{\pi}_k(s) = \frac{\lambda^k}{(s + \lambda)^{k+1}}, \quad k \geq 0.$$

(This expression can also be obtained using the result of problem 5 at the end of Section 8.1)

In order to invert this transform, we note that if Y is a $(k+1)$ -stage Erlang random variable with parameter λ , then the LST of Y is (which is the same

as the Laplace transform of its density f_Y)

$$L_Y(s) = \bar{f}_Y(s) = \frac{\lambda^{k+1}}{(s + \lambda)^{k+1}}.$$

It follows that

$$\pi_k(t) = \frac{1}{\lambda} f_Y(t).$$

Therefore

$$\pi_k(t) = P(N(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k \geq 0; \quad t \geq 0. \quad (8.77)$$

Thus, $N(t)$ is Poisson distributed with parameter λt .

An alternative way to derive the expression for the state probability $\pi_k(t)$ for such an acyclic CTMC is to use the convolution integration approach (refer to problem 6 at the end of Section 8.1). For the special case here, we have

$$\pi_k(t) = \pi_k(0)e^{-\lambda t} + \int_0^t \pi_{k-1}(x)\lambda e^{-\lambda(t-x)} dx, \quad k = 1, 2, \dots$$

and

$$\pi_0(t) = \pi_0(0)e^{-\lambda t}.$$

Since $\pi_0(0) = 1$, we have $\pi_0(t) = e^{-\lambda t}$ and since $\pi_k(0) = 0$, we have

$$\pi_k(t) = \int_0^t \pi_{k-1}(x)\lambda e^{-\lambda(t-x)} dx.$$

It is easy to show that (8.77) is a solution to this equation.

It follows that the mean value function $m(t) = E[N(t)]$ is given by

$$m(t) = \sum_{k=0}^{\infty} k\pi_k(t) = \lambda t.$$

This can be seen as the expected reward rate at time t after assigning reward rate $r_k = k$ to state k .

The Poisson process can be generalized to the case where the birth rate λ is varying with time. Such a process is called a **nonhomogeneous Poisson process** (see Figure 8.22). The generalized version of equation (8.77) in this case is given by

$$\pi_k(t) = e^{-m(t)} \frac{[m(t)]^k}{k!}, \quad k \geq 0, \quad (8.78)$$

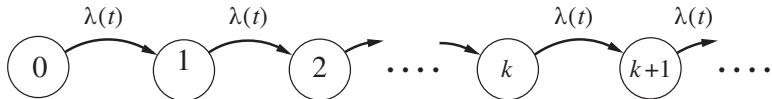


Figure 8.22. Nonhomogeneous Poisson process

where the mean-value function is

$$m(t) = \int_0^t \lambda(x) dx.$$

The nonhomogeneous Poisson process finds its use in reliability computations when the constant-failure-rate assumptions cannot be tolerated. Thus, for instance, if $\lambda(t) = c\alpha t^{\alpha-1}$ ($\alpha > 0$), then the time to failure of the component is Weibull distributed with parameters c and α , and if we assume that a component is instantaneously replaced by a new component, then the pmf of the number of failures $N(t)$ in the interval $(0, t]$ is

$$\pi_k(t) = P(N(t) = k) = e^{-ct^\alpha} \frac{(ct^\alpha)^k}{k!}, \quad k \geq 0.$$

Example 8.14 (Software Reliability Growth Models)

Consider a nonhomogeneous Poisson process (NHPP) proposed by Goel and Okumoto [GOEL 1979] as a model of software reliability growth during the testing phase. It is assumed that the number of failures $N(t)$ occurring in time interval $(0, t]$ has a pmf given by (8.78) with failure intensity

$$\lambda(t) = abe^{-bt}. \quad (8.79)$$

This implies that the expected number of software failures by time t is

$$m(t) = E[N(t)] = a(1 - e^{-bt}). \quad (8.80)$$

$m(t)$, the mean-value function, is a nondecreasing function of t . Since

$$\lim_{t \rightarrow \infty} m(t) = a,$$

it follows that the parameter a represents the expected number of software faults to be eventually detected if testing is carried out indefinitely. Using (8.80), the instantaneous failure intensity (8.79) can be rewritten as

$$\lambda(t) = b[a - m(t)],$$

which means that the failure intensity is proportional to $a - m(t)$, the expected number of undetected faults at t . It is clear that the parameter b can be interpreted as the failure occurrence rate per fault at an arbitrary testing time t .

Many commonly used NHPP software reliability growth models are obtained by choosing different failure intensities $\lambda(t)$. Thus, for instance, allowing time-dependent failure occurrence rate per fault $h(t)$, the failure intensity can be defined as

$$\lambda(t) = h(t)[a - m(t)]. \quad (8.81)$$

The generalized Goel–Okumoto model [GOEL 1985] uses the hazard rate of the Weibull distribution $h(t) = bct^{c-1}$ as a failure occurrence rate per fault at time t . It can capture constant ($c = 1$), monotonically increasing ($c > 1$), and monotonically decreasing ($c < 1$) failure occurrence rates per fault. The hazard rate of the log-logistic distribution $h(t) = [\lambda\kappa(\lambda t)^{\kappa-1}]/[1 + (\lambda t)^\kappa]$ has been used by Gokhale and Trivedi [GOKH 1998] to describe the failure occurrence rate per fault, which initially increases and then decreases ($\kappa > 1$). These models are examples of the so-called finite-failure NHPP models since they assume that the expected number of faults detected given infinite amount of testing time will be finite.

The Musa–Okumoto logarithmic Poisson execution time model [MUSA 1983] uses the failure intensity function

$$\lambda(t) = \frac{\gamma}{\gamma\theta t + 1},$$

which implies that the mean-value function of the NHPP

$$m(t) = \frac{\ln(\gamma\theta t + 1)}{\theta}$$

is an unbounded function as

$$\lim_{t \rightarrow \infty} m(t) = \infty.$$

Thus, this is an example of infinite-failure NHPP software reliability growth model that assumes that an infinite number of faults would be detected in infinite testing time. Notice that the time to failure distribution of a finite-failure NHPP is defective (for the definition of defective distribution see Section 3.4.9) while that of an infinite-failure NHPP model is nondefective. Hence the MTTF is infinite in the former case while it is finite in the latter case.

#

Problems

1. Set up the differential equation for $\pi_k(t)$ for the case of the nonhomogeneous Poisson process. Show that (8.78) is a solution to this equation.
2. * Consider the nonhomogeneous Poisson process with $\lambda(t) = cat^{\alpha-1}$ [PARZ 1962]. Let T_k denote the occupancy time in state k . Note that T_k is the interevent time of the process. Show that T_0 has the Weibull distribution with parameters c and α . Next show that T_1 does not, in general, have the Weibull distribution by first showing that the conditional pdf of T_1 given T_0 is

$$f_{T_1|T_0}(t | u) = e^{-m(t+u)+m(u)} \lambda(t+u),$$

and hence show that

$$f_{T_1}(t) = \int_0^\infty \lambda(t+u)\lambda(u)e^{-m(t+u)}du.$$

3. Write the mean-value function of a finite failure NHPP software reliability growth model as

$$m(t) = aF(t)$$

so that $F(t)$ has properties of a distribution function. Now derive equation (8.81) so that $h(t)$ is the hazard rate of $F(t)$. Plot $h(t)$ and $\lambda(t)$ vs. t for the Goel–Okumoto, generalized Goel–Okumoto, and log-logistic models. Use $a = 2$, $b = 0.001$, $c = 3$, $\kappa = 3$ and $\lambda = 0.01$.

8.3.2 Pure Death Processes

Another special case of a birth–death process occurs when the birth rates are all assumed to be zero; that is, $\lambda_k = 0$ for all k . The system starts in some state $n > 0$ at time $t = 0$ and eventually decays to state 0. Thus, state 0 is an absorbing state. We consider two special cases of interest.

8.3.2.1 Death Process with a Constant Rate. Besides $\lambda_i = 0$ for all i , we have $\mu_i = \mu$ for all i . This implies that the differential-difference equations (8.28) reduce to

$$\begin{aligned} \frac{d\pi_n(t)}{dt} &= -\mu\pi_n(t), & k = n, \\ \frac{d\pi_k(t)}{dt} &= -\mu\pi_k(t) + \mu\pi_{k+1}(t), & 1 \leq k \leq n-1, \\ \frac{d\pi_0(t)}{dt} &= \mu\pi_1(t), & k = 0, \end{aligned}$$

where we have assumed that the initial state $N(0) = n$, so that

$$\pi_n(0) = 1, \quad \pi_k(0) = 0, \quad 0 \leq k \leq n-1.$$

Taking Laplace transforms and rearranging, we reduce this system of equations to

$$\bar{\pi}_k(s) = \begin{cases} \frac{1}{s+\mu}, & k = n, \\ \frac{\mu}{s+\mu}\bar{\pi}_{k+1}(s), & 1 \leq k \leq n-1, \\ \frac{\mu}{s}\bar{\pi}_1(s), & k = 0, \end{cases}$$

so:

$$\bar{\pi}_k(s) = \frac{1}{\mu} \left(\frac{\mu}{s + \mu} \right)^{n-k+1}, \quad 1 \leq k \leq n.$$

If Y is an $(n - k + 1)$ -stage Erlang random variable with parameter μ , then its LST is known to be $L_Y(s) = \bar{f}_Y(s) = [\mu/(s + \mu)]^{n-k+1}$. It follows that

$$\begin{aligned}\pi_k(t) &= \frac{1}{\mu} f_Y(t) \\ &= e^{-\mu t} \frac{(\mu t)^{n-k}}{(n-k)!}, \quad 1 \leq k \leq n.\end{aligned}$$

Now, recalling that $\sum_{k=0}^n \pi_k(t) = 1$, we have

$$\begin{aligned}\pi_0(t) &= 1 - \sum_{k=1}^n \pi_k(t) \\ &= 1 - \sum_{k=1}^n e^{-\mu t} \frac{(\mu t)^{n-k}}{(n-k)!} \\ &= 1 - \sum_{k=0}^{n-1} e^{-\mu t} \frac{(\mu t)^k}{k!}.\end{aligned}\tag{8.82}$$

$\pi_0(t)$ is easily recognized to be the CDF of an n -stage Erlang random variable with mean n/μ .

Example 8.15

Consider a cold standby redundant system with n components, each with a constant failure rate μ . Then $\pi_0(t)$ above gives the distribution of the time to failure of such a system. This verifies our earlier result [see equation (3.74)].

#

Example 8.16

Consider the conditional response time distribution in an $M/M/1$ FCFS queue (see Section 8.2.1). If we assume that there are $n - 1$ jobs in the system when a new (tagged) job arrives, verify that its conditional response time distribution is given by $\pi_0(t)$ in equation (8.82).

#

8.3.2.2 Death Process with a Linear Rate. In this case we assume that $\lambda_i = 0$ for all i and $\mu_i = i\mu$, $i = 1, 2, \dots, n$. The state diagram is

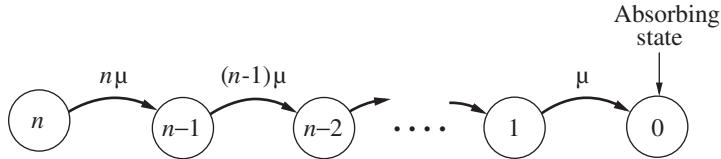


Figure 8.23. The state diagram of a death process with a linear death rate

given in Figure 8.23. The differential-difference equations in this case are given by

$$\begin{aligned}\frac{d\pi_n(t)}{dt} &= -n\mu\pi_n(t), \quad k = n, \\ \frac{d\pi_k(t)}{dt} &= -k\mu\pi_k(t) + (k+1)\mu\pi_{k+1}(t), \quad 1 \leq k \leq n-1, \\ \frac{d\pi_0(t)}{dt} &= \mu\pi_1(t), \quad k = 0,\end{aligned}$$

where we have assumed that the initial state $N(0) = n$, so that

$$\pi_n(0) = 1, \quad \pi_k(0) = 0, \quad 0 \leq k \leq n-1.$$

Using the method of Laplace transforms, we can obtain the solution to this system of equations as

$$\pi_k(t) = \binom{n}{k} (e^{-\mu t})^k (1 - e^{-\mu t})^{n-k}, \quad 0 \leq k \leq n, \quad t \geq 0,$$

which can be verified by differentiation. For a fixed t , this is recognized as a binomial pmf with parameters n and $p = e^{-\mu t}$.

Example 8.17

Consider a parallel redundant system with n components, each having a constant failure rate μ . If we let k denote the number of components operating properly, then the death process with linear rate describes the behavior of this system. The distribution of time to failure is then given by $\pi_0(t)$, and system reliability $R(t) = 1 - \pi_0(t) = 1 - (1 - e^{-\mu t})^n$. This agrees with the expression derived in Chapter 3 [equation (3.66)]. The distinction between the CTMC state diagram (Figure 8.23) and a distribution represented as a network of exponential stages (e.g., Figure 3.40) should be noted.

#

Example 8.18

The software reliability growth model proposed by Jelinski and Moranda [JELI 1972] is based on the following assumptions:

- The number of faults introduced initially into the software is fixed, say, n .
- At each failure occurrence, the underlying fault is removed immediately and no new faults are introduced.
- Failure rate is proportional to the number of remaining faults, that is, $\mu_i = i\mu$, $i = 1, 2, \dots, n$.

Clearly, this model can be described by the pure death process of Figure 8.23, where if k denotes the number of failures, state $i = n - k$ of the process at time t denotes the number of faults remaining at that time. The constant of proportionality μ denotes the failure intensity contributed by each fault, which means that all the remaining faults contribute the same amount to the failure intensity, that is, have the same size. The mean-value function is given by

$$m(t) = \sum_{k=0}^n k\pi_{n-k}(t) = n(1 - e^{-\mu t}). \quad (8.83)$$

This measure can be seen as the expected reward rate at time t after assigning reward rate $r_i = i$ to state i . ‡

If we consider a more general death process with variable death rates and the initial state $N(0) = n$, then we can show $\pi_0(t)$ to be the distribution function of a HYPO $(\mu_1, \mu_2, \dots, \mu_n)$ random variable. Reliability of a hybrid k -out-of- n system with perfect coverage can then be modeled by such a death process.

Problems

1. For the death process with linear death rate (see Figure 8.23), derive the formula for $\pi_k(t)$ given in the text starting from its differential equations and using the method of Laplace transforms. Also derive the formula for $\pi_k(t)$ using the convolution-integral approach developed in problem 6 at the end of Section 8.1.
2. Show that the mean value function of the Jelinski–Moranda model is as given on the right-hand side of equation (8.83).
3. Model the conditional response time distribution of an $M/M/n$ FCFS queue by a pure death process. Draw the state diagram and specify all the transition rates.

8.4 NON-BIRTH–DEATH PROCESSES

So far, we have discussed special cases of the birth–death process. Not all Markov chains of interest satisfy the restriction of nearest-neighbor-only transitions. In this section we study several examples of non-birth–death processes. These examples are divided into three subsections; in Section 8.4.1, we discuss availability models; in Section 8.4.2, performance models; and in Section 8.4.3, composite performance and availability models.

8.4.1 Availability Models

Availability models capture failure and repair behavior of systems and their components. States of the underlying Markov chain will be classified as up states or down states. We will discuss models that deal with hardware failures only as well as those that consider both hardware and software failures.

Example 8.19

The two-state model of component failure–repair (Example 8.6) assumed that the failure and the repair time distributions are both exponential. Assume now that the exponential failure law is reasonable, but the repair process can be broken down into two phases: (1) fault detection and location and, (2) actual repair. These two phases have exponential distributions with means $1/\mu_1$ and $1/\mu_2$, respectively. The overall repair time distribution is then hypoexponential (see Section 3.8). Since the sojourn time in the down state is two-stage hypoexponentially distributed (rather than exponentially distributed), the system being modeled is a semi-Markov process (not a homogeneous CTMC). However, by noting that the repair time distribution is an instance of a Coxian stage-type distribution (see Figure 5.11), we can transform the given system into a homogeneous CTMC. Define the following three states of the system:

- 0: the component is functioning properly
- 1: the component is in the detection–location phase
- 2: the component is in the final phase of repair.

The state diagram is given in Figure 8.24. Because of the transition from state 2 to state 0, this is not a birth–death process.

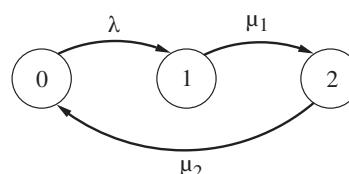


Figure 8.24. The state diagram for Example 8.19

We may compute the steady-state probabilities by first writing down the balance equations:

$$\lambda\pi_0 = \mu_2\pi_2, \quad \mu_1\pi_1 = \lambda\pi_0, \quad \mu_2\pi_2 = \mu_1\pi_1,$$

which yield the following relations:

$$\pi_1 = \frac{\lambda}{\mu_1}\pi_0, \quad \pi_2 = \frac{\mu_1}{\mu_2}\pi_1 = \frac{\mu_1}{\mu_2}\frac{\lambda}{\mu_1}\pi_0 = \frac{\lambda}{\mu_2}\pi_0.$$

Now, since

$$\pi_0 + \pi_1 + \pi_2 = 1,$$

we have

$$\pi_0 = \frac{1}{1 + \frac{\lambda}{\mu_1} + \frac{\lambda}{\mu_2}}.$$

Thus, the steady-state availability A is given by

$$A = \pi_0 = \frac{1}{1 + \lambda(\frac{1}{\mu_1} + \frac{1}{\mu_2})}.$$

This result can be extended to the case of a k -stage hypoexponential repair time distribution with parameters $\mu_1, \mu_2, \dots, \mu_k$ with the following result:

$$A = \frac{1}{1 + \lambda \left(\frac{1}{\mu_1} + \frac{1}{\mu_2} + \cdots + \frac{1}{\mu_k} \right)}.$$

If we denote the average total repair time by $1/\mu$, then from the formula of $E[X]$ for a hypoexponentially distributed random variable X , we have

$$\frac{1}{\mu} = \sum_{i=1}^k \frac{1}{\mu_i}.$$

With this value of μ , we can use formula (8.57) for steady-state availability derived from the two-state model (Example 8.6), even when the repair times are hypoexponentially distributed.

‡

Example 8.20 (A Preventive Maintenance Model)

It is known that preventive maintenance (PM) does not help in case the failure rate of a device is constant. The device time to failure distribution must have an increasing failure rate for PM to increase availability (or reliability). Since the hypoexponential distribution is IFR (refer to Chapter 3), we can use such a distribution of time to failure to demonstrate the efficacy of PM. Assume that device time to failure is two-stage hypoexponential with rates λ_1 and λ_2 . Assume also that the time to repair is exponentially distributed with rate μ . Further assume that an inspection is

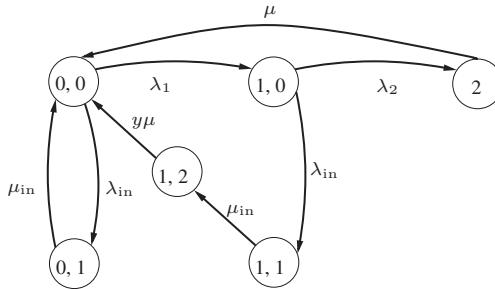


Figure 8.25. CTMC for preventive maintenance model

triggered after a mean duration $1/\lambda_{\text{in}}$, and takes an average time of $1/\mu_{\text{in}}$. After an inspection is completed, no action is taken if the device is found to be in the first stage of its lifetime. On the other hand, if the device is found to be in the second stage of its lifetime, a preventive maintenance is carried out. We assume that the time to carry out repair is y times the time to carry out preventive maintenance. The resulting CTMC is shown in Figure 8.25. This type of PM is called condition based maintenance [HOSS 2000].

Writing down and solving the steady-state balance equations, we find

$$\begin{aligned}
 (\lambda_1 + \lambda_{\text{in}})\pi_{00} &= y\mu\pi_{12} + \mu\pi_2 + \mu_{\text{in}}\pi_{01} \\
 \mu_{\text{in}}\pi_{01} &= \lambda_{\text{in}}\pi_{00} \quad \Rightarrow \quad \pi_{01} = \frac{\lambda_{\text{in}}}{\mu_{\text{in}}}\pi_{00} \\
 (\lambda_2 + \lambda_{\text{in}})\pi_{10} &= \lambda_1\pi_{00} \quad \Rightarrow \quad \pi_{10} = \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}}\pi_{00} \\
 y\mu\pi_{12} &= \mu_{\text{in}}\pi_{11} \quad \Rightarrow \quad \pi_{12} = \frac{\mu_{\text{in}}}{y\mu}\pi_{11} \\
 \mu_{\text{in}}\pi_{11} &= \lambda_{\text{in}}\pi_{10} \quad \Rightarrow \quad \pi_{11} = \frac{\lambda_{\text{in}}}{\mu_{\text{in}}}\pi_{10} \\
 \mu\pi_2 &= \lambda_2\pi_{10} \quad \Rightarrow \quad \pi_2 = \frac{\lambda_2}{\mu}\pi_{10}.
 \end{aligned}$$

Hence,

$$\pi_2 = \frac{\lambda_2}{\mu} \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} \pi_{00}$$

$$\pi_{11} = \frac{\lambda_{\text{in}}}{\mu_{\text{in}}} \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} \pi_{00}$$

$$\pi_{12} = \frac{\mu_{\text{in}}}{y\mu} \frac{\lambda_{\text{in}}}{\mu_{\text{in}}} \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} \pi_{00}.$$

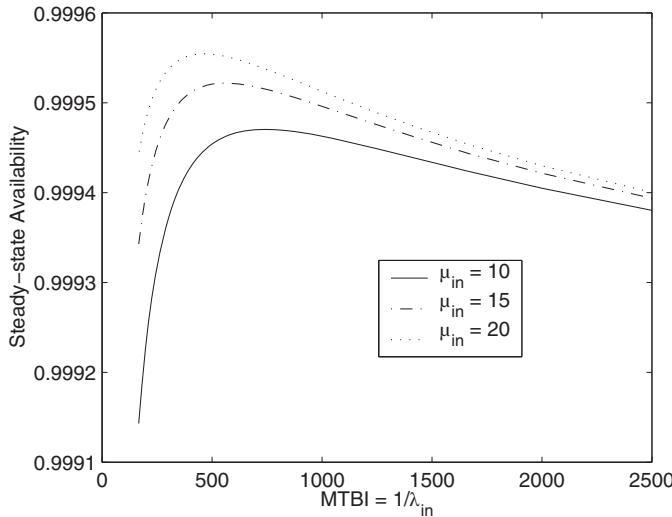


Figure 8.26. Steady-state availability of the preventive maintenance model with different μ_{in}

Thus,

$$\pi_{00} = \frac{1}{1 + (1 + \frac{\lambda_2}{\mu} + \frac{\lambda_{\text{in}}}{\mu_{\text{in}}} + \frac{\lambda_{\text{in}}}{y_\mu}) \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} + \frac{\lambda_{\text{in}}}{\mu_{\text{in}}}}. \quad (8.84)$$

Since only up states are $(0, 0)$ and $(1, 0)$, we obtain an expression for the steady-state availability:

$$A = \frac{1 + \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}}}{1 + \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} + \frac{\lambda_1 \lambda_2}{\mu(\lambda_2 + \lambda_{\text{in}})} + \frac{\lambda_{\text{in}}}{\mu_{\text{in}}} + \frac{\lambda_{\text{in}}}{\mu_{\text{in}}} \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}} + \frac{\lambda_{\text{in}}}{y\mu} \frac{\lambda_1}{\lambda_2 + \lambda_{\text{in}}}} \quad (8.85)$$

In Figure 8.26, we plot the steady-state availability as a function of the mean time between inspections $\text{MTBI} = 1/\lambda_{\text{in}}$. We use $\lambda_1 = 0.0001$, $\lambda_2 = 0.0005$, $\mu = 0.1$ and $y = 5$. We use several different values of the time to carry out the inspection. Note that the steady-state availability reaches a maximum at $\text{MTBI} = 714.29$ for $\mu_{\text{in}} = 10$. In Problem 7 at the end of this section, you are asked to obtain an expression for the optimal value of λ_{in}^* so as to maximize the steady-state availability.

#

Example 8.21

We return to the parallel redundant system with a single shared repair facility (Section 8.2.3.1). Figure 8.27 shows the model of a two-component system. State i denotes that i components are working [note that the indices have been changed; e.g., state 0 was labeled state M ($M = 2$ in this example) in Figure 8.16]. We consider several variations wherein we introduce non-zero detection delay for a fault and then

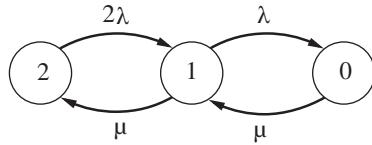


Figure 8.27. Two-component system availability model

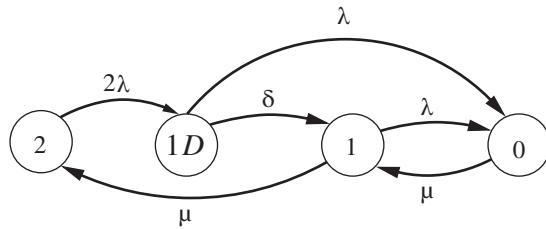


Figure 8.28. Two-component system with fault detection delay

imperfect coverage for faults. The solution of the simple model of Figure 8.27 was presented in Section 8.2.3.1 so that downtime in minutes per year is given by

$$D = 8760 \times 60 \times \pi_0 = \frac{8760 \times 60}{1 + \frac{2\lambda}{\mu} + \frac{2\lambda^2}{\mu^2}} \cdot \frac{2\lambda^2}{\mu^2} = \frac{8760 \times 60}{1 + \frac{\mu}{\lambda} + \frac{\mu^2}{2\lambda^2}}.$$

Now we introduce detection delay that is exponentially distributed with mean $1/\delta$. The state diagram is shown in Figure 8.28.

We have introduced a new state labeled $1D$ that represents the detection stage. The transition from state $1D$ to state 1 (at rate δ) indicates the detection of the fault. During the time the fault is being detected, should the second component fail (at rate λ), the system is assumed to fail. Such a fault has been called a *near-coincident fault*.

After writing the steady-state balance equations and solving these equations, we obtain the following expressions for the steady-state probabilities:

$$\begin{aligned}\pi_0 &= \frac{1}{E}, \\ \pi_1 &= \frac{\mu(\lambda + \delta)}{\lambda(\lambda + \mu + \delta)E}, \\ \pi_{1D} &= \frac{\mu^2}{\lambda(\lambda + \mu + \delta)E}, \\ \pi_2 &= \frac{\mu^2(\lambda + \delta)}{2\lambda^2(\lambda + \mu + \delta)E},\end{aligned}$$

where

$$E = 1 + \frac{\mu(\lambda + \delta)}{\lambda(\lambda + \mu + \delta)} + \frac{\mu^2}{\lambda(\lambda + \mu + \delta)} + \frac{\mu^2(\lambda + \delta)}{2\lambda^2(\lambda + \mu + \delta)}.$$

First assume that the delay state $1D$ is a system down state; then the steady-state unavailability is

$$U(\delta) = \pi_0 + \pi_{1D} = \frac{\lambda(\lambda + \mu + \delta) + \mu^2}{\lambda(\lambda + \mu + \delta)E},$$

and the downtime in minutes per year is

$$D(\delta) = U(\delta) \times 8760 \times 60.$$

The equivalent failure and repair rates in this case are given by

$$\lambda_{eq} = \frac{2\lambda\pi_2 + \lambda\pi_1}{\pi_2 + \pi_1} = \lambda \left(1 + \frac{\pi_2}{\pi_2 + \pi_1} \right)$$

and

$$\mu_{eq} = \frac{\delta\pi_{1D} + \mu\pi_0}{\pi_{1D} + \pi_0}.$$

Verify that

$$U(\delta) = \frac{\lambda_{eq}}{\lambda_{eq} + \mu_{eq}}.$$

Quite often in practice if the actual delay in state $1D$ is less than some threshold, say, t_{th} seconds, the state is not considered to be down. To capture this behavior, we observe that the probability that the sojourn time in state $1D$ is less than t_{th} seconds is $1 - e^{-\delta t_{th}}$. Thus, if we assign a reward rate $e^{-\delta t_{th}}$ to state $1D$, reward rate 1 to the state 0, and reward rate 0 to the remaining states, we will get the modified unavailability expression as

$$U(\delta, t_{th}) = \pi_0 + e^{-\delta t_{th}} \pi_{1D}.$$

Thus state $1D$ is considered down if its sojourn time exceeds t_{th} . From this we can get the downtime expression

$$D(\delta, t_{th}) = U(\delta, t_{th}) \times 8760 \times 60.$$

In Figure 8.29 we have plotted $D(\delta)$, $D(\delta, t_{th})$, and D as functions of $1/\delta$ (in seconds) for $1/\lambda = 10,000$ h and $1/\mu = 2$ h.

#

Example 8.22

We consider another variation of the parallel redundant system of Figure 8.27. As the state diagram (Figure 8.30) shows, when one of the components (say, a processor) fails, the system enters state 1 with probability c and enter state $1C$ with probability $1 - c$. Quantity c is known as the *coverage factor* or *coverage probability*. State $1C$ incurs a reboot delay with mean $1/\beta$. Note that although this is a hardware availability model, aspects of software unreliability are included in the sense that the unreliability of recovery software is likely to be the cause of imperfect coverage.

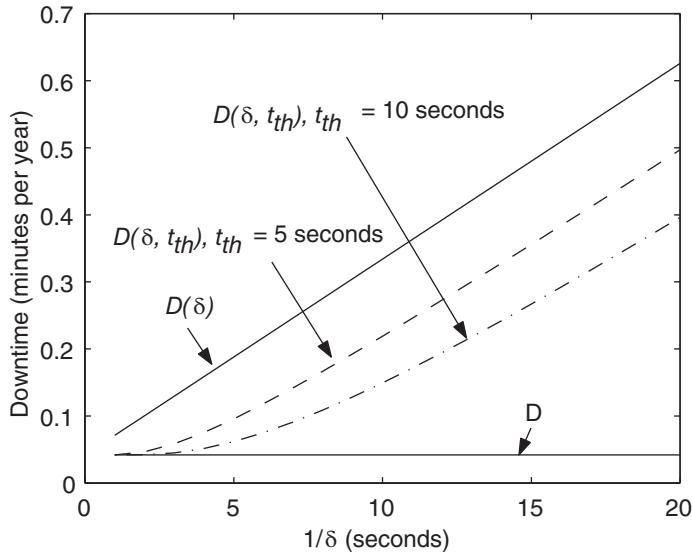


Figure 8.29. Downtime of two-component system with fault detection delay

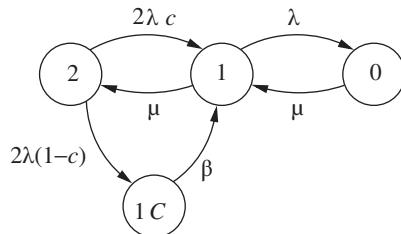


Figure 8.30. Two-component availability model with imperfect coverage

Solving steady-state balance equations, we can obtain steady-state probabilities:

$$\pi_0 = \frac{\lambda}{\mu E}$$

$$\pi_1 = \frac{1}{E}$$

$$\pi_{1C} = \frac{\mu(1-c)}{\beta E}$$

$$\pi_2 = \frac{\mu}{2\lambda E},$$

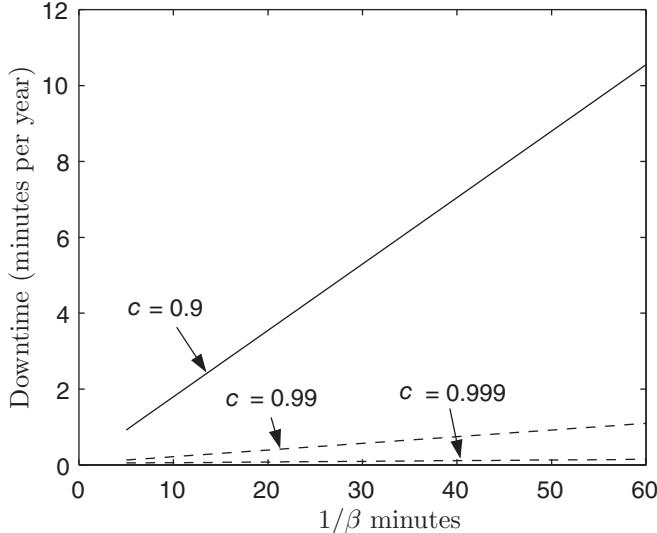


Figure 8.31. Downtime due to imperfect coverage

where

$$E = \frac{\lambda}{\mu} + 1 + \frac{\mu(1-c)}{\beta} + \frac{\mu}{2\lambda}.$$

Assume that the reboot state $1C$ is a system down state; then the steady-state unavailability is

$$U(\beta, c) = \pi_0 + \pi_{1C} = \frac{\lambda\beta + \mu^2(1-c)}{\mu\beta E}$$

and the downtime in minutes per year is

$$D(\beta, c) = U(\beta, c) \times 8760 \times 60.$$

In Figure 8.31 we have plotted $D(\beta, c)$ as a function of $1/\beta$ (in minutes) for $1/\lambda = 10,000$ h and $1/\mu = 2$ h.

†

Example 8.23

We now combine the deleterious effects of the detection delay and imperfect coverage as shown in the CTMC model of Figure 8.32. States $1D$ and $1C$ are both delay states. The delay in state $1D$ will be of the order of seconds, while that in state $1C$ will be of the order of minutes.

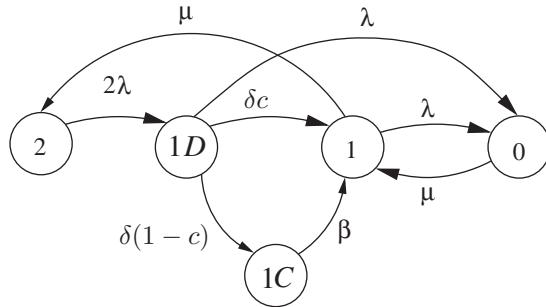


Figure 8.32. Two-component availability model with imperfect coverage and detection delay

Solving steady-state balance equations, we can obtain steady-state probabilities:

$$\begin{aligned}\pi_2 &= \frac{\mu}{2\lambda} \cdot \frac{1}{F} \\ \pi_{1D} &= \frac{\mu}{\delta + \lambda} \cdot \frac{1}{F} \\ \pi_0 &= \left(\frac{\lambda}{\mu} + \frac{\lambda}{\delta + \lambda} \right) \cdot \frac{1}{F} \\ \pi_{1C} &= \frac{\delta(1 - c)\mu}{\beta(\delta + \lambda)} \cdot \frac{1}{F} \\ \pi_1 &= \frac{1}{F},\end{aligned}$$

where

$$F = \frac{\mu}{2\lambda} + \frac{\lambda + \mu}{\delta + \lambda} + 1 + \frac{\lambda}{\mu} + \frac{\delta(1 - c)\mu}{\beta(\delta + \lambda)}.$$

Assume both states $1D$ and $1C$ are system down states; then the steady-state unavailability is

$$U(\delta, \beta, c) = \pi_0 + \pi_{1C} + \pi_{1D} = \frac{\lambda\beta(\delta + \lambda + \mu) + \mu^2(\delta(1 - c) + \beta)}{\mu\beta(\delta + \lambda)F}$$

and the downtime in minutes per year is

$$D(\delta, \beta, c) = U(\delta, \beta, c) \times 8760 \times 60.$$

In Figure 8.33 we have plotted $D(\delta, \beta, c)$ as a function of $1/\delta$ (in seconds) for $1/\lambda = 10,000$ h, $1/\mu = 2$ h and $1/\beta = 5$ min.

#

Example 8.24

We return to the workstation and file server (WFS) example that we considered in Chapters 3, 4, and 6 with two workstations and one file server. As we did in Example

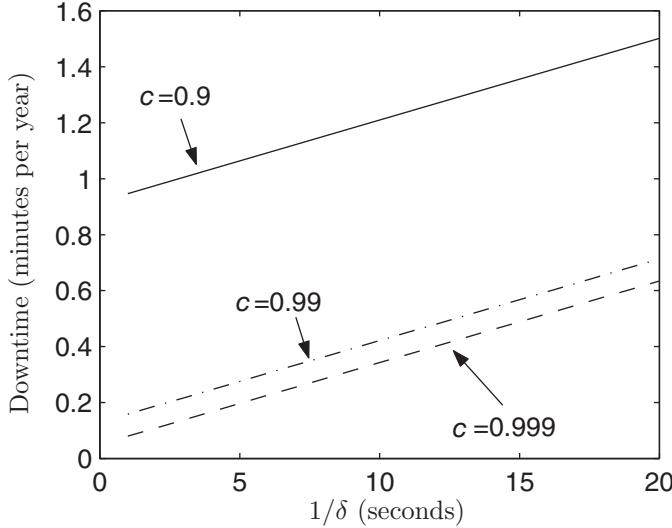


Figure 8.33. Downtime due to imperfect coverage and detection delay

6.12, we study the availability of the system but now we introduce repair dependency, in which a single repair facility is shared by all components. Assume that the file server has a preemptive repair priority over a workstation and once the system is down, no more failures can take place. The homogeneous CTMC for this system is shown in Figure 8.34. In this figure the label (i, j) of each state is interpreted as follows: i represents the number of workstations that are still functioning and $j = 1$ if the file server is functioning. Solving this Markov chain involves computing either $\pi_{i,j}(t)$, the probability of being in state (i, j) at time t , or $\pi_{i,j}$, the steady-state probability of being in state (i, j) .

The generator matrix Q is given by

$$Q = \begin{bmatrix} -(\lambda_f + 2\lambda_w) & \lambda_f & 2\lambda_w & 0 & 0 \\ \mu_f & -\mu_f & 0 & 0 & 0 \\ \mu_w & 0 & -(\mu_w + \lambda_f + \lambda_w) & \lambda_f & \lambda_w \\ 0 & 0 & \mu_f & -\mu_f & 0 \\ 0 & 0 & \mu_w & 0 & -\mu_w \end{bmatrix}.$$

The balance equations for computing the steady-state probabilities are

$$(\lambda_f + 2\lambda_w)\pi_{2,1} = \mu_w\pi_{1,1} + \mu_f\pi_{2,0} \quad (8.86)$$

$$(\lambda_w + \lambda_f + \mu_w)\pi_{1,1} = \mu_w\pi_{0,1} + \mu_f\pi_{1,0} + 2\lambda_w\pi_{2,1} \quad (8.87)$$

$$\mu_w\pi_{0,1} = \lambda_w\pi_{1,1} \quad (8.88)$$

$$\mu_f\pi_{2,0} = \lambda_f\pi_{2,1} \quad (8.89)$$

$$\mu_f\pi_{1,0} = \lambda_f\pi_{1,1}. \quad (8.90)$$

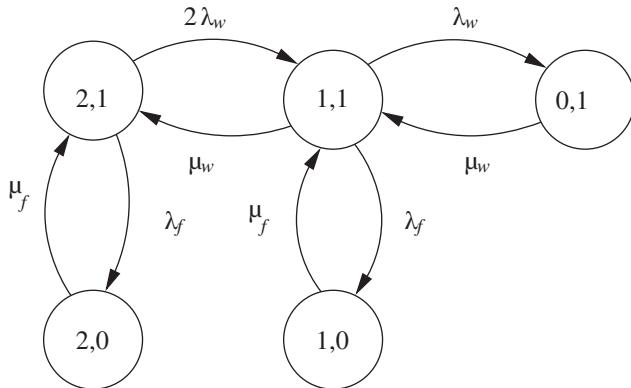


Figure 8.34. CTMC availability model for the WFS example

Solving the above equations for $\pi_{2,1}$ and $\pi_{1,1}$ in terms of $\pi_{0,1}$ we get

$$\pi_{2,1} = \frac{\mu_w^2}{2\lambda_w^2} \pi_{0,1}$$

$$\pi_{1,1} = \frac{\mu_w}{\lambda_w} \pi_{0,1},$$

where

$$\pi_{0,1} = \frac{1}{1 + \left(\frac{\lambda_f}{\mu_f} + 1 \right) \left(\frac{\mu_w^2}{2\lambda_w^2} + \frac{\mu_w}{\lambda_w} \right)}.$$

Hence the steady-state availability is given by

$$A = \pi_{2,1} + \pi_{1,1}.$$

If transient probabilities were computed, perhaps using a software package like SHARPE [SAHN 1996], we could then compute the instantaneous availability as

$$A(t) = \pi_{2,1}(t) + \pi_{1,1}(t).$$

The interval availability is given by

$$A_I(t) = \frac{L_{2,1}(t) + L_{1,1}(t)}{t},$$

where $L_{i,j}(t) = \int_0^t \pi_{i,j}(u) du$ is the expected total time spent by the CTMC in state (i, j) during the interval $(0, t]$.

#

Example 8.25

Consider a 2-node cluster where both hardware and Operating System software(OS) failures may occur [HUNT 1999]. The node hardware fails at the constant rate λ and the OS fails at the constant rate λ_{OS} . We assume here that hardware failures are permanent and hence require a repair or replacement action while OS failures are cleared by a reboot. Repair or reboot takes place at rates μ and β for the hardware and OS respectively. A node is considered down when either the OS or the hardware has failed. The cluster is down when both nodes have failed. In case of a hardware failure in one node and an OS failure in the other, the OS is always recovered first.

The CTMC corresponding to this cluster system is shown in Figure 8.35. In state 1, both nodes and their OSs are functioning properly. In state 2, one of the nodes has a hardware failure and in state 3, both the nodes have hardware failure. In state 4, one of the OSs has failed while in state 5, both OSs have failed. In state 6, one node has a hardware failure while the other has an OS failure. For the steady state balance equations we have

$$\pi_1(2\lambda_{OS} + 2\lambda) = \pi_2\mu + \pi_4\beta,$$

$$\pi_2(\lambda + \mu + \lambda_{OS}) = \pi_6\beta + \pi_3\mu + \pi_1 \cdot 2\lambda,$$

$$\pi_3\mu = \pi_2\lambda,$$

$$\pi_4(\lambda_{OS} + \beta + \lambda) = \pi_1 \cdot 2\lambda_{OS} + \pi_5 \cdot 2\beta,$$

$$\pi_5 \cdot 2\beta = \pi_4\lambda_{OS}, \quad \text{and}$$

$$\pi_6\beta = \pi_4\lambda + \pi_2\lambda_{OS}.$$

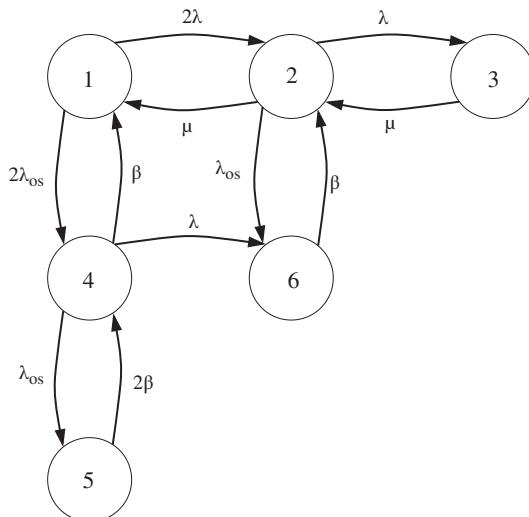


Figure 8.35. CTMC for the 2-node cluster system

These equations can be solved, in conjunction with $\sum_{i=1}^6 \pi_i = 1$, to obtain the steady state probabilities as shown below:

$$\begin{aligned}\pi_1 &= \frac{\lambda + \beta}{E}, & \pi_2 &= \frac{2\left(\frac{\lambda}{\mu}\right)(\lambda_{OS} + \lambda + \beta)}{E}, & \pi_3 &= \frac{2\left(\frac{\lambda}{\mu}\right)^2 (\lambda_{OS} + \lambda + \beta)}{E}, \\ \pi_4 &= \frac{2\lambda_{OS}}{E}, & \pi_5 &= \frac{(\lambda_{OS}^2/\beta)}{E},\end{aligned}$$

and

$$\pi_6 = \frac{2\left(\frac{\lambda_{OS}}{\beta}\right) \left[\lambda + \left(\frac{\lambda}{\mu}\right) (\lambda_{OS} + \lambda + \beta) \right]}{E},$$

where

$$E = \lambda + \beta + 2\lambda_{OS} \left(1 + \frac{\lambda_{OS}}{2\beta} + \frac{\lambda}{\beta} \right) + 2\frac{\lambda}{\mu}(\lambda_{OS} + \lambda + \beta) \left(1 + \frac{\lambda}{\mu} + \frac{\lambda_{OS}}{\beta} \right).$$

The steady state availability can be written as

$$\begin{aligned}A &= \pi_1 + \pi_2 + \pi_4 \\ &= \frac{\lambda + \beta + 2\left(\frac{\lambda}{\mu}\right)(\lambda_{OS} + \lambda + \beta) + 2\lambda_{OS}}{E}.\end{aligned}$$

#

Example 8.26 [GARG 1999]

In this example, we consider both hardware and (application) software failures. We consider a Web server software, that fails at the rate γ_p , running on a machine (node) that fails independently at the rate γ_m . An automatic failure detection mechanism based on polling is installed. Assume that the mean time to detect server process failure is δ_p^{-1} and the mean time to detect machine failure is δ_m^{-1} . Furthermore, when the machine is detected to have failed, the server process is started on another machine, if available. The mean restart time of a machine is τ_m^{-1} . When only the server process is detected to have failed, it is automatically restarted on the same machine. For details on process and machine failure detection and recovery, see papers by Garg *et al.* and Huang and Kintala [GARG 1999, HUAN 1993]. The mean restart time of the server software is τ_p^{-1} . Typically, $\tau_p > \tau_m$. There is a small probability $1 - c$ that the process restart on the same machine is unsuccessful, in which case it is restarted on another machine, if available. Such a scheme of automatic restart after failures is also called “cold replication” [HUAN 1993]. The Web server is considered available when the server process as well as the machine it is running on are up. We calculate the steady-state availability of the server, assuming that no further failures can occur after a failure of either the process or the machine until it has been dealt with.

Figure 8.36 shows the homogeneous CTMC for a Web server with cold replication using one spare machine. The states are labeled using the notation $\binom{P_p, P_m}{S_p, S_m}$, where

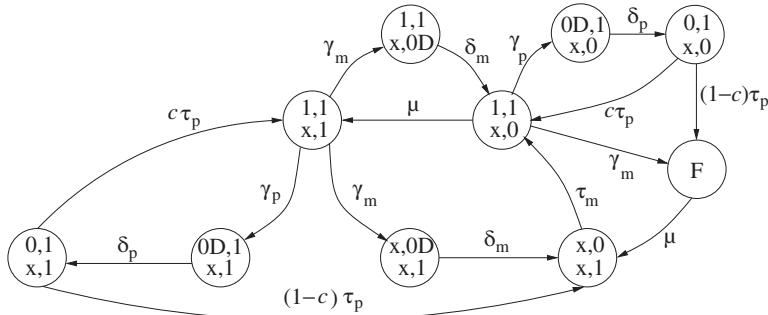


Figure 8.36. CTMC model for Web server with cold replication

P_p and P_m denote the status of the primary server process and machine, and S_p and S_m denote the status of the spare server process and machine, respectively. A status of “1” indicates that the process or machine is up, a status of “0” indicates that the process or machine is down, and “0D” indicates that the process or machine has failed but the failure is yet to be detected. A status of “x” indicates that the status is of no consequence (don’t care) and is used to indicate the status of the server process on the spare machine. To simplify our discussion we relabel the states as shown in Table 8.4. The state space is denoted by $I = \{1, 2, \dots, 10\}$. The Web server processes requests only in states 1, 6 and 7. State 6 represents failure of the spare machine. States 8 and 9 denote primary process failures when the spare machine is down. State 10 denotes the state when both the primary and spare machines are down. Whenever a machine has crashed (states 7 and 10), a more elaborate recovery with rate μ is required. The transition from state 5 to state 7 with rate τ_m denotes the starting of the process on a machine. This happens as a result of a failover (after reaching state 5 from state 4), or after recovery of a machine after complete crash (after reaching state 5 from state 10). Note that although the server is unavailable in states 2, 3 and 8, the failure is not observable until it is detected.

The steady-state probabilities π_i , $i = 1, 2, \dots, 10$ can be derived as

$$\begin{aligned} \pi_1 &= \frac{1}{E}, & \pi_2 &= \frac{1}{E} \cdot \frac{\gamma_p}{\delta_p}, & \pi_3 &= \frac{1}{E} \cdot \frac{\gamma_m}{\delta_m}, & \pi_4 &= \frac{1}{E} \cdot \frac{\gamma_p}{\tau_p} \\ \pi_5 &= \frac{1}{E} \cdot \frac{[\gamma_m + (1-c)\gamma_p][\mu + 2\gamma_m + (1-c)\gamma_p]}{\mu\tau_m} \\ \pi_6 &= \frac{1}{E} \cdot \frac{\gamma_m}{\delta_m}, & \pi_7 &= \frac{1}{E} \cdot \frac{2\gamma_m + (1-c)\gamma_p}{\mu} \\ \pi_8 &= \frac{1}{E} \cdot \frac{[2\gamma_m + (1-c)\gamma_p]\gamma_p}{\mu\delta_p}, & \pi_9 &= \frac{1}{E} \cdot \frac{[2\gamma_m + (1-c)\gamma_p]\gamma_p}{\mu\tau_p} \\ \pi_{10} &= \frac{1}{E} \cdot \frac{[2\gamma_m + (1-c)\gamma_p][\gamma_m + (1-c)\gamma_p]}{\mu^2} \end{aligned}$$

TABLE 8.4. State indices

State name	State index
1, 1	1
x, 1	
0D, 1	2
x, 1	
x, 0D	3
x, 1	
0, 1	4
x, 1	
x, 0	5
x, 1	
1, 1	6
x, 0D	
1, 1	7
x, 0	
0D, 1	8
x, 0	
0, 1	9
x, 0	
F	10

where

$$\begin{aligned}
 E = & 1 + \frac{\gamma_p}{\delta_p} + \frac{2\gamma_m}{\delta_m} + \frac{\gamma_p}{\tau_p} + \frac{[\gamma_m + (1 - c)\gamma_p][\mu + 2\gamma_m + (1 - c)\gamma_p]}{\mu\tau_m} \\
 & + \frac{2\gamma_m + (1 - c)\gamma_p}{\mu} \cdot [1 + \frac{\gamma_p}{\delta_p} + \frac{\gamma_p}{\tau_p}] \\
 & + \frac{[2\gamma_m + (1 - c)\gamma_p][\gamma_m + (1 - c)\gamma_p]}{\mu^2}.
 \end{aligned}$$

The steady-state availability is given by

$$A = \pi_1 + \pi_6 + \pi_7 = \frac{1}{E} \cdot \left[1 + \frac{\gamma_p}{\delta_p} + \frac{2\gamma_m + (1 - c)\gamma_p}{\mu} \right].$$

Example 8.27

We now return to the two-component system (Example 8.22) but consider one component as active and the other as a standby (spare) unit. The failure rates of the active unit and the standby unit are different, and also the effect of failure of the standby unit is different from that of the active unit. The state diagram is shown in Figure 8.37. Initially both units are working and the system is in state (1,1). Let the time to failure of an active unit, the time to failure of a standby unit, and the time to restoration of a failed unit be exponentially distributed with parameters λ , λ_s , and μ , respectively. When the active unit fails, with probability c a protection switch successfully restores service by switching in the standby unit, and the system enters state (1,0). With probability $1 - c$ the protection switch fails to cover the failure of the active unit and the system enters state 1C. The failure of the standby unit while the active unit is still working is detected immediately with probability c_s , and when this happens, the system enters state (1,0). If the failure of the standby unit is not detected (with probability $1 - c_s$), the system enters state 1D. There is a latent fault in the spare unit when the system is in state 1D. If a unit failure occurs when the system is in one of the states: 1C, (1,0), or 1D, the system fails and enters state (0,0).

Solving the steady-state balance equations, we obtain

$$\begin{aligned}\pi_{1,0} &= \pi_{1,1} \frac{\lambda + \lambda_s}{\mu} \\ \pi_{1C} &= \pi_{1,1} \frac{\lambda(1 - c)}{\beta + \lambda_s} \\ \pi_{1D} &= \pi_{1,1} \frac{\lambda_s(1 - c_s)}{\lambda}\end{aligned}$$

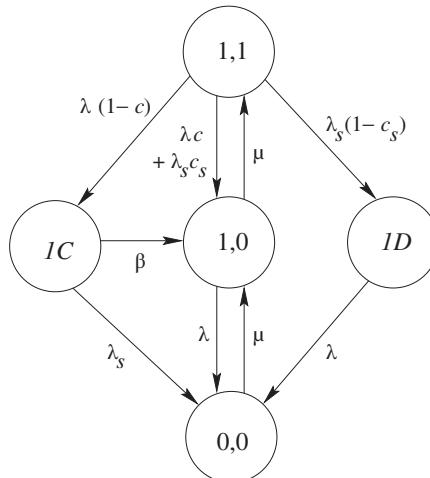


Figure 8.37. Differentiating between failures of the active and spare units

$$\pi_{0,0} = \pi_{1,1} \left(\frac{\lambda(1-c)\lambda_s}{\mu(\beta + \lambda_s)} + \frac{(\lambda + \lambda_s)\lambda}{\mu^2} + \frac{\lambda_s(1-c_s)}{\mu} \right),$$

where

$$\pi_{1,1} = \frac{1}{1 + \frac{\lambda + \lambda_s}{\mu} \left(1 + \frac{\lambda}{\mu} \right) + \frac{\lambda(1-c)}{\beta + \lambda_s} \left(1 + \frac{\lambda_s}{\mu} \right) + \lambda_s(1-c_s) \left(\frac{1}{\lambda} + \frac{1}{\mu} \right)}.$$

Now let us consider a routine diagnostic that is run every T time units, intended to detect the latent fault of the standby unit. While unit (active and standby) failure and restoration times are exponentially distributed, the routine diagnostic time interval is not. Thus the underlying stochastic process is not a continuous time Markov chain. The transition from state $1D$ to state $(1,0)$ has a rate that depends on how long the system has been in state $1D$, but not on which states the system had been in before it got there. Such a process is called a **semi-Markov process** (SMP). The model for the system with the diagnostic routine, shown in Figure 8.38, is called a **semi-Markov chain**.

To solve this model, we could crudely approximate the time to the next diagnostic to be exponentially distributed with mean $T/2$. Solving the steady-state balance equations with this approximation, we obtain

$$\pi_{1,0} = \pi_{1,1} \frac{\lambda + \lambda_s}{\mu}$$

$$\pi_{1C} = \pi_{1,1} \frac{\lambda(1-c)}{\beta + \lambda_s}$$

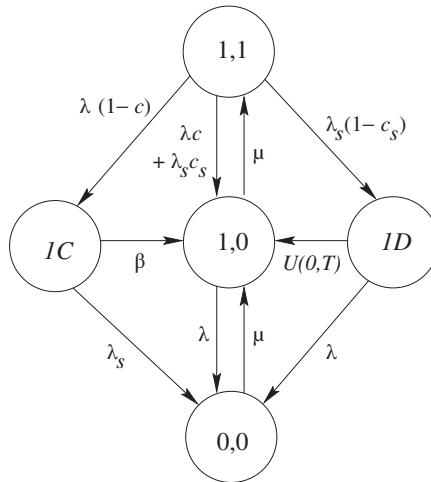


Figure 8.38. A semi-Markov model

$$\pi_{1D} = \pi_{1,1} \frac{\lambda_s(1 - c_s)}{\lambda + \frac{2}{T}}$$

$$\pi_{0,0} = \pi_{1,1} \left(\frac{\lambda(1 - c)\lambda_s}{\mu(\beta + \lambda_s)} + \frac{(\lambda + \lambda_s)\lambda}{\mu^2} + \frac{\lambda_s(1 - c_s)\lambda}{\mu(\lambda + \frac{2}{T})} \right),$$

where

$$\pi_{1,1} = \frac{1}{1 + \frac{\lambda + \lambda_s}{\mu} \left(1 + \frac{\lambda}{\mu} \right) + \frac{\lambda(1 - c)}{\beta + \lambda_s} \left(1 + \frac{\lambda_s}{\mu} \right) + \frac{\lambda_s(1 - c_s)T}{\lambda T + 2} \left(1 + \frac{\lambda}{\mu} \right)}.$$

A better approach would be to take the time to the next diagnostic to be uniformly distributed over $[0, T]$, resulting in a semi-Markov chain. This is indicated in Figure 8.38 by the transition labeled $U(0, T)$. One way to describe an SMP is to think of transitions as occurring in two stages. In the first stage, the SMP stays in state i for an amount of time described by $H_i(t)$, the sojourn time distribution in state i . In the second stage, the SMP moves from state i to state j with probability p_{ij} . Thus in the two-stage method, the SMP is described by a transition probability matrix P and the vector of sojourn time distributions, $\mathbf{H}(t)$.

Note that for all states other than state $1D$, the sojourn time distribution is exponential with rate equal to the net rate out of that state. For state $1D$, the sojourn time is the minimum of $\text{EXP}(\lambda)$ and $U(0, T)$ random variables. From these observations, we have

$$H_{1,1}(t) = 1 - e^{-(\lambda + \lambda_s)t}$$

$$H_{1C}(t) = 1 - e^{-(\beta + \lambda_s)t}$$

$$H_{1,0}(t) = 1 - e^{-(\lambda + \mu)t}$$

$$H_{1D}(t) = \begin{cases} 1 - (1 - \frac{t}{T})e^{-\lambda t}, & t < T, \\ 1, & t \geq T, \end{cases}$$

$$H_{0,0}(t) = 1 - e^{-\mu t}.$$

In order to compute the transition probability from state $1D$ to state $(1, 0)$, we proceed as follows. Let $X \sim \text{EXP}(\lambda)$ and $Y \sim U(0, T)$ random variables. We are interested in $P(X > Y)$. Recalling the technique in Example 5.4, we have

$$P(X > Y) = \int_0^T P(X > t) f_Y(t) dt$$

$$= \int_0^T e^{-\lambda t} \frac{1}{T} dt = \frac{1}{\lambda T} (1 - e^{-\lambda T}).$$

This expression can also be obtained from equation (5.30). The one-step transition probability matrix P of the DTMC embedded at the time of transitions

(see Section 7.7) is given by

$$P = \begin{bmatrix} 1,1 & 1C & 1,0 & 1D & 0,0 \\ 1,1 & 0 & \frac{\lambda(1-c)}{\lambda+\lambda_s} & \frac{\lambda c + \lambda_s c_s}{\lambda+\lambda_s} & \frac{\lambda_s(1-c_s)}{\lambda+\lambda_s} & 0 \\ 1C & 0 & 0 & \frac{\beta}{\beta+\lambda_s} & 0 & \frac{\lambda_s}{\beta+\lambda_s} \\ 1,0 & \frac{\mu}{\lambda+\mu} & 0 & 0 & 0 & \frac{\lambda}{\lambda+\mu} \\ 1D & 0 & 0 & \frac{1}{\lambda T}(1 - e^{-\lambda T}) & 0 & 1 - \frac{1}{\lambda T}(1 - e^{-\lambda T}) \\ 0,0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

We denote by the vector \mathbf{v} , the state probabilities of the embedded DTMC. In our example, since we have five states, we have $\mathbf{v} = [v_{1,1}, v_{1C}, v_{1,0}, v_{1D}, v_{0,0}]$. To obtain the steady-state probabilities, solve the equation [the same as equation (7.18)],

$$\mathbf{v} = \mathbf{v}P.$$

This yields

$$\mathbf{v} = \left[v_{1,1}, \frac{\lambda(1-c)}{\lambda+\lambda_s} v_{1,1}, \frac{\lambda+\mu}{\mu} v_{1,1}, \frac{\lambda_s(1-c_s)}{\lambda+\lambda_s} v_{1,1}, \left(\frac{\lambda(1-c)\lambda_s}{(\lambda+\lambda_s)(\beta+\lambda_s)} + \frac{\lambda}{\mu} + \frac{\lambda_s(1-c_s)}{\lambda+\lambda_s} \left(1 - \frac{1}{\lambda T}(1 - e^{-\lambda T}) \right) \right) v_{1,1} \right],$$

where

$$v_{1,1} = \left[1 + \frac{\lambda(1-c)}{\lambda+\lambda_s} + \frac{\lambda+\mu}{\mu} + \frac{\lambda_s(1-c_s)}{\lambda+\lambda_s} + \frac{\lambda(1-c)\lambda_s}{(\lambda+\lambda_s)(\beta+\lambda_s)} + \frac{\lambda}{\mu} + \frac{\lambda_s(1-c_s)}{\lambda+\lambda_s} \left(1 - \frac{1}{\lambda T}(1 - e^{-\lambda T}) \right) \right]^{-1}.$$

The mean sojourn time h_i in state i is given by

$$h_i = \int_0^\infty [1 - H_i(t)] dt.$$

Hence we have

$$\begin{aligned} h_{1,1} &= \frac{1}{\lambda+\lambda_s} \\ h_{1C} &= \frac{1}{\beta+\lambda_s} \\ h_{1,0} &= \frac{1}{\lambda+\mu} \\ h_{1D} &= \frac{1}{\lambda} - \frac{1}{T\lambda^2}(1 - e^{-\lambda T}) \\ h_{0,0} &= \frac{1}{\mu}. \end{aligned}$$

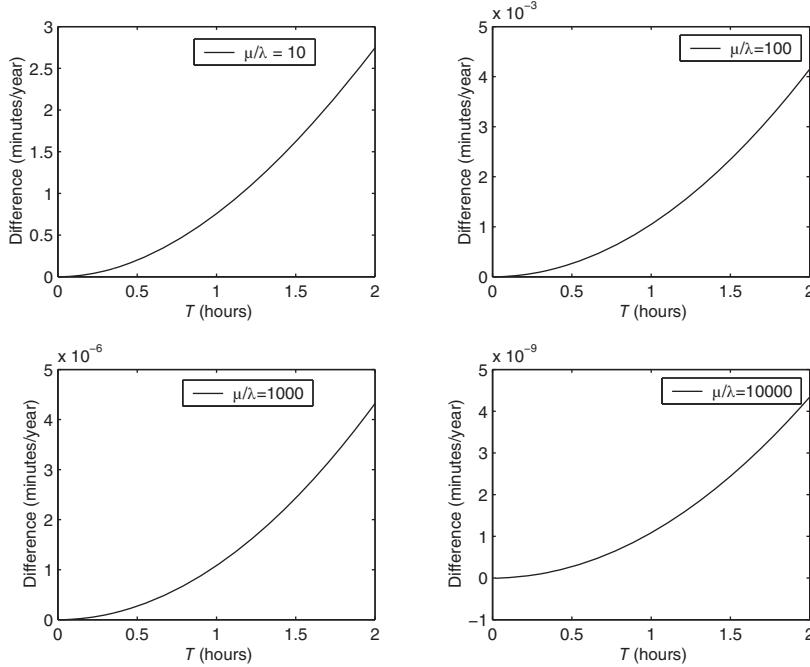


Figure 8.39. Difference in downtime of the SMP model and the approximate CTMC model

The state probabilities of the semi-Markov chain are then given by

$$\pi_i = \frac{v_i h_i}{\sum_j v_j h_j},$$

where $i, j \in \{(1, 1), 1C, (1, 0), 1D, (0, 0)\}$ [CINL 1975]. In all the methods above, the steady-state unavailability for the system is obtained as $\pi_{1C} + \pi_{0,0}$.

Figure 8.39 plots the difference between downtime estimates obtained using the SMP model above and that obtained by approximating the $U(0, T)$ distribution by an exponential distribution with mean $T/2$. For the illustration in Figure 8.39, we take $c = 0.9$, $c_s = 0.9$, $\mu = 1$ per hour, $\beta = 12$ per hour, and $\lambda_s = \lambda/4$. We see that the higher the μ/λ ratio, the lower the difference in the downtime computed by the two models.

#

More detailed studies of availability models are available in the literature [FRIC 1999, HEIM 1990, IBE 1989b, MUPP 1992a, MUPP 1996].

Example 8.28 (Hierarchical Modeling)

Consider the availability model of a workstation consisting of three subsystems: a cooling subsystem with two fans, a dual power supply subsystem, and a two-CPU

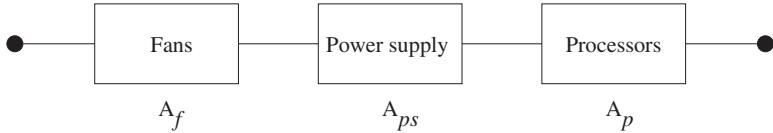


Figure 8.40. Top-level model for Example 8.28

processing subsystem. The workstation is considered to be unavailable when one or more of the subsystems have failed. It is possible to construct a composite CTMC model for the entire workstation, but if the failures and repairs occurring in the three subsystems are independent of each other, then a *hierarchical* model can be constructed. A hierarchical model consists of multiple levels of models, where the lower-level models capture the detailed behavior of subsystems and the topmost level is the system-level model. Hierarchical models scale better with the number of subsystems and subsystem components than does a composite model. For our example, the top-level model consists of the series reliability block diagram shown in Figure 8.40.

The availability of the workstation is then given by

$$A = A_f \cdot A_{ps} \cdot A_p.$$

The availabilities of the cooling, power supply, and processor subsystems, that is, A_f , A_{ps} , and A_p , respectively, can be obtained by solving detailed lower-level models. For instance, if the two fans form a parallel redundant system, the availability of the cooling subsystem, A_f can be computed using the model in Figure 8.27. Adding the subscript f to the rates, we obtain

$$\begin{aligned} A_f &= \pi_2 + \pi_1 \\ &= 1 - \pi_0 \\ &= 1 - \frac{1}{1 + \frac{\mu_f}{\lambda_f} + \frac{\mu_f^2}{2\lambda_f^2}}. \end{aligned}$$

Let us consider that when one of the power supplies fails, the other working supply can be automatically switched in. With probability c_{ps} this switching is successful, and with probability $1 - c_{ps}$ the switching fails, incurring a longer reconfiguration delay. The availability of the power supply subsystem is obtained by solving the model in Figure 8.30. Adding the subscript ps to the rates and the coverage factor, we obtain

$$\begin{aligned} A_{ps} &= \pi_2 + \pi_1 \\ &= \frac{\frac{\mu_{ps}}{2\lambda_{ps}} + 1}{\frac{\lambda_{ps}}{\mu_{ps}} + 1 + \frac{\mu_{ps}(1 - c_{ps})}{\beta_{ps}} + \frac{\mu_{ps}}{2\lambda_{ps}}}. \end{aligned}$$

Consider that the processors have a detection delay and imperfect coverage when one of them fails, as in the two-component system in Example 8.23. The availability of the processing subsystem is then given by the solution to the model in Figure 8.32. Adding the subscript p to the rates and the coverage factor yields

$$\begin{aligned} A_p &= \pi_2 + \pi_1 \\ &= \frac{\frac{\mu_p}{2\lambda_p} + 1}{\frac{\mu_p}{2\lambda_p} + \frac{\lambda_p + \mu_p}{\delta_p + \lambda_p} + 1 + \frac{\lambda_p}{\mu_p} + \frac{\delta_p(1 - c_p)\mu_p}{\beta_p(\delta_p + \lambda_p)}}. \end{aligned}$$

For further examples of hierarchical availability models with independent subsystems see the paper by Ibe *et al.* [IBE 1989b], and for nearly independent subsystems, see the paper by Tomek and Trivedi [TOME 1991].

‡

Problems

1. Consider a variation of the two-state availability model (Example 8.6) so that the time to failure is a k -stage hypoexponentially distributed random variable with parameters $\lambda_1, \lambda_2, \dots, \lambda_k$ and the repair times are exponentially distributed with parameter μ . Compute the steady-state availability. Recall that the time to failure of a hybrid k -out-of- n system (which includes the class of parallel redundant, standby redundant, and TMR systems) is hypoexponentially distributed. The model of this example thus gives the steady-state availability for this class of systems, provided that the repair process cannot begin until the system breaks down. Show that the availability of such a system is obtained from the two-state model by substituting for λ , from the equation

$$\frac{1}{\lambda} = \sum_{i=1}^k \frac{1}{\lambda_i}.$$

2. Consider another variation of the two-state availability model where the time to failure of the unit is exponentially distributed with parameter λ , while the repair times are hyperexponentially distributed with phase selection probabilities $\alpha_1, \alpha_2, \dots, \alpha_k$, and individual phase durations have exponential distributions with parameters $\mu_1, \mu_2, \dots, \mu_k$, respectively. Obtain an expression for the steady-state availability.
3. Show that the equivalent repair rate μ_{eq} for the availability model in Example 8.19 is given by $1/\mu_{\text{eq}} = 1/\mu_1 + 1/\mu_2$. Recall the method of computing μ_{eq} from Example 8.11.
4. In the two-component availability model of Figure 8.27, we assumed that a unit is available for repair as soon as it breaks down. However, in many systems it is not possible to service a failed unit until the complete system fails. This can happen if only the system's output is monitored rather than the status of individual units. Consider a two-unit parallel redundant configuration in which repairs may

not begin until both units break down. Assume a constant failure rate λ for each unit and an exponentially distributed repair time with mean $1/\mu$. Show that the steady-state availability is given by

$$\frac{3\mu^2 + 2\lambda\mu}{3\mu^2 + 3\lambda\mu + \lambda^2}.$$

Compare the downtime of this maintenance policy with that of the maintenance policy that allows repairs as soon as the failure occurs. Use $\lambda = 0.0001$ per hour, $\mu = 1.0$ per hour, and compute the expected downtime over a period of 8760 h for the two cases. In both cases, assume that each unit has its own repair facility.

5. For the parallel redundant system with non-zero detection delay (Example 8.21), compute and plot the percentage of downtime contributed by state $1D$ as function of $1/\delta$ for several different values of $t_{th} \geq 0$. Also separate out two components of the equivalent failure rate λ_{eq} in the same way.
6. For the WFS example, plot the steady-state, instantaneous, and interval availabilities as functions of time.
7. For the preventive maintenance problem (Example 8.20), derive an expression for the optimal inspection rate λ_{in}^* so as to maximize the steady-state availability.
8. Consider an approximation for the CTMC model of Example 8.21 where the delay state is replaced by an instantaneous branch point. Using the technique of Example 5.4 or using equation (5.30). Show that the coverage probability is $\delta/(\delta + \lambda)$. Solve the reduced model and obtain an expression for the error of approximation in steady-state availability.
9. Modify and solve the CTMC of Example 8.23 so as to introduce transient faults that are treated by a retry. Resulting CTMC will have an arc leading from state $1D$ to state 2 at the rate δr where r is the probability of transient restoration. Clearly, the arc from state $1D$ to $1C$ will now be relabeled as $\delta(1 - r - c)$.

8.4.2 Performance Models

In this subsection we will discuss several non-birth-death performance models. Performance measures that we consider include the mean number in the system, the mean response time, and loss probability, among others. Examples that we consider include a heterogeneous $M/M/2$ queue and a single-server queue with a non-Poisson arrival stream, specifically, a Markov modulated Poisson process.

Example 8.29 ($M/M/2$ Queue with Heterogeneous Servers) [BHAT 1984]

We consider a variant of the $M/M/2$ queue where the service rates of the two processors are not identical. This would be the case, for example, in a heterogeneous multiprocessor system. The queuing structure is shown in Figure 8.41. Assume without loss of generality that $\mu_1 > \mu_2$.

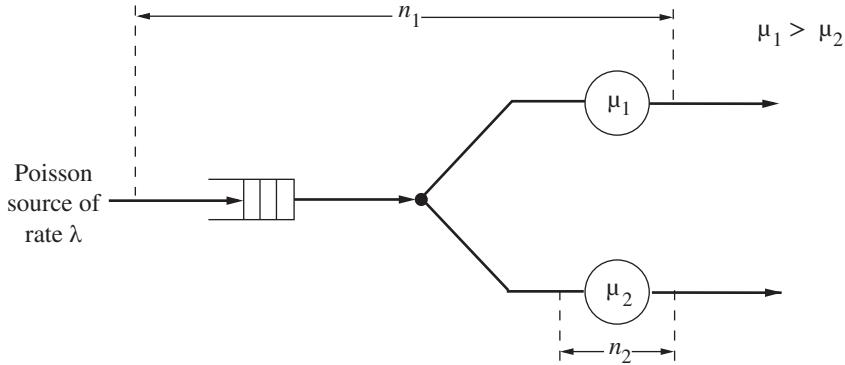


Figure 8.41. $M/M/2$ heterogeneous system

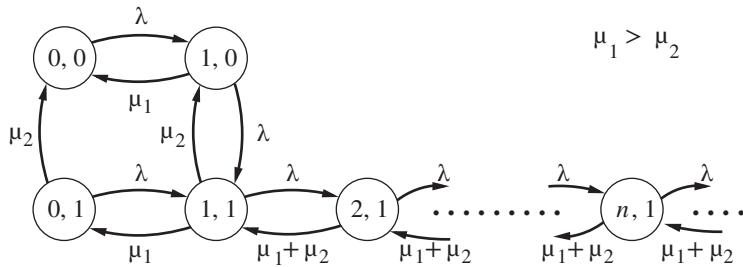


Figure 8.42. The state diagram for the $M/M/2$ heterogeneous queue

The state of the system is defined to be the tuple (n_1, n_2) where $n_1 \geq 0$ denotes the number of jobs in the queue including any at the faster server, and $n_2 \in \{0, 1\}$ denotes the number of jobs at the slower server. Jobs wait in line in the order of their arrival. When both servers are idle, the faster server is scheduled for service before the slower one. The state diagram of the system is given in Figure 8.42. Balance equations, in the steady state, can be written by equating the rate of flow into a state to the rate of flow out of that state:

$$\lambda\pi(0, 0) = \mu_1\pi(1, 0) + \mu_2\pi(0, 1), \quad (8.91)$$

$$(\lambda + \mu_1)\pi(1, 0) = \mu_2\pi(1, 1) + \lambda\pi(0, 0), \quad (8.92)$$

$$(\lambda + \mu_2)\pi(0, 1) = \mu_1\pi(1, 1), \quad (8.93)$$

$$(\lambda + \mu_1 + \mu_2)\pi(1, 1) = (\mu_1 + \mu_2)\pi(2, 1) + \lambda\pi(0, 1) + \lambda\pi(1, 0), \quad (8.94)$$

$$(\lambda + \mu_1 + \mu_2)\pi(n, 1) = (\mu_1 + \mu_2)\pi(n+1, 1) + \lambda\pi(n-1, 1), \quad n > 1. \quad (8.95)$$

The traffic intensity for this system is

$$\rho = \frac{\lambda}{\mu_1 + \mu_2}.$$

The form of equation (8.95) is similar to the balance equation of a birth-death process (equation (8.29)). Therefore

$$\pi(n, 1) = \frac{\lambda}{\mu_1 + \mu_2} \pi(n - 1, 1), \quad n > 1 \quad (8.96)$$

can easily be seen to satisfy equation (8.95). By repeated use of equation (8.96), we have

$$\pi(n, 1) = \rho \pi(n - 1, 1) = \rho^{n-1} \pi(1, 1), \quad n > 1. \quad (8.97)$$

From equations (8.92)–(8.93) we can obtain the following by elimination:

$$\begin{aligned} \pi(0, 1) &= \frac{\rho}{1 + 2\rho} \frac{\lambda}{\mu_2} \pi(0, 0), \\ \pi(1, 0) &= \frac{1 + \rho}{1 + 2\rho} \frac{\lambda}{\mu_1} \pi(0, 0), \\ \pi(1, 1) &= \frac{\rho}{1 + 2\rho} \frac{\lambda(\lambda + \mu_2)}{\mu_1 \mu_2} \pi(0, 0). \end{aligned}$$

Now, observing that

$$\left[\sum_{n \geq 1} \pi(n, 1) \right] + \pi(0, 1) + \pi(1, 0) + \pi(0, 0) = 1,$$

we have

$$\left(\sum_{n \geq 1} \rho^{n-1} \right) \pi(1, 1) + \pi(0, 0) \left[\frac{\rho}{1 + 2\rho} \frac{\lambda}{\mu_2} + \frac{1 + \rho}{1 + 2\rho} \frac{\lambda}{\mu_1} + 1 \right] = 1,$$

or

$$\frac{1}{1 - \rho} \frac{\rho}{1 + 2\rho} \frac{\lambda(\lambda + \mu_2)}{\mu_1 \mu_2} \pi(0, 0) + \pi(0, 0) \left[\frac{\rho}{1 + 2\rho} \frac{\lambda}{\mu_2} + \frac{1 + \rho}{1 + 2\rho} \frac{\lambda}{\mu_1} + 1 \right] = 1,$$

from which we get

$$\pi(0, 0) = \left[1 + \frac{\lambda(\lambda + \mu_2)}{\mu_1 \mu_2 (1 + 2\rho)(1 - \rho)} \right]^{-1}. \quad (8.98)$$

The average number of jobs in the system may now be computed by assigning the reward rate $r_{n_1, n_2} = n_1 + n_2$ equal to the number of customers in the system in state (n_1, n_2) . Therefore, the average number of jobs is given by

$$\begin{aligned} E[N] &= \sum_{k \geq 0} k\pi(k, 0) + \sum_{k \geq 0} (k+1)\pi(k, 1) \\ &= \pi(1, 0) + \pi(0, 1) + \sum_{k \geq 1} (k+1)\pi(k, 1) \\ &= \pi(1, 0) + \pi(0, 1) + \sum_{k \geq 1} \pi(k, 1) + \sum_{k \geq 1} k\pi(k, 1) \\ &= 1 - \pi(0, 0) + \pi(1, 1) \sum_{k=1}^{\infty} k\rho^{k-1} \\ &= 1 - \pi(0, 0) + \frac{\pi(1, 1)}{(1-\rho)^2}, \end{aligned}$$

so

$$E[N] = \frac{1}{F(1-\rho)^2}, \quad (8.99)$$

where

$$F = \left[\frac{\mu_1 \mu_2 (1+2\rho)}{\lambda(\lambda+\mu_2)} + \frac{1}{1-\rho} \right].$$

‡

Example 8.30 [FULL 1975]

A computing center initially had an IBM 360/50 computer system. The job stream could be modeled as a Poisson process with rate λ jobs/minute, and the service times were exponentially distributed with an average service rate μ_2 jobs/minute. Thus, $\rho_2 = \lambda/\mu_2$, and the average response time is given by equation (8.38) as $E[R_2] = (1/\mu_2)/(1 - \rho_2)$.

Suppose this response time is considered intolerable by the users and an IBM 370/155 is purchased and added to the system. Let the service rate μ_1 of the 370/155 be equal to $\alpha\mu_2$ for some $\alpha > 1$. Assuming that a common-queue heterogeneous $M/M/2$ structure is used, we can compute the average response time $E[R]$ as follows: Let

$$\begin{aligned} \rho &= \frac{\lambda}{\mu_1 + \mu_2} = \frac{\rho_2}{1 + \alpha}, \\ F &= \frac{\alpha\mu_2^2(1+2\rho)}{\lambda(\lambda+\mu_2)} + \frac{1}{1-\rho} \\ &= \frac{\alpha(1+2\rho)}{\rho_2(1+\rho_2)} + \frac{1}{1-\rho}. \end{aligned}$$

TABLE 8.5. Average response times

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
$E[R_1]$	20	3.33	1.818	1.25	0.95
$E[R_2]$	20	20	20	20	20
$E[R]$	4.7619	2.6616	1.875	1.459	1.20

Using equation (8.99), the average number of customers in the system is

$$\begin{aligned} E[N] &= \frac{1}{F(1-\rho)^2} \\ &= \frac{\rho(1+\alpha)(1+\rho+\rho\alpha)}{(1-\rho)(\alpha+\rho+\rho^2+2\alpha\rho+\rho^2\alpha^2)}. \end{aligned}$$

The average response time is then computed using Little's formula as

$$E[R] = \frac{E[N]}{\lambda} = \frac{1+\rho+\rho\alpha}{(1-\rho)\mu_2(\alpha+\rho+\rho^2+2\alpha\rho+\rho^2\alpha^2)}. \quad (8.100)$$

For any $\alpha \geq 1$, $E[R] < E[R_2]$.

Suppose that we want to consider the possibility of disconnecting the 360/50 and using the 370/155 by itself, thus reducing the response time. In this case we have $\rho_1 = \lambda/(\alpha\mu_2) = \rho_2/\alpha$, and the average response time is given by equation (8.38) as

$$E[R_1] = \frac{1/\mu_1}{1 - \rho_1} = \frac{1/\mu_2}{\alpha - \rho_2}.$$

The condition under which $E[R_1] \leq E[R]$ can be simplified to

$$\rho^2(1+\alpha^2) - \rho(1+2\alpha^2) + (\alpha^2 - \alpha - 2) \geq 0,$$

or, in terms of ρ_2 and α :

$$\frac{\rho_2^2(1+\alpha^2)}{(1+\alpha)^2} - \rho_2 \frac{1+2\alpha^2}{1+\alpha} + \alpha^2 - \alpha - 2 \geq 0.$$

Thus, for example, if $\lambda = 0.2$ and $\mu_2 = 0.25$ so that $\rho_2 = 0.8$, then if the 370/155 is more than 3 times faster than the 360/50, the inequality shown above is satisfied, and, surprisingly, it is better to disconnect the slower machine altogether. Of course, this conclusion holds only if we want to minimize response time. If we are interested in processing a larger throughput (λ), particularly if $\lambda \geq \alpha\mu_2$, then we are forced to use both machines.

Table 8.5 gives the average response times (in minutes) of the three configurations for different values of α with $\lambda = 0.2$ and $\mu_2 = 0.25$.

8.4.2.1 Markov Modulated Poisson Process (MMPP). Markov modulated Poisson process (MMPP) has been extensively used for telecommunication traffic modeling [FISC 1993, WANG 1995, YOUS 1996, KANG 1997, CHOI 1998]. The main reason for its popularity in traffic modeling is that MMPP has the capability of capturing some of the most important correlations between interarrival times and still remains analytically tractable. MMPP is a special case of the Markovian arrival process (MAP) introduced by D. Lucantoni *et al.* [LUCA 1990].

An MMPP is a doubly stochastic Poisson process whose arrival rate is “modulated” by an irreducible continuous time Markov chain. Let $Q = [q_{ij}]_{m \times m}$ be the generator matrix of the CTMC with m states. Each state i is assigned a Poisson arrival rate, λ_i , $i = 1, 2, \dots, m$. The Poisson arrival rate is determined by the state of the CTMC; thus, when the Markov chain is in state i , arrivals occur according to a Poisson process of rate λ_i . To facilitate our discussion, we use λ to denote the arrival rate vector $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]^T$. We may also use the diagonal matrix of arrival rates, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$. Clearly, $\Lambda e = \lambda$, where $e = [1, 1, \dots, 1]^T$.

8.4.2.2 The Counting Process. We are interested in the associated counting process of MMPP. Let $N(t)$ be the number of arrivals in $(0, t]$ and $J(t)$ the state of the modulating CTMC. The bivariate process $\{J(t), N(t), t \geq 0\}$ is the counting process whose state space is $\{1, 2, \dots, m\} \times \{0, 1, \dots\}$. Transitions between states with the same number of arrivals—say, (i, n) and (j, n) —are the same as transitions between state i and j of the CTMC with rate q_{ij} and q_{ji} , respectively. An arrival may occur in any of the modulating CTMC states, resulting in the counter increasing by one. Thus, we also have transition from state (i, n) to state $(i, n + 1)$ with rate λ_i . The state diagram of the bivariate process for a three-state MMPP is illustrated in Figure 8.43. Clearly, the counting process is also a homogeneous CTMC. Let π be the steady-state vector of the MMPP, which is the solution to

$$\pi Q = 0, \quad \pi e = 1. \quad (8.101)$$

It can be shown that [NEUT 1978]

$$\lim_{t \rightarrow \infty} E[N(t)]/t = \pi \lambda. \quad (8.102)$$

This result is indeed expected; the steady-state expected number of arrivals in an interval of length t is the product of time duration t and the average arrival rate, $\pi \lambda$. $\pi \lambda$ is the sum of rates weighted by steady-state probabilities of the modulating CTMC.

8.4.2.3 The MMPP/M/1 Queue. We now consider an MMPP/M/1 queue in which the arrival process is an MMPP characterized by the generator

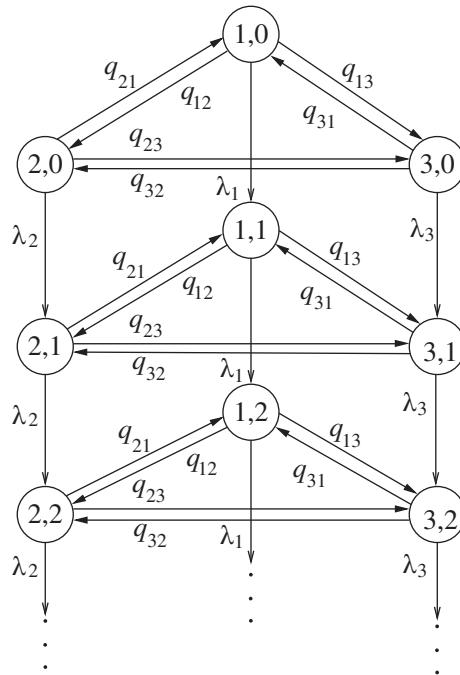


Figure 8.43. The counting process of MMPP with three states

matrix Q of the modulating CTMC and the arrival rate vector λ . The service time is exponentially distributed with mean $1/\mu$. The buffer size of the queue is assumed to be infinite. The state of the system can be described by the number of arrivals (customers) in the queuing system at time t , $N(t)$, and the state of the modulating CTMC at time t , $J(t)$. The underlying process $\{J(t), N(t), t \geq 0\}$ has the same state space as the MMPP counting process: $\{1, 2, \dots, m\} \times \{0, 1, \dots\}$. The state diagram for an MMPP/M/1 queue is illustrated in Figure 8.44, in which the MMPP arrival process has three states. As we can see, the diagram is similar to that of the MMPP counting process. The only difference is that the MMPP/M/1 has transitions from $(i, n+1)$ to (i, n) for $i = 1, 2, \dots, m$, due to the departure of a customer after service. The process $\{J(t), N(t)\}$ is again a homogeneous CTMC.

The structure of the Markov chain suggests that its steady-state solution may have the same form as that of an $M/M/1$ queue. Let $\pi_{i,n}$ be the steady-state probability that the MMPP modulating process is in state i and the system has n jobs. Solution to this and related queues based on matrix geometric methods can be found in [FISC 1993, WANG 1995, YOUS 1996, KANG 1997, CHOI 1998] and related papers.

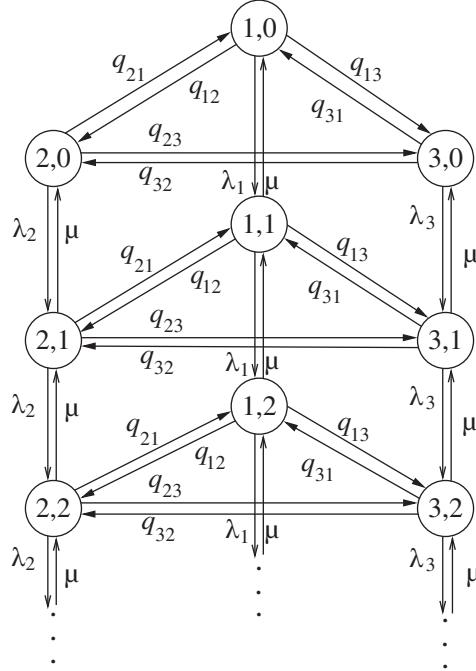


Figure 8.44. State diagram for the $MMPP/M/1$ queue

Problems

- * Consider the following concurrent program [TOWS 1978] with a cyclic structure:

```

repeat
  TCPU1;
  if B then TIO1,
  else
    cobegin
      TCPU2; TIO2
    coend
  forever.

```

Assume that successive tests on condition B form a sequence of Bernoulli trials with probability of failure q . The execution times of the statement groups (or tasks) $TCPU1$ and $TCPU2$ are $EXP(\mu_1)$ and $EXP(\mu_2)$ random variables, respectively, while the execution times of $TIO1$ and $TIO2$ are both $EXP(\lambda)$ random variables. Draw the CTMC state diagram of this system and solve for

the steady-state probabilities. Assuming that TCPU1 and TCPU2 are executed on a single CPU and that TIO1 and TIO2 are executed on a single I/O processor, compute steady-state utilizations of the two processors. Use $1/\mu_1 = 8$ ms, $1/\mu_2 = 26.6$ ms, $1/\lambda = 46.1$ ms, and vary q from 0 to 1.

- Suppose we want to purchase a two-processor system, to be operated as an $M/M/2$ queue. Keeping the total amount of computing power constant, we want to investigate the tradeoffs between a system with homogeneous processors (total power 2μ) and a heterogeneous system (total power $\mu_1 + \mu_2 = 2\mu$). Using the average response time as the criterion function, which system will be superior? To take a concrete case, let $\rho = \lambda/2\mu$ vary from 0.1 to 0.9 and compute the average response times of the two systems, choosing different sets of values of μ_1 and μ_2 and assuming $\lambda = 1$. [It can be shown that the optimum values of μ_1 and μ_2 are given by

$$\lambda \left(\frac{1}{\rho} + 1 - \sqrt{1 + \frac{1}{\rho}} \right) \text{ and } \lambda \left(\sqrt{1 + \frac{1}{\rho}} - 1 \right) \text{ where } \rho = \frac{\lambda}{2\mu}.$$

- Write a discrete-event simulation program to simulate the $M/M/2$ heterogeneous queuing system with $\mu_2 = 0.25$, $\lambda = 0.2$, and α varied as in Table 8.5. In order to estimate the steady-state response times $E[R_1]$, $E[R_2]$, and $E[R]$ as defined in Table 8.5, you have to execute three different simulations (two for an $M/M/1$ queue and one for the $M/M/2$ case), discard the statistics corresponding to initial transients, and then collect the steady-state values. The attainment of the steady state is determined by experimentation. For statistical analysis of outputs, see Chapter 10 of this book.
- First obtain an expression for $E[N^2]$ and then for the perceived mean queue length $E[N^2]/E[N]$. Now solve the optimization in problem 2 above with the objective of minimizing the perceived mean queue length [GEIS 1983].
- For the example of the $M/M/2$ heterogeneous queue, let α_0 denote that value of α for which $E[R] = E[R_1]$. Study the variation of α_0 as a function of the job arrival rate λ . Graph this relationship, using the equation relating α and ρ_2 developed in Example 8.30.
- For the special case of $\mu_1 = \mu_2$ in Example 8.29, show that equation (8.98) reduces to $M/M/2$ equation, (8.46), and that equation (8.99) reduces to the corresponding $M/M/2$ equation, (8.47).
- * Write down the steady-state balance equations for the $MMPP/M/1$ queue of Figure 8.44, and solve them using the matrix geometric method. Also compute the average number in the system and the average response time in the steady state.

8.4.3 Performance and Availability Combined

In Sections 8.4.1 and 8.4.2, respectively, we have separately discussed availability models and performance models that commonly occur in computer communication systems. The analysis of a communication network from the pure performance viewpoint tends to be optimistic since it ignores the failure-repair

behavior in the system. On the other hand, pure availability analysis tends to be too conservative since performance considerations are not taken into account. Also, in a well-designed communication network, the failure of a communication link or node will cause partial outage of the system, specifically, the decrease of network capacity available to the users, which further affects the performance and quality of service (QoS) to the users. Therefore, in real systems, availability, capacity, and performance are important QoS indices which should be studied in a composite manner. The combined evaluation of the indices described above is useful when the system under study can operate in a gracefully degradable manner in the presence of component failures. In this section we discuss several examples of such combined performance–availability analysis.

Example 8.31 (Erlang Loss Model)

Consider a telephone switching system consisting of n trunks (or channels) with an infinite caller population as discussed in problem 1 at the end of Section 8.2.2. If an arriving call finds all n trunks busy, it does not enter the system and is lost instead. The call arrival process is assumed to be Poissonian with rate λ . We assume that call holding times are independent, exponentially distributed random variables with parameter μ and independent of the call arrival process.

We first present an availability model that accounts for failure–repair behavior of trunks; then, we use a performance model to compute performance indices such as blocking probability given the number of nonfailed trunks; finally, we combine the two models together and give performability measures of interest. Assume that the times to trunk failure and repair are exponentially distributed with mean $1/\gamma$ and $1/\tau$, respectively. Also assume that a single repair facility is shared by all the trunks. The availability model is then a homogeneous CTMC with the state diagram shown in Figure 8.45. Here, the state index denotes the number of nonfailed trunks in the system. The steady-state probability for the number of nonfailed channels in the system is given by

$$\pi_i = \frac{1}{i!} (\tau/\gamma)^i \pi_0, \quad i = 1, 2, \dots, n,$$

where the steady-state system unavailability

$$U = \pi_0 = \left[\sum_{i=0}^n \frac{1}{i!} (\tau/\gamma)^i \right]^{-1}.$$

Consider the performance model with the given number i of nonfailed channels. The principal quantity of interest is the *blocking probability*, that is, the steady-state

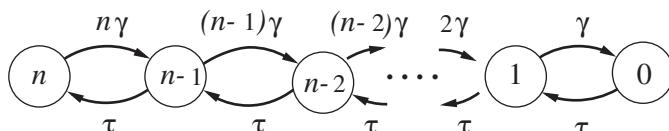


Figure 8.45. State diagram for the Erlang loss availability model

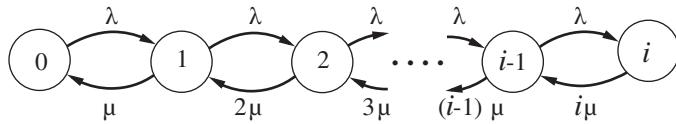


Figure 8.46. State diagram for the Erlang loss performance model

probability that all trunks are busy, in which case the arriving call is refused service. Note that in this performance model, the assumption is that blocked calls are lost (not reattempted). The performance model of this telephony system is an $M/M/i$ loss system, and the state diagram is shown in Figure 8.46. The blocking probability with i channels in the system is given by

$$P_b(i) = \frac{(\lambda/\mu)^i / i!}{\sum_{j=0}^i (\lambda/\mu)^j / j!}.$$

This equation is known as the *Erlang's B loss formula*. It can be shown to hold even if the call holding time follows arbitrary distribution with mean $1/\mu$ [ROSS 1983].

Now, we construct the composite model for the combined performance and availability analysis and the state diagram is shown in Figure 8.47. Here, the state (i, j)

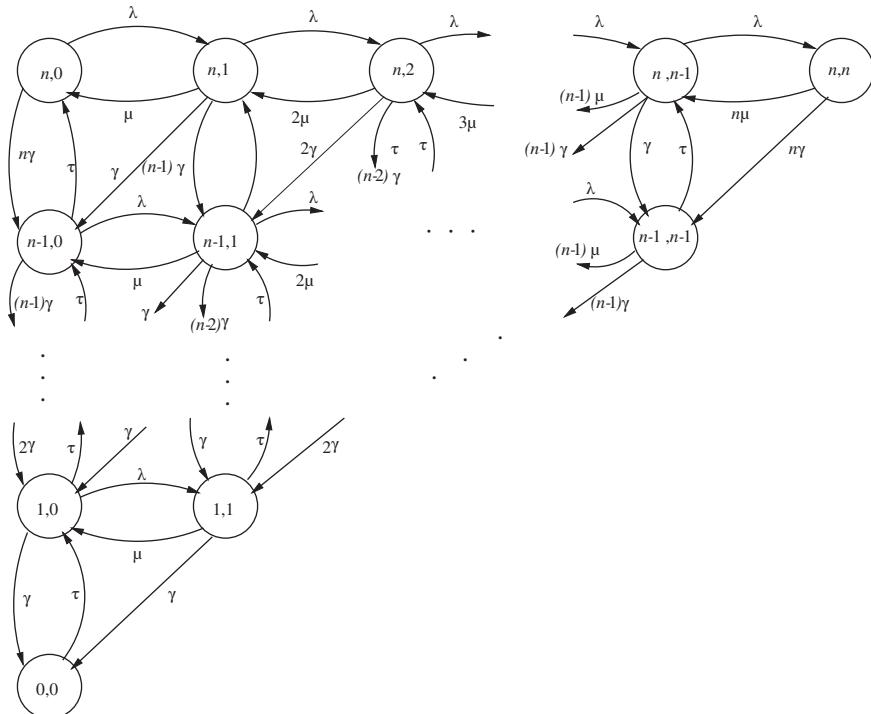


Figure 8.47. State diagram for the Erlang loss composite model

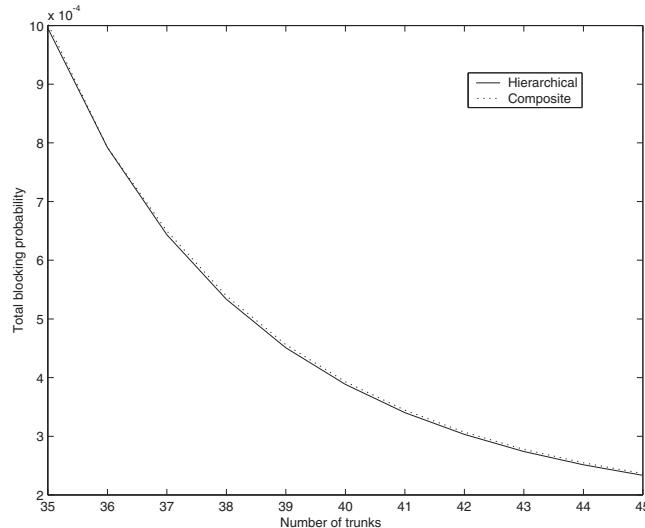


Figure 8.48. Total blocking probability in the Erlang loss model

denotes i nonfailed trunks and $j \leq i$ ongoing calls in the system. Note that trunks that are in use as well as those that are free can fail with the corresponding failure rate. This composite model is a homogeneous irreducible CTMC with $(n+1)(n+2)/2$ states, and the steady-state probability can be obtained (see Theorem 8.1) by solving the linear system of homogeneous equations. Such a solution may be obtained using a software package such as SHARPE [SAHN 1996]. The total call blocking probability is then given by

$$T_b = \sum_{i=0}^n \pi_{i,i}.$$

In a composite model, finding the required measure will be cumbersome and numerically error-prone when the number of trunks is large. To avoid these problems, we can compute the required measure approximately using a hierarchical model. In this approach, a top-level availability model (Figure 8.45) is turned into a Markov reward model (MRM), where the reward rates come from a sequence of performance models (Figure 8.46) and are supplied to the top-level availability model. Attach a reward rate r_i to the state i of the availability model as the blocking probability with i trunks in the system, that is, $r_i = P_b(i)$, $i \geq 1$ and $r_0 = 1$. Then the required total blocking probability can be computed as the expected reward rate in the steady-state and is given by

$$\hat{T}_b = \left[\sum_{i=0}^n r_i \pi_i \right] = \left[\sum_{i=1}^n P_b(i) \pi_i \right] + \pi_0,$$

where π_i is the steady-state probability that i nonfailed trunks are there in the system. In Figure 8.48, we compare the exact total blocking probability T_b with approximate result \hat{T}_b as functions of the number of trunks.



Example 8.32 (Multiprocessor Model) [TRIV 1990]

Consider a multiprocessor system with n processors. Assume that the failure rate of each processor is γ . There is a single repair facility to be shared by all processors, and the repair times are exponentially distributed with mean $1/\tau$. A processor fault is covered with probability c and is not covered with probability $1 - c$. Subsequent to a covered fault, the system comes up in a degraded mode after a brief reconfiguration delay, while after an uncovered fault, a longer reboot action is required. The reconfiguration times and the reboot times are exponentially distributed with parameters δ and β , respectively. We assume that no other event can take place during a reconfiguration or a reboot and a homogeneous CTMC model for this system is shown in Figure 8.49. In state $i \in \{0, 1, \dots, n\}$, i processors are functioning and $n - i$ processors are waiting for repair whereas in state $x_i \in \{x_n, x_{n-1}, \dots, x_2\}$ the system is undergoing a reconfiguration from a state with i operational processors to a state with $i - 1$ operational processors. Similarly, in state $y_i \in \{y_n, y_{n-1}, \dots, y_2\}$ the system is being rebooted from a state with i operational processors to a state with $i - 1$ operational processors. Solving for the steady-state probabilities, we get

$$\begin{aligned}\pi_{n-i} &= \frac{n!}{(n-i)!} \left(\frac{\gamma}{\tau}\right)^i \pi_n, \quad i = 1, 2, \dots, n \\ \pi_{x_{n-i}} &= \frac{n!}{(n-i)!} \frac{\gamma(n-i)c}{\delta} \left(\frac{\gamma}{\tau}\right)^i \pi_n, \quad i = 0, 1, \dots, n-2 \\ \pi_{y_{n-i}} &= \frac{n!}{(n-i)!} \frac{\gamma(n-i)(1-c)}{\beta} \left(\frac{\gamma}{\tau}\right)^i \pi_n, \quad i = 0, 1, \dots, n-2,\end{aligned}$$

where

$$\pi_n = \left[\sum_{i=0}^n \left(\frac{\gamma}{\tau}\right)^i \frac{n!}{(n-i)!} + \sum_{i=0}^{n-2} \left(\frac{\gamma}{\tau}\right)^i \frac{\gamma(n-i)c n!}{\delta(n-i)!} + \sum_{i=0}^{n-2} \left(\frac{\gamma}{\tau}\right)^i \frac{\gamma(n-i)(1-c)n!}{\beta(n-i)!} \right]^{-1}.$$

Let the steady-state availability, $A(n)$, be defined as a function of n . Then

$$A(n) = \sum_{i=0}^{n-1} \pi_{n-i} = \frac{\sum_{i=0}^{n-1} \theta^i / (n-i)!}{Q_1},$$

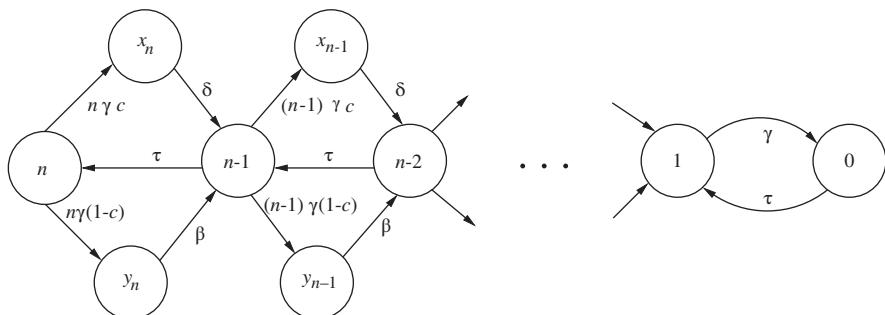


Figure 8.49. State diagram for the multiprocessor availability model

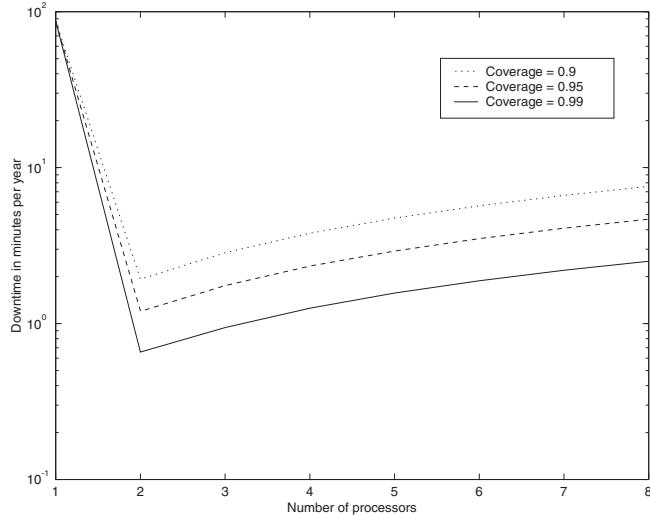


Figure 8.50. Downtime Vs. the number of processors

where $\theta = \gamma/\tau$ and

$$Q_1 = \sum_{i=0}^n \frac{\theta^i}{(n-i)!} + \sum_{i=0}^{n-2} \frac{\gamma(n-i)\theta^i}{(n-i)!} \left\{ \frac{c}{\delta} + \frac{(1-c)}{\beta} \right\}.$$

From these equations, system unavailability $U(n)$ is given by $U(n) = 1 - A(n)$; and downtime during an observation interval of duration T is given by $D(n) = U(n) \times T$.

In Figure 8.50, we have plotted the downtime in minutes per year as a function of the number of processors for different values of the coverage factor (c). Here the downtime $D(n) = U(n) \times 8760 \times 60$ min per year. We use $\gamma = \frac{1}{6000}$ per hour, $\beta = 12$ per hour, and $\tau = 1$ per hour. We use the mean reconfiguration delay, $1/\delta = 10$ s, unless otherwise specified. We see that the downtime is not monotonically decreasing with the number of processors. The reason for this behavior is that as the number of processors increases beyond 2, the primary cause of downtime is reconfigurations and the number of reconfigurations is nearly linearly increasing with the number of processors.

As in the previous example, we present another Markov reward model where the top-level model is the availability model of Figure 8.49. The lower-level model, which captures the performance of the system within a given availability model, will again be a CTMC model. The resulting Markov reward model can then be analyzed for various combined measures of performance and availability.

Turning to the lower-level performance model (from which we will get reward rates), we assume that jobs arriving to the system form a Poisson process with rate λ and that the service requirements of jobs are independent, identically distributed according to an exponential distribution with mean $1/\mu$. Assume also that there are a limited number b of buffers available for the jobs. Arriving tasks are rejected when all the buffers are full. We will also assume that no faults occur during the execution of a task. When i processors are functioning properly, we can use an $M/M/i/b$ queuing

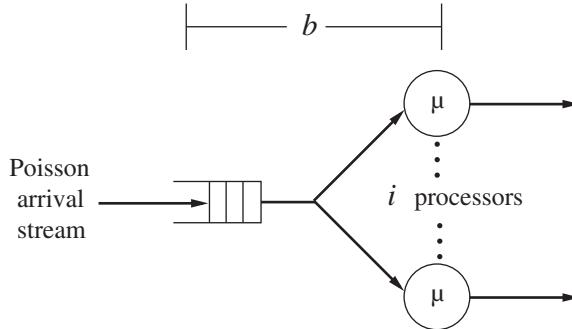


Figure 8.51. Queuing system for the multiprocessor performance model

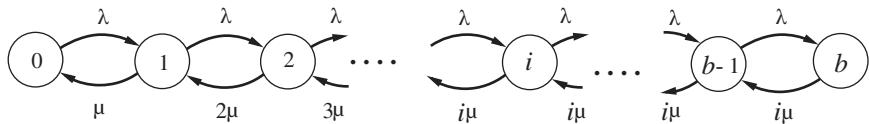


Figure 8.52. State diagram for the multiprocessor performance model

model to describe the performance of the multiprocessor system and the queuing system is shown in Figure 8.51. The state diagram for this performance model is shown in Figure 8.52. The state indices indicate the number of jobs in the system and $\rho = \lambda/\mu$.

The simplest reward assignment for the availability model of Figure 8.49 is to let the reward rate for a system up state i be $r_i = i/n$ and let the reward rate of any down state be 0. Then the expected reward rate in the steady state is specialized to the capacity-oriented availability COA(n) and is given by

$$\text{COA}(n) = \sum_{i=0}^{n-1} \left(\frac{n-i}{n} \right) \pi_{n-i}.$$

Next, we set r_i to be the scaled throughput when i processors are functioning for system up state i and assign reward rate zero to all the remaining states. This gives throughput-oriented availability TOA(n).

Now, we study the total loss probability of a task by suitably assigning reward rates in the top-level availability model. Note that, a task will be rejected whenever b tasks are already in the system. Then the probability of rejection of a task in a configuration with i processors functioning is given by the probability that the CTMC of Figure 8.52 is in state b . We denote this probability as $q_b(i)$:

$$q_b(i) = \begin{cases} \frac{\rho^b}{i^{b-i} i! \sum_{j=0}^{i-1} \frac{\rho^j}{j!} + \sum_{j=i}^b \frac{\rho^j}{i!} \frac{1}{i^{j-i}}}, & b \geq i, \\ \frac{\rho^b}{b! \sum_{j=0}^b \frac{\rho^j}{j!}}, & b < i. \end{cases} \quad (8.103)$$

We now set the reward rate $r_i = q_b(i)$ for up states and we set the reward rate to 1 for all the down states. With this reward assignment, the expected steady-state reward rate specializes to the probability of task rejection due to limited buffers or because the system is down:

$$\pi_{LP}(n, b) = \sum_{i=0}^{n-1} q_b(n-i) \pi_{n-i} + \sum_{i=2}^n (\pi_{x_i} + \pi_{y_i}) + \pi_0.$$

We now impose a deadline on the task response time. Thus if a task response takes longer than d time units then we consider that task as late. Using the formula for the response time distribution in an $M/M/i/b$ queue (see problem 2 before the start of Section 8.2.3.1),

$$P(R_b(i) > d) = \begin{cases} \left[\sum_{j=0}^{i-1} q'_j + \sum_{j=i}^{b-1} q'_j \binom{i}{i-1}^{j-i+1} \right] e^{-\mu d} \\ - \sum_{j=i}^{b-1} q'_j \sum_{k=0}^{j-i} \frac{(\mu id)^k}{k!} e^{-\mu id} \left[\binom{i}{i-1}^{j-i-k+1} - 1 \right], & i > 1, \\ \sum_{j=0}^{b-1} q'_j \sum_{k=0}^j \frac{(\mu d)^k}{k!} e^{-\mu d}, & i = 1, \end{cases}$$

where $R_b(i)$ is the response time random variable with i functioning processors. In this equation we have defined $q'_j = q_j / (1 - q_b(i))$, where q_j is the steady-state probability that there are j jobs in the system of Figure 8.52. We now make the following reward rate assignment:

$$r_i = \begin{cases} 1, & \text{for down states,} \\ q_b(i) + [1 - q_b(i)][P(R_b(i) > d)], & \text{for an up state } i. \end{cases}$$

The expected reward rate in the steady state is now the overall probability of a “lost” task (due to system down or system full or too slow) and is given by

$$\begin{aligned} \pi_{TLP}(n, b, d) &= \sum_{i=0}^{n-1} q_b(n-i) \pi_{n-i} + \sum_{i=0}^{n-1} \{ (1 - q_b(n-i)) P(R_b(i) > d) \} \pi_{n-i} \\ &\quad + \sum_{i=0}^{n-2} (\pi_{x_{n-i}} + \pi_{y_{n-i}}) + \pi_0. \end{aligned}$$

$\pi_{TLP}(n, b, d)$ will be called “total loss probability.”

In Figure 8.53, we have plotted the total loss probability of a task $\pi_{TLP}(n, b, d)$ as a function of the number of processors for different values of λ . We have used $b = 10$, $d = 8 \times 10^{-5}$, and $\mu = 100$ per second. We see that as the value of λ increases, the optimal number of processors, n^* , increases as well. For instance, $n^* = 4$ for $\lambda = 20$, $n^* = 5$ for $\lambda = 40$, $n^* = 6$ for $\lambda = 60$. (Note that $n^* = 2$ for $\lambda = 0$ from Figure 8.50.)



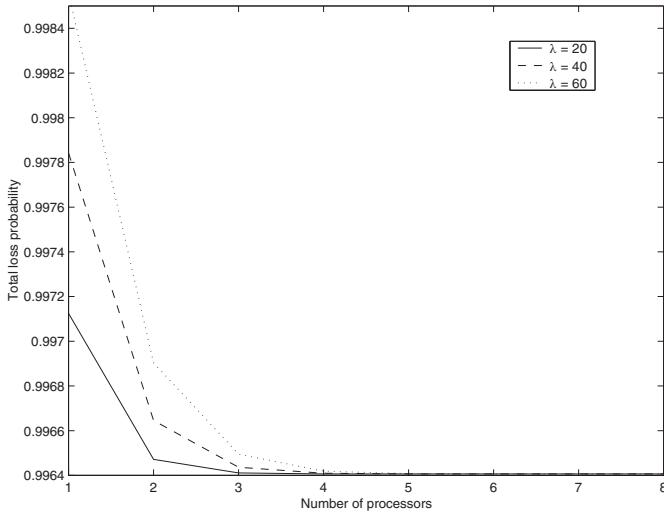


Figure 8.53. Total loss probability for a multiprocessor model

Example 8.33 (Wireless Handoff Model) [CAO 2000]

Consider a wireless cellular system consisting of N_b base repeaters where each base repeater has M channels as discussed in Section 8.2.3.2. Such a system consists of several operational areas, called *cells*. Assume that cells are statistically identical. A cell has multiple base repeaters. Each base repeater provides a number of channels for mobile terminals to communicate with the system. Normally, one of the channels is dedicated to transmitting control messages. Such a channel is called the *control channel*. Failure of the control channel will cause the whole system to fail. To avoid this undesirable situation, an automatic protection switching (APS) scheme has been suggested [SUN 1999] to enable the system to automatically select a channel from the rest of the available channels to substitute for the failed control channel. If all channels are in use (talking), then one of them is forcefully terminated and is used as the control channel. Therefore, a total number of $N_b M$ channels are available when the whole system is working properly. Since one of the channels has to be used as the control channel, the total number of available talking channels is $N_b M - 1$. We also assume that the control channel is selected randomly out of $N_b M$ channels. We use the traditional two-level performability model [HAVE 2001, TRIV 1992]: we first present an availability model that accounts for the failure and repair of base repeaters; then, we use a performance model to compute performance indices (new-call *blocking probability* and handoff-call *dropping probability*) given the number of nonfailed base repeaters; finally, we combine them together and give performability measures of interest.

All failure events are assumed to be mutually independent. Times to platform failures and repair are assumed to be exponentially distributed with mean $1/\lambda_s$ and $1/\mu_s$, respectively. Also assumed is that times to base repeater failure and repair are exponentially distributed with mean $1/\lambda_b$ and $1/\mu_b$, respectively, and that a single repair facility is shared by all the base repeaters.

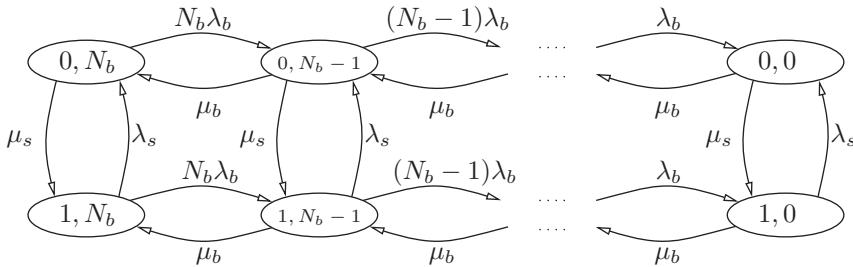


Figure 8.54. Markov chain of availability model

Let $s \in S = \{0, 1\}$ denote a binary value indicating whether the system is down as a result of a platform failure (0 —system down due to a platform failure; 1 —no platform failure has occurred). Also let $k \in B = \{0, 1, \dots, N_b\}$ denote the number of nonfailed base repeaters. The tuple $\{(s, k), s \in S, k \in B\}$ defines a state in which the system is undergoing a (no) platform failure if $s = 0$ (if $s = 1$) and k base repeaters are up. The underlying stochastic process is a homogeneous CTMC with state space $S \times B$. Let $\pi_{s,k}(N_b)$ be the corresponding steady-state probability. The state diagram of this irreducible CTMC is depicted in Figure 8.54.

Solving the above mentioned Markov chain, we have

$$\pi_{s,k}(N_b) = \begin{cases} \frac{\lambda_s}{\lambda_s + \mu_s} \frac{1}{k!} \left(\frac{\mu_b}{\lambda_b} \right)^k \left[1 + \sum_{j=1}^{N_b} \frac{1}{j!} \left(\frac{\mu_b}{\lambda_b} \right)^j \right]^{-1}, & \text{if } s = 0, \\ \frac{\mu_s}{\lambda_s + \mu_s} \frac{1}{k!} \left(\frac{\mu_b}{\lambda_b} \right)^k \left[1 + \sum_{j=1}^{N_b} \frac{1}{j!} \left(\frac{\mu_b}{\lambda_b} \right)^j \right]^{-1}, & \text{if } s = 1. \end{cases} \quad (8.104)$$

The system is unavailable in all the states in which either the system has a platform failure, or in a system without APS, if a base repeater hosting the control channel fails, or the system even without platform failure has no working base repeater left. For a system without APS, the probability that one of the $(N_b - k)$ failed base repeaters happens to host the control channel is $(N_b - k)/N_b$. Let $U(N_b)$ denote the steady-state system unavailability. For both systems with and without APS, we thus express unavailability as

$$U(N_b) = \begin{cases} \sum_{k=0}^{N_b} \pi_{0,k}(N_b) + \sum_{k=0}^{N_b} \pi_{1,k}(N_b) \frac{N_b - k}{N_b}, & \text{without APS,} \\ \sum_{k=0}^{N_b} \pi_{0,k}(N_b) + \pi_{1,0}(N_b), & \text{with APS.} \end{cases} \quad (8.105)$$

For each of the states of the availability model of Figure 8.54, we now seek to obtain key performance indices. Performance indices of interest are the steady-state new-call blocking probability and handoff-call dropping probability. Given the number of available channels, the previous work in [HARI 2001, SUN 1999] provided

TABLE 8.6. Reward rates for systems without APS

State (s, k)	Reward rate	
	New-call blocking	Handoff-call dropping
$(0, k)$, for $k = 0, 1, \dots, N_b$	1	1
$(1, 0)$	1	1
$(1, k)$, for $k = 1, 2, \dots, N_b$	$1, \text{ if } kM - 1 \leq g$ $\frac{N_b - k}{N_b} + P_b(kM - 1, g) \frac{k}{N_b}, \text{ o.w. } ^a$	$\frac{N_b - k}{N_b} + P_d(kM - 1, g) \frac{k}{N_b}$

^a otherwise

formulas for these indices. We recall the results here [refer to equations (8.72) and (8.73)]. For a system having k nonfailed channels and g guard channels, let λ_1 be the rate of new-call arrivals, λ_2 be the handoff-call arrival rate, μ_1 be the rate for an ongoing call (new call or handoff call) to terminate, and μ_2 be the handoff-call departure rate. On the basis of a birth-death process, the new-call blocking probability is given as

$$P_b(k, g) = \frac{\sum_{n=k-g}^k \frac{A^{k-g}}{n!} A_1^{n-(k-g)}}{\sum_{n=0}^{k-g-1} \frac{A^n}{n!} + \sum_{n=k-g}^k \frac{A^{k-g}}{n!} A_1^{n-(k-g)}} \quad (8.106)$$

and the handoff-call dropping probability is given as

$$P_d(k, g) = \frac{\frac{A^{k-g}}{k!} A_1^g}{\sum_{n=0}^{k-g-1} \frac{A^n}{n!} + \sum_{n=k-g}^k \frac{A^{k-g}}{n!} A_1^{n-(k-g)}}, \quad (8.107)$$

where $A = (\lambda_1 + \lambda_2)/(\mu_1 + \mu_2)$ and $A_1 = \lambda_2/(\mu_1 + \mu_2)$.

We notice that calls can be blocked (or dropped) when the system is either *down* or *full*. The former type of loss is captured by the pure availability model; the latter type of loss is captured by the pure performance model. We now wish to combine the two types of losses. The primary vehicle for doing this is to determine pure performance losses for each of the availability model states and attach these loss probabilities as reward rates (or weights) to these states.

We list reward rates for the states of the availability model in Table 8.6 for systems without APS and Table 8.7 for system with APS. Let us first consider states of system being down.

Clearly, for both systems with and without APS, a cell is not able to accept any new call or handoff call if it has platform failure that corresponds to the states $(0, k)$ for $k = 0, 1, \dots, N_b$, or all base repeaters are down, which corresponds to the state $(1, 0)$. Therefore, reward rates of both overall new-call blocking and handoff-call

TABLE 8.7. Reward rates for systems with APS

State (s, k)	Reward rate	
	New-call blocking	Handoff-call dropping
$(0, k)$, for $k = 0, 1, \dots, N_b$	1	1
$(1, 0)$	1	1
$(1, k)$ for $k = 1, 2, \dots, N_b$	$1, \text{ if } kM - 1 \leq g$ $P_b(kM - 1, g), \text{ o.w.}^a$	$P_d(kM - 1, g)$

^a otherwise

dropping are 1s. In addition, for a system without APS, control channel down may occur in states $(1, k)$ for $k = 1, 2, \dots, N_b$ with probability $(N_b - k)/N_b$ and cause new-call blocking and handoff-call dropping. This corresponds to the rates with $(N_b - k)/N_b$ in the last row of Table 8.6.

All the states mentioned above contribute to system unavailability, $U(N_b)$. Hence, system unavailability, $U(N_b)$, also consists of one of the parts of the overall new-call blocking probability and handoff-call dropping probability. We now consider states in which the system is not undergoing a full outage caused by failures of platform, control channel (if system without APS), or all base repeaters being down. The corresponding states are $(1, k)$ for $k = 1, 2, \dots, N_b$. The total number of available channels for state $(1, k)$ is $kM - 1$. It is observed that new-call blocking probability and handoff-call dropping probability in these states are $P_b(kM - 1, g)$ and $P_d(kM - 1, g)$, respectively. Thus, these probabilities are used as reward rates to these states for overall new-call blocking and handoff-call dropping.

For a system without APS, we note that the probability of not having the control channel down in state $(1, k)$ for $k > 0$ is k/N_b . Therefore, the reward rates, $P_b(kM - 1, g)$ and $P_d(kM - 1, g)$, are also weighted by k/N_b (shown in the last row of Table 8.6). Also, if the number of idle channels is less than the number of guard channels, that is, $kM - 1 < g$ for states $(1, k)$, $k = 1, \dots, N_b$, a cell is not able to set up any new calls because all available channels are reserved for handoff calls. Hence, the new-call blocking reward rates assigned to the corresponding state are 1s. Now let $G = \lfloor (g + 1)/M \rfloor$. Summarizing Tables 8.6 and 8.7, the overall call blocking probability can be written as the expected steady-state reward rate (where w/ = with and w/o = without)

$$P_b^o(N_b, M, g) =$$

$$U(N_b) + \begin{cases} I_{(G>0)} \sum_{k=1}^G \pi_{1,k}(N_b) \left(\frac{k}{N_b} \right) \\ + \sum_{k=G+1}^{N_b} \pi_{1,k}(N_b) P_b(kM - 1, g) \left(\frac{k}{N_b} \right), & \text{w/o APS}, \\ I_{(G>0)} \sum_{k=1}^G \pi_{1,k}(N_b) \\ + \sum_{k=G+1}^{N_b} \pi_{1,k}(N_b) P_b(kM - 1, g), & \text{w/ APS}, \end{cases}$$

TABLE 8.8. Parameters used in numerical study

Parameter	Meaning	Value
N_b	Number of base repeaters	10
M	Number of channels/base repeater	8
λ_1	New-call arrival rate	20 calls/min
$1/\mu_1$	Mean call holding time	2.5 min
$1/\mu_2$	Mean time to handout	1.25 min
λ_s	Platform failure rate	1/year
$1/\mu_s$	Mean repair time of platform	8 h
λ_b	Base repeater failure rate	2/year
$1/\mu_b$	Mean repair time of base repeater	2 h

and similarly the overall handoff-call dropping probability can be given as

$$P_d^o(N_b, M, g) = U(N_b) + \begin{cases} \sum_{k=1}^{N_b} \pi_{1,k}(N_b) P_d(kM - 1, g) \left(\frac{k}{N_b} \right), & \text{w/o APS,} \\ \sum_{k=1}^{N_b} \pi_{1,k}(N_b) P_d(kM - 1, g), & \text{w/ APS.} \end{cases}$$

We should note that the hierarchical approach we have followed to obtain performability expressions is indeed an approximate solution to the model. Since we are interested in the steady-state performability measures rather than those in the transient regime, we neglect the fact that a failing base repeater may also bluntly discard all ongoing calls on it and therefore cause call dropping. We consider these simplifications to have a negligible effect on the steady-state measures. In fact, many authors have shown that the hierarchical decomposition method leads to very accurate results illustrated by numerical examples with realistic parameters in [TRIV 1992, SUN 1999].

We now present numerical results and Table 8.8 summarizes the parameters used.

In Figures 8.55 and 8.56, for both systems without APS and with APS, we plot new-call blocking probability, P_b^o , and handoff-call dropping probability, P_d^o , respectively, against new-call arrival rate, λ_1 . The plots show that both probabilities increase but stay nearly flat when new call traffic is low (< 20 calls/min). The probabilities then increase sharply after λ_1 exceeds 20 calls/min. The improvement by APS can be seen as reductions of P_b^o and P_d^o . Improvement remains steady (a greater than 30% relative reduction of both P_b^o and P_d^o) given low traffic but diminishes rapidly as traffic becomes heavier.

In the same figure, we also plot the percentage of unavailability U in the overall new-call blocking probability and handoff-call dropping probability to see to what extent failures of platform, base repeaters, and control channel (i.e., system being down) contribute to overall performability measures. It can be seen from the plots that system unavailability dominates under light traffic and becomes a less important factor under intense traffic when heavy traffic under limited system capacity becomes the major factor causing blocking and dropping.

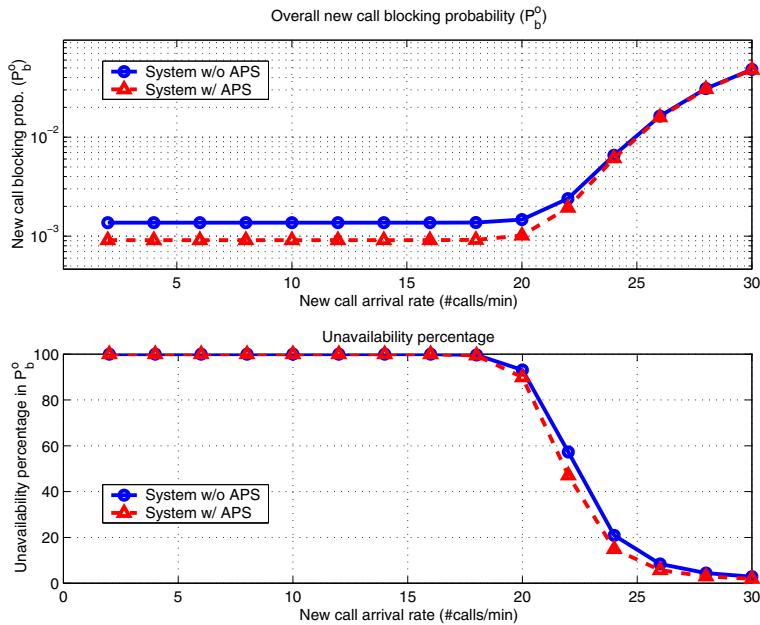


Figure 8.55. $P_b^o(N_b, M, g)$ versus g for systems without APS and with APS (top); percentage of unavailability $U(N_b)$ in $P_b^o(N_b, M, g)$ (bottom)

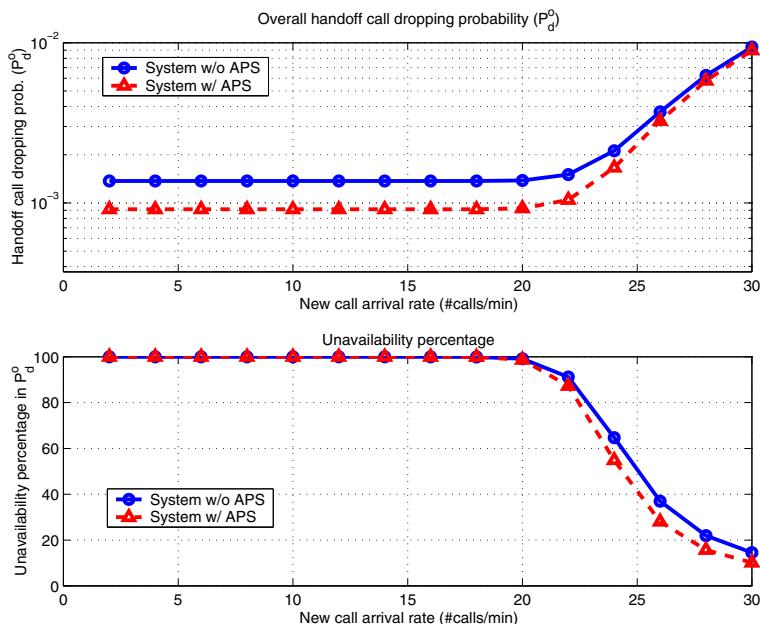


Figure 8.56. $P_d^o(N_b, M, g)$ versus g for systems without APS and with APS (top); percentage of unavailability $U(N_b)$ in $P_d^o(N_b, M, g)$ (bottom)

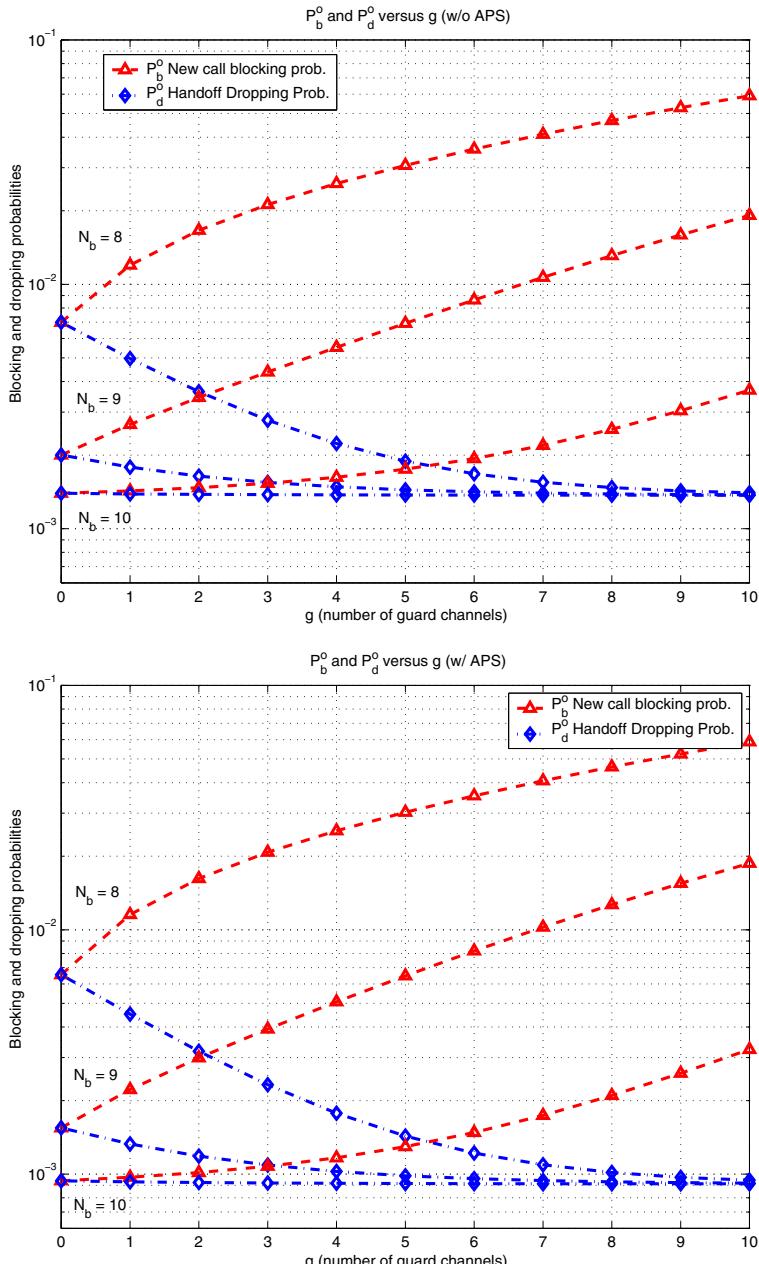


Figure 8.57. $P_b^o(N_b, M, g)$ and $P_d^o(N_b, M, g)$ versus N_b and g for system without APS (top) and with APS (bottom)

We plot curves of both overall new-call blocking probability, P_b^o , and overall handoff-call dropping probability against the number of guard channels g for $N_b = 8, 9, 10$ in Figure 8.57. It can be seen that, for each N_b , (1) $P_b^o = P_d^o$ when $g = 0$ and (2) increasing g results in a decrease in P_d^o and an increase in P_b^o . From Figure 8.57, it is also clear that when the number of base repeaters, N_b , increases, both curves of P_b^o and P_d^o move down, indicating the performance improvement.

The plot in Figure 8.57 also provides a graphic way to determine the optimum number of base repeaters and an optimum number of guard channels g . For example, for a system with APS, if we wish to ensure that $P_b^0 \leq 0.003$ and $P_d^0 \leq 0.002$, it now becomes easy with the plots in Figure 8.57 (bottom). We may draw two lines, $P_b^o = 0.003$ and $P_d^o = 0.002$. Pairs of triangle marks (Δ) for new-call blocking probability under line $P_b^o = 0.003$ and diamond marks (\diamond) for handoff-call dropping probability under line $P_d^o = 0.002$ consist of the set of possible solutions. We then choose the minimum N_b . In this case, $N_b^* = 9$ and $g^* = 0$ or 1.

‡

8.5 MARKOV CHAINS WITH ABSORBING STATES

With the exception of Section 8.3, our analysis has been concerned with irreducible Markov chains. We will introduce Markov chains with absorbing states through examples.

Example 8.34

Assume that we have a two-component parallel redundant system with a single repair facility of rate μ . Assume that the failure rate of both components is λ . When both components have failed, the system is considered to have failed and no recovery is possible. Let the number of properly functioning components be the state of the system. The state space is $\{0, 1, 2\}$, where 0 is the absorbing state. The state diagram is given in Figure 8.58.

Assume that the initial state of the Markov chain is 2; that is, $\pi_2(0) = 1$, $\pi_k(0) = 0$ for $k = 0, 1$. Then $\pi_j(t) = p_{2j}(t)$, and the system of differential equations (8.18) becomes

$$\begin{aligned} \frac{d\pi_2(t)}{dt} &= -2\lambda\pi_2(t) + \mu\pi_1(t), \\ \frac{d\pi_1(t)}{dt} &= 2\lambda\pi_2(t) - (\lambda + \mu)\pi_1(t), \\ \frac{d\pi_0(t)}{dt} &= \lambda\pi_1(t). \end{aligned} \quad (8.108)$$

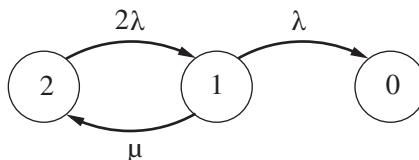


Figure 8.58. The state diagram for Example 8.34

Using the technique of Laplace transform, we can reduce this system to

$$\begin{aligned}s\bar{\pi}_2(s) - 1 &= -2\lambda\bar{\pi}_2(s) + \mu\bar{\pi}_1(s), \\ s\bar{\pi}_1(s) &= 2\lambda\bar{\pi}_2(s) - (\lambda + \mu)\bar{\pi}_1(s), \\ s\bar{\pi}_0(s) &= \lambda\bar{\pi}_1(s).\end{aligned}\tag{8.109}$$

Solving (8.109) for $\bar{\pi}_0(s)$, we get

$$\bar{\pi}_0(s) = \frac{2\lambda^2}{s[s^2 + (3\lambda + \mu)s + 2\lambda^2]}.$$

After an inversion, we can obtain $\pi_0(t)$, the probability that no components are operating at time $t \geq 0$. Let Y be the time to failure of the system; then $\pi_0(t)$ is the probability that the system has failed at or before time t . Thus the reliability of the system is

$$R(t) = 1 - \pi_0(t).$$

The Laplace transform of the failure density

$$f_Y(t) = -\frac{dR}{dt} = \frac{d\pi_0(t)}{dt},$$

is then given by

$$L_Y(s) = \bar{f}_Y(s) = s\bar{\pi}_0(s) - \pi_0(0) = \frac{2\lambda^2}{s^2 + (3\lambda + \mu)s + 2\lambda^2}.$$

The denominator can be factored so that

$$s^2 + (3\lambda + \mu)s + 2\lambda^2 = (s + \alpha_1)(s + \alpha_2),$$

and the preceding expression can be rearranged so that

$$L_Y(s) = \frac{2\lambda^2}{\alpha_1 - \alpha_2} \left(\frac{1}{s + \alpha_2} - \frac{1}{s + \alpha_1} \right),\tag{8.110}$$

where

$$\alpha_1, \alpha_2 = \frac{(3\lambda + \mu) \pm \sqrt{\lambda^2 + 6\lambda\mu + \mu^2}}{2}.$$

Inverting the transform in (8.110), we get

$$f_Y(t) = \frac{2\lambda^2}{\alpha_1 - \alpha_2} (e^{-\alpha_2 t} - e^{-\alpha_1 t})$$

and hence the reliability

$$R(t) = \int_t^\infty f_Y(x) dx = \frac{2\lambda^2}{\alpha_1 - \alpha_2} \left(\frac{e^{-\alpha_2 t}}{\alpha_2} - \frac{e^{-\alpha_1 t}}{\alpha_1} \right).\tag{8.111}$$

Then the MTTF of the system is given by

$$\begin{aligned} E[Y] &= \int_0^\infty R(t) dt = \frac{2\lambda^2}{\alpha_1 - \alpha_2} \left[\frac{1}{\alpha_2^2} - \frac{1}{\alpha_1^2} \right] = \frac{2\lambda^2(\alpha_1 + \alpha_2)}{\alpha_1^2 \alpha_2^2} \\ &= \frac{2\lambda^2(3\lambda + \mu)}{(2\lambda^2)^2} = \frac{3}{2\lambda} + \frac{\mu}{2\lambda^2}. \end{aligned} \quad (8.112)$$

Note that the MTTF of the two-component parallel redundant system, in the absence of a repair facility (i.e., $\mu = 0$), would have been equal to the first term, $3/(2\lambda)$, in expression (8.112). Therefore, the effect of a repair facility is to increase the mean life by $\mu/(2\lambda^2)$, or by a factor

$$\frac{\mu/2\lambda^2}{3/2\lambda} = \frac{\mu}{3\lambda}.$$

‡

Example 8.35

Next consider a modification of Example 8.34 proposed by Arnold [ARNO 1973] as a model of duplex processors of an electronic switching system. We assume that not all faults are recoverable and that c is the coverage factor denoting the conditional probability that the system recovers, given that a fault has occurred. The state diagram is now given by Figure 8.59. Note that this chain is *not* an example of a birth-death process.

Assume that the initial state is 2, so that

$$\pi_2(0) = 1, \quad \pi_0(0) = \pi_1(0) = 0.$$

Then $p_{2j}(t) = \pi_j(t)$ and the system of equation (8.18) yields

$$\begin{aligned} \frac{d\pi_2(t)}{dt} &= -2\lambda c \pi_2(t) - 2\lambda(1-c)\pi_2(t) + \mu\pi_1(t), \\ \frac{d\pi_1(t)}{dt} &= -(\lambda + \mu)\pi_1(t) + 2\lambda c \pi_2(t), \\ \frac{d\pi_0(t)}{dt} &= \lambda\pi_1(t) + 2\lambda(1-c)\pi_2(t). \end{aligned} \quad (8.113)$$

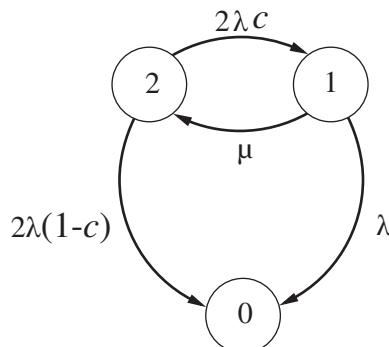


Figure 8.59. The state diagram of a duplex system with imperfect coverage

Using Laplace transforms as before, this system reduces to:

$$\begin{aligned}s\bar{\pi}_2(s) - 1 &= -2\lambda\bar{\pi}_2(s) + \mu\bar{\pi}_1(s), \\ s\bar{\pi}_1(s) &= -(\lambda + \mu)\bar{\pi}_1(s) + 2\lambda c\bar{\pi}_2(s), \\ s\bar{\pi}_0(s) &= \lambda\bar{\pi}_1(s) + 2\lambda(1 - c)\bar{\pi}_2(s).\end{aligned}\tag{8.114}$$

This system of linear equations can be solved to yield

$$\begin{aligned}\bar{\pi}_0(s) &= \frac{2\lambda}{s} \frac{s + \lambda + \mu - c(s + \mu)}{(s + 2\lambda)(s + \lambda + \mu) - 2\lambda\mu c}, \\ \bar{\pi}_1(s) &= \frac{2\lambda c}{(s + 2\lambda)(s + \lambda + \mu) - 2\lambda c\mu}, \\ \bar{\pi}_2(s) &= \frac{s + \lambda + \mu}{(s + 2\lambda)(s + \lambda + \mu) - 2\lambda c\mu}.\end{aligned}$$

As before, if X is the time to system failure, then

$$F_X(t) = \pi_0(t);$$

therefore

$$\begin{aligned}\bar{f}_X(s) &= L_X(s) = s\bar{\pi}_0(s) \\ &= \frac{2\lambda[(s + \lambda + \mu) - c(s + \mu)]}{(s + 2\lambda)(s + \lambda + \mu) - 2\lambda\mu c}.\end{aligned}$$

Let this be rewritten as

$$L_X(s) = \frac{2\lambda U}{V},$$

where $U = s + \lambda + \mu - c(s + \mu)$ and $V = (s + 2\lambda)(s + \lambda + \mu) - 2\lambda\mu c$. Instead of inverting this expression to obtain the distribution of X , we will be content with obtaining $E[X]$ using the moment generating property of Laplace transforms:

$$\begin{aligned}E[X] &= -\frac{dL_X}{ds}|_{s=0} \\ &= \frac{2\lambda[U(2s + 3\lambda + \mu) - V(1 - c)]}{V^2}|_{s=0} \\ &= \frac{2\lambda[(\lambda + \mu - \mu c)(3\lambda + \mu) - (2\lambda(\lambda + \mu) - 2\lambda\mu c)(1 - c)]}{[2\lambda(\lambda + \mu) - 2\lambda\mu c]^2},\end{aligned}$$

which, when reduced, finally gives us the required expression for mean time to system failure:

$$E[X] = \frac{\lambda(1 + 2c) + \mu}{2\lambda[\lambda + \mu(1 - c)]}.\tag{8.115}$$

Note that as c approaches 1, this expression reduces to the MTTF given in equation (8.112) for Example 8.34. As the coverage factor c approaches 0, expression (8.115)

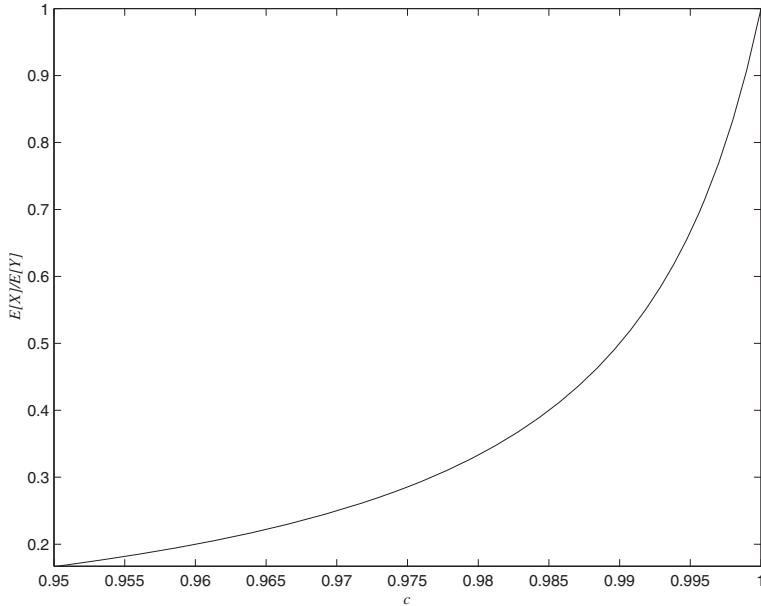


Figure 8.60. The effect of coverage on MTTF

yields the value $1/(2\lambda)$, which corresponds to the MTTF of a series system consisting of the two components. It should be clear that the system MTTF is critically dependent on the coverage factor, as can be seen from Figure 8.60, in which the ratio $E[X]/E[Y]$ is plotted as a function of the coverage factor c (we have taken $\lambda/\mu = 10^{-2}$). Recall that $E[Y]$ is the MTTF of a system with perfect coverage; it is obtained from equation (8.112).

#

Example 8.36 (NHCTMC Model of the Duplex System)

Consider a *duplex system* with two processors, each of which has a time-dependent failure rate $\lambda(t) = \lambda_0 \alpha t^{\alpha-1}$. Initially, both processors are operational (in state 2). The first fault may be detected with probability c_2 or not detected with probability $1 - c_2$. In the former case, the duplex system is still functioning in a degraded mode (state 1), while in the latter case, an unsafe failure has occurred (state *UF*). From state 1, the system can experience a safe shutdown with probability c_1 to enter state *SF* or an unsafe system failure with probability $1 - c_1$ to state *UF* when another processor failure occurs. The Markov model of the duplex system is shown in Figure 8.61.

The system shown in Figure 8.61 is a nonhomogeneous CTMC, because, as its name suggests, it contains one or more time-dependent transition rates. The transient behavior of a NHCTMC satisfies the linear system of first order differential equations:

$$\frac{d\pi(t)}{dt} = \boldsymbol{\pi}(t)Q(t), \quad \text{with } \pi_2(0) = 1. \quad (8.116)$$

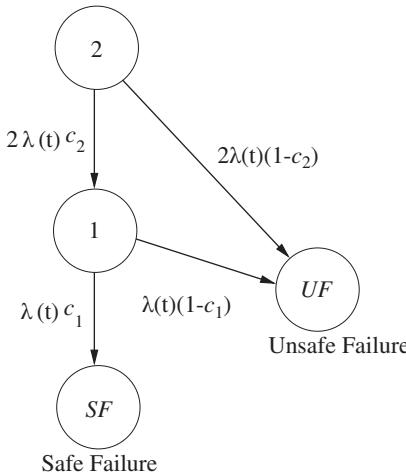


Figure 8.61. The NHCTMC model of the duplex system

At first, it may be thought that the solution to this equation may be written as $\pi(t) = \pi(0)e^{\int_0^t Q(\tau)d\tau}$. But as has been shown [RIND 1995], this is not true in general. The time-dependent infinitesimal generator matrix for this problem can be factored as follows:

$$\begin{aligned} Q(t) &= \begin{bmatrix} -\lambda(t) & 0 & \lambda(t)c_1 & \lambda(t)(1-c_1) \\ 2\lambda(t)c_2 & -2\lambda(t) & 0 & 2\lambda(t)(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \lambda(t) \begin{bmatrix} -1 & 0 & c_1 & 1-c_1 \\ 2c_2 & -2 & 0 & 2(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \lambda(t)W, \end{aligned}$$

where

$$W = \begin{bmatrix} -1 & 0 & c_1 & 1-c_1 \\ 2c_2 & -2 & 0 & 2(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In such cases when the time-dependent $Q(t)$ matrix can be factored into a time-independent matrix and a scalar function of time, the solution to the equation (8.116) is given by

$$\pi(t) = \pi(0)e^{[\int_0^t \lambda(\tau)d\tau]W}.$$

Hence we can define an average failure rate:

$$\bar{\lambda} = \frac{1}{t} \int_0^t \lambda(\tau)d\tau,$$

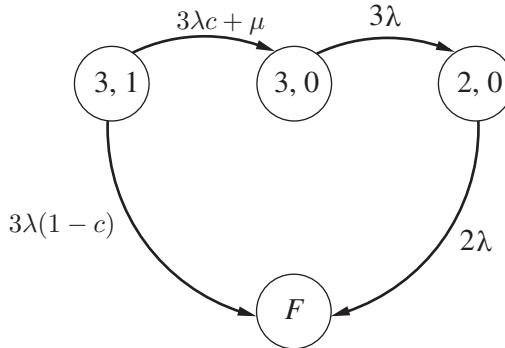


Figure 8.62. The state diagram of a TMR system with spare and imperfect coverage

and get the solution to the NHCTMC by solving a homogeneous CTMC with the generator matrix:

$$\bar{Q} = W\bar{\lambda}.$$

We use such a method for this simple example. To get numerical results, we choose the parameters as in Rindos *et al.* [RIND 1995]: $\lambda_0 = 0.0001$, $\alpha = 2.0$, $c_1 = 0.9$, and $c_2 = 0.9999$. At time $t = 100$ h, the system unreliability $\pi_{SF}(t) + \pi_{UF}(t)$ is calculated to be 0.39962291 and system unsafety $\pi_{UF}(t)$, to be 0.04004111.

#

Example 8.37

Consider a system with three active units and one spare. The active configuration is operated in TMR mode. An active unit has a failure rate λ , while a standby spare unit has a failure rate μ . If i active units and j standby units are functioning properly, then the state of the system will be denoted by (i, j) . The state F denotes the system failure state. We assume that in state $(3, 1)$, failure of an active unit can be recovered with probability c (≤ 1). The state diagram of the homogeneous CTMC is given in Figure 8.62.

Differential equations for this CTMC are written as follows:

$$\frac{d\pi_{3,1}}{dt} = -(3\lambda + \mu)\pi_{3,1}(t),$$

$$\frac{d\pi_{3,0}}{dt} = -3\lambda\pi_{3,0}(t) + (3\lambda c + \mu)\pi_{3,1}(t),$$

$$\frac{d\pi_{2,0}}{dt} = -2\lambda\pi_{2,0}(t) + 3\lambda\pi_{3,0}(t),$$

$$\frac{d\pi_F}{dt} = 3\lambda(1 - c)\pi_{3,1}(t) + 2\lambda\pi_{2,0}(t),$$

where we have assumed that the initial state is (3,1) so that $\pi_{3,1}(0) = 1$, and $\pi_{i,j}(0) = 0 = \pi_F(0)$ otherwise. Using Laplace transforms, we get

$$\begin{aligned}s\bar{\pi}_{3,1}(s) - 1 &= -(3\lambda + \mu)\bar{\pi}_{3,1}(s), \\ s\bar{\pi}_{3,0}(s) &= -3\lambda\bar{\pi}_{3,0}(s) + (3\lambda c + \mu)\bar{\pi}_{3,1}(s), \\ s\bar{\pi}_{2,0}(s) &= -2\lambda\bar{\pi}_{2,0}(s) + 3\lambda\bar{\pi}_{3,0}(s), \\ s\bar{\pi}_F(s) &= 3\lambda(1 - c)\bar{\pi}_{3,1}(s) + 2\lambda\bar{\pi}_{2,0}(s).\end{aligned}$$

Solving this system of equations, we get

$$\begin{aligned}\bar{\pi}_{3,1}(s) &= \frac{1}{s + 3\lambda + \mu}, \\ \bar{\pi}_{3,0}(s) &= \frac{3\lambda c + \mu}{(s + 3\lambda + \mu)(s + 3\lambda)}, \\ \bar{\pi}_{2,0}(s) &= \frac{3\lambda(3\lambda c + \mu)}{(s + 3\lambda + \mu)(s + 3\lambda)(s + 2\lambda)},\end{aligned}$$

and

$$s\bar{\pi}_F(s) = \frac{3\lambda(1 - c)}{(s + 3\lambda + \mu)} + \frac{6\lambda^2(3\lambda c + \mu)}{(s + 2\lambda)(s + 3\lambda)(s + 3\lambda + \mu)}.$$

If X is the time to failure of the system, then $\pi_F(t)$ is the distribution function of X . It follows that

$$L_X(s) = \bar{f}_X(s) = s\bar{\pi}_F(s) - \pi_F(0) = s\bar{\pi}_F(s),$$

which can be rewritten as:

$$\bar{f}_X(s) = \frac{3\lambda + \mu}{(s + 3\lambda + \mu)} \left[\frac{3\lambda(1 - c)}{(3\lambda + \mu)} + \frac{3\lambda c + \mu}{3\lambda + \mu} \left\{ \frac{2\lambda}{(s + 2\lambda)} \cdot \frac{3\lambda}{(s + 3\lambda)} \right\} \right]. \quad (8.117)$$

The expression outside the square brackets is the Laplace–Stieltjes transform of EXP ($3\lambda + \mu$), while the expression within the braces is the LST of HYPO ($2\lambda, 3\lambda$). Therefore, the system lifetime X has the stage-type distribution shown in Figure 8.63. It follows that system MTTF is

$$\begin{aligned}E[X] &= \frac{1}{3\lambda + \mu} + \frac{3\lambda c + \mu}{3\lambda + \mu} \left[\frac{1}{2\lambda} + \frac{1}{3\lambda} \right] \\ &= \frac{1}{3\lambda + \mu} + \frac{3\lambda c + \mu}{3\lambda + \mu} \cdot \frac{5}{6\lambda}.\end{aligned}$$

This can be verified by computing $-d\bar{f}_X/ds|_{s=0}$, which is easy from (8.117). ‡

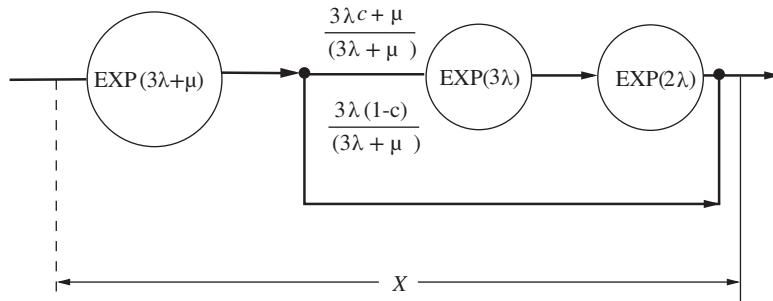


Figure 8.63. The stage-type lifetime distribution for the system of Example 8.37

So far we used a transform-based approach for computing the MTTF. Practical problems often give rise to very large CTMCs where the transform-based analysis becomes difficult or even impossible. We now describe a method of computing the MTTF via a linear system of equations.

We begin with equation (8.19). Denote the set of all system failure states by set D (all the absorbing states belong to set D), and the set of all states in which the system is operational by set U . Let \hat{Q} denote the submatrix of Q pertaining to states in U only. Similarly, let $\hat{\pi}(0)$ be the subvector of $\pi(0)$ pertaining to states in U . Now, if $\tau_i = \lim_{t \rightarrow \infty} L_i(t)$ exists for $i \in U$, equation (8.19) becomes

$$\hat{\tau}\hat{Q} = -\hat{\pi}(0). \quad (8.118)$$

Because τ_i is the expected time the CTMC spends in (nonfailure) state i until system failure (entering D), the mean time to (system) failure (MTTF) is given by

$$\text{MTTF} = \sum_{i \in U} \tau_i.$$

For small problems, this system can be solved symbolically as we shall see in several examples. For larger CTMCs numerical solution is adopted. Iterative numerical solution of equation (8.118) using methods such as successive overrelaxation (SOR) have proved to be efficient for most MTTF problems. But as Heidelberger *et al.* [HEID 1996] pointed out, solving equation (8.118) requires an extremely large number of iterations for highly reliable systems, where system failure is a rare event. They proposed a two-step procedure for accelerating MTTF computation, in which one or more frequently visited states are made absorbing artificially. The convergence rate is improved significantly; this method is implemented in both SHARPE [SAHN 1996] and SPNP [CIAR 1993].

Example 8.38

Consider a variation of the workstations and file server (WFS) example in which there is no repair action when the system fails. We are now interested in the system reliability and system MTTF with and without repair for the components (see Figures 8.64 and 8.65, respectively).

States (2,0), (1,0), and (0,1) in Figure 8.34 are absorbing states in both the CTMCs for the system with and without repair for the components, and hence are merged into a single state F . The corresponding generator matrices for Figures 8.64 and 8.65 are Q_R and Q_{NR} . New matrices \hat{Q}_R and \hat{Q}_{NR} are obtained from Q_R and Q_{NR} , respectively, by restricting the generator matrices only to the transient states

$$\hat{Q}_R = \begin{bmatrix} -(\lambda_f + 2\lambda_w) & 2\lambda_w \\ \mu_w & -(\mu_w + \lambda_f + \lambda_w) \end{bmatrix},$$

and

$$\hat{Q}_{NR} = \begin{bmatrix} -(\lambda_f + 2\lambda_w) & 2\lambda_w \\ 0 & -(\lambda_f + \lambda_w) \end{bmatrix}.$$

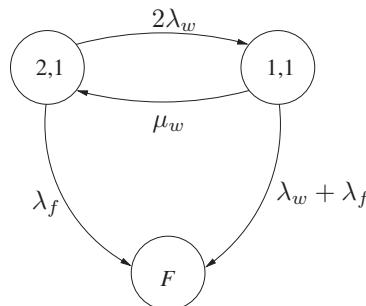


Figure 8.64. CTMC reliability model for the WFS example with repair

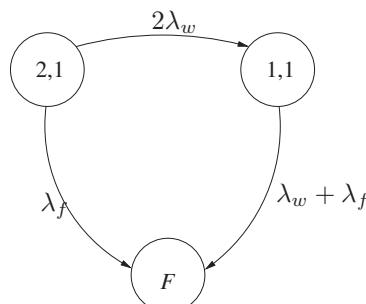


Figure 8.65. CTMC reliability model for the WFS example without repair

The system reliability is obtained by adding the transient probabilities of being in the working states (2,1) and (1,1):

$$R(t) = \pi_{2,1}(t) + \pi_{1,1}(t),$$

where the state probabilities are obtained by solving Kolmogorov equations using the two matrices shown above.

Also

$$\begin{aligned}\hat{\tau}_R \hat{Q}_R &= -\hat{\pi}_R(0) \\ \hat{\tau}_{NR} \hat{Q}_{NR} &= -\hat{\pi}_{NR}(0).\end{aligned}$$

The system MTTF is obtained by

$$\text{MTTF} = \tau_{2,1} + \tau_{1,1}.$$

For the system with repair, the following equations are solved

$$\begin{aligned}\mu_w \tau_{1,1} - (\lambda_f + 2\lambda_w) \tau_{2,1} &= -1 \\ -(\mu_w + \lambda_f + \lambda_w) \tau_{1,1} + 2\lambda_w \tau_{2,1} &= 0\end{aligned}$$

to get

$$\tau_{1,1} = \frac{2\lambda_w}{\lambda_f \mu_w + (\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}, \quad (8.119)$$

$$\tau_{2,1} = \frac{\mu_w + \lambda_f + \lambda_w}{\lambda_f \mu_w + (\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}. \quad (8.120)$$

Hence MTTF for the system with repair is

$$\text{MTTF}_R = \frac{\mu_w + \lambda_f + 3\lambda_w}{\lambda_f \mu_w + (\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}.$$

For the system without repair the following equations are solved:

$$\begin{aligned}-(\lambda_f + 2\lambda_w) \tau_{2,1} &= -1 \\ -(\lambda_f + \lambda_w) \tau_{1,1} + 2\lambda_w \tau_{2,1} &= 0\end{aligned}$$

to get

$$\begin{aligned}\tau_{1,1} &= \frac{2\lambda_w}{(\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)} \\ \tau_{2,1} &= \frac{1}{(\lambda_f + 2\lambda_w)}.\end{aligned}$$

Hence the MTTF for the system without repair is

$$\text{MTTF}_{NR} = \frac{\lambda_f + 3\lambda_w}{(\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}.$$



Figure 8.66. RBD of workstations and file server system

Observe that since file server repair is never considered, there is no repair dependency and hence the workstation and file server subsystems are independent. This fact can be exploited and a hierarchical reliability model can be developed. The top-level model is an RBD (reliability block diagram) with two blocks in series as shown in Figure 8.66. The first block can be expanded into a three-state CTMC (same structure as in Figure 8.58, but with subscript w added to the transition rates) at the lower level representing the reliability model with repair of the workstation subsystem, while the other block represents file server failure. The reliability $R_w(t)$ and the mean time to failure MTTF_w of the workstation subsystem can be obtained from equations (8.111) and (8.112), respectively as

$$R_w^{(R)}(t) = \frac{2\lambda_w^2}{\alpha_1 - \alpha_2} \left(\frac{e^{-\alpha_2 t}}{\alpha_2} - \frac{e^{-\alpha_1 t}}{\alpha_1} \right)$$

$$\text{MTTF}_w^{(R)} = \frac{3}{2\lambda_w} + \frac{\mu_w}{2\lambda_w^2},$$

where

$$\alpha_1, \alpha_2 = \frac{(3\lambda_w + \mu_w) \pm \sqrt{\lambda_w^2 + 6\lambda_w\mu_w + \mu_w^2}}{2}.$$

The reliability and MTTF of the file server subsystem are

$$R_f(t) = e^{-\lambda_f t} \quad (8.121)$$

$$\text{MTTF}_f = \frac{1}{\lambda_f}. \quad (8.122)$$

The reliability and MTTF of the workstations and file server system with repair is obtained from the RBD of Figure 8.66 as

$$R_R(t) = R_w^{(R)}(t) \cdot R_f(t)$$

$$= \frac{2\lambda_w^2}{\alpha_1 - \alpha_2} \left(\frac{e^{-\alpha_2 t}}{\alpha_2} - \frac{e^{-\alpha_1 t}}{\alpha_1} \right) \cdot e^{-\lambda_f t},$$

$$\text{MTTF}_R = \int_0^\infty R_R(t) dt$$

$$= \frac{2\lambda_w^2}{\alpha_2(\alpha_1 - \alpha_2)(\alpha_2 + \lambda_f)} - \frac{2\lambda_w^2}{\alpha_1(\alpha_1 - \alpha_2)(\alpha_1 + \lambda_f)}$$

$$= \frac{\mu_w + \lambda_f + 3\lambda_w}{\lambda_f\mu_w + (\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}.$$

For the case without repair, too, a hierarchical reliability model with the RBD of Figure 8.66 as the top-level model can be used. In this case, the reliability of the workstation subsystem is obtained as in Example 8.17 (with $n = 2$):

$$\begin{aligned} R_w^{(NR)}(t) &= 1 - (1 - e^{-\lambda_w t})^2 \\ &= 2e^{-\lambda_w t} - e^{-2\lambda_w t}. \end{aligned}$$

The MTTF of the workstation subsystem is obtained as

$$\text{MTTF}_w^{(NR)} = \frac{3}{2\lambda_w}.$$

The reliability and MTTF of the file server subsystem are the same as in equations (8.121) and (8.122), respectively.

The reliability and MTTF of the WFS system without repair are obtained as

$$\begin{aligned} R_{NR}(t) &= R_w^{(NR)}(t) \cdot R_f(t) \\ &= [1 - (1 - e^{-\lambda_w t})^2] \cdot e^{-\lambda_f t} \\ &= 2e^{-(\lambda_f + \lambda_w)t} - e^{-(\lambda_f + 2\lambda_w)t} \\ \text{MTTF}_{NR} &= \int_0^\infty R_{NR}(t) \\ &= \frac{2}{\lambda_f + \lambda_w} - \frac{1}{\lambda_f + 2\lambda_w} \\ &= \frac{\lambda_f + 3\lambda_w}{(\lambda_f + \lambda_w)(\lambda_f + 2\lambda_w)}. \end{aligned}$$

For more examples of hierarchical reliability models, see Sahner et al. [SAHN 1996].

#

Example 8.39 [ORTA 1999]

The vulnerabilities exhibited by an operational computing system can be represented in a privilege graph. In such a graph, a node A represents a set of privileges owned by a user or a set of users (e.g., a UNIX group). An arc represents a vulnerability. An arc exists from node A to node B if there is a method allowing a user owning A privileges to obtain those of node B . If a path exists between an attacker node and a target node, then security breach can potentially occur since an attacker can exploit system vulnerabilities to obtain target privileges. Situations where attackers may give up or interrupt their process are not considered; that is, the attack stops when the target is reached.

Assuming that at each newly visited node of the privilege graph, the attacker chooses one of the elementary attacks that can be issued from that node only (memoryless property) and assigning to each arc a rate at which the attacker succeeds with the corresponding elementary attack, the privilege graph is transformed into a CTMC. Figure 8.67 gives an example of such a CTMC.

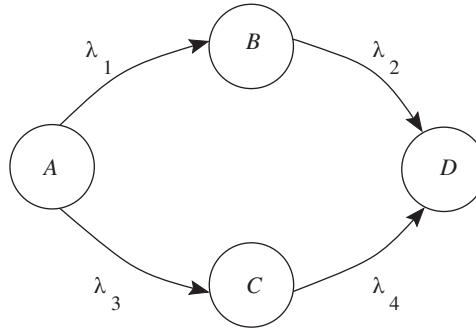


Figure 8.67. CTMC for operational security

Various probabilistic measures can be derived from the Markov model, including the mean effort (or time) for a potential attacker to reach the specified target, denoted as mean effort to (security) failure (METF), by analogy with mean time to failure (MTTF). This measure allows easy physical interpretation of the results—the higher the METF, the better the security.

The matrix \hat{Q} obtained from generator matrix Q by restricting only to the transient states is

$$\hat{Q} = \begin{bmatrix} -(\lambda_1 + \lambda_3) & \lambda_1 & \lambda_3 \\ 0 & -\lambda_2 & 0 \\ 0 & 0 & -\lambda_4 \end{bmatrix}.$$

Using equation (8.118) where $\hat{\tau} = [\tau_A, \tau_B, \tau_C]$ and $\hat{\pi}(0) = [1, 0, 0]$ we get

$$\begin{aligned}\tau_A &= \frac{1}{\lambda_1 + \lambda_3} \\ \tau_B &= \frac{\lambda_1}{\lambda_2(\lambda_1 + \lambda_3)} \\ \tau_C &= \frac{\lambda_3}{\lambda_4(\lambda_1 + \lambda_3)}.\end{aligned}$$

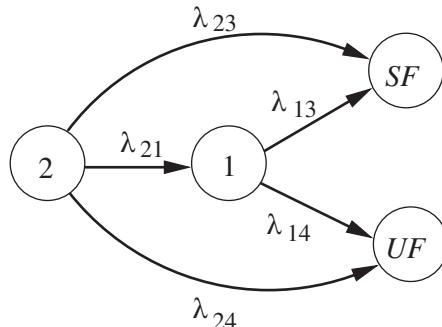
It follows that the METF is

$$\text{METF} = \sum_{i \in \{A, B, C\}} \tau_i = \frac{1}{\lambda_1 + \lambda_3} \left(1 + \frac{\lambda_1}{\lambda_2} + \frac{\lambda_3}{\lambda_4} \right).$$

#

Example 8.40 [LAPR 1995]

Consider a recovery block (RB) architecture implemented on a dual processor system that is able to tolerate one hardware fault and one software fault. The hardware faults can be tolerated due to the hot standby hardware component with a duplication of the RB software and a concurrent comparator for acceptance tests. The CTMC state diagram is shown in Figure 8.68. The transition rates and their

**Figure 8.68.** The state diagram for Example 8.40**TABLE 8.9. Parameters for Example 8.40**

Transition rate	Value	Meaning
λ_{21}	$2c\lambda_H$	covered hardware component failure
λ_{23}	$2\bar{c}\lambda_H + \lambda_{SD}$	Not covered hardware component failure or detected RB failure
λ_{24}	λ_{SU}	undetected RB failure
λ_{13}	$c\lambda_H + \lambda_{SD}$	detected RB failure or covered hardware component failure
λ_{14}	$\bar{c}\lambda_H + \lambda_{SU}$	Not covered hardware component failure or undetected RB failure

meanings are given in Table 8.9. In the table, λ_H denotes the hardware component failure rate; λ_{SD} and λ_{SU} denote respectively the detected and undetected failure rates of the recovery block software, and c is the hardware coverage factor with $\bar{c} = 1 - c$. Note that states SF and UF represent the safe and unsafe failure states, respectively.

The system is initially in state 2; that is $\pi_2(0) = 1, \pi_k(0) = 0$ for all other states. Then the system of differential equations (8.18) yields:

$$\begin{aligned}
 \frac{d\pi_2(t)}{dt} &= -(\lambda_{21} + \lambda_{23} + \lambda_{24})\pi_2(t), \\
 \frac{d\pi_1(t)}{dt} &= -(\lambda_{13} + \lambda_{14})\pi_1(t) + \lambda_{21}\pi_2(t), \\
 \frac{d\pi_{SF}(t)}{dt} &= \lambda_{23}\pi_2(t) + \lambda_{13}\pi_1(t), \\
 \frac{d\pi_{UF}(t)}{dt} &= \lambda_{24}\pi_2(t) + \lambda_{14}\pi_1(t),
 \end{aligned} \tag{8.123}$$

Using Laplace transforms as before, the above system is reduced to:

$$\begin{aligned}s\bar{\pi}_2(s) - 1 &= -(\lambda_{21} + \lambda_{23} + \lambda_{24})\bar{\pi}_2(s), \\ s\bar{\pi}_1(s) &= -(\lambda_{13} + \lambda_{14})\bar{\pi}_1(s) + \lambda_{21}\bar{\pi}_2(s); \\ s\bar{\pi}_{SF}(s) &= \lambda_{23}\bar{\pi}_2(s) + \lambda_{13}\bar{\pi}_1(s); \\ s\bar{\pi}_{UF}(s) &= \lambda_{24}\bar{\pi}_2(s) + \lambda_{14}\bar{\pi}_1(s);\end{aligned}\tag{8.124}$$

Solving the above system of equations for $\bar{\pi}_i(s)$ and using inverse Laplace transforms, we get $\pi_i(t)$. Thus the reliability of the system is:

$$\begin{aligned}R(t) &= \pi_2(t) + \pi_1(t) \\ &= 2ce^{-(\lambda_H + \lambda_S)t} - (2c - 1)e^{-(2\lambda_H + \lambda_S)t}\end{aligned}\tag{8.125}$$

where $\lambda_S = \lambda_{SD} + \lambda_{SU}$.

Similarly, the absorption probability to the safe failure state is:

$$\begin{aligned}P_{SF} &= \pi_{SF}(\infty) \\ &= \frac{2\bar{c}\lambda_H + \lambda_{SD}}{2\lambda_H + \lambda_S} + \frac{2c\lambda_H(c\lambda_H + \lambda_{SD})}{(2\lambda_H + \lambda_S)(\lambda_H + \lambda_S)}\end{aligned}\tag{8.126}$$

And the absorption probability to the unsafe failure state is:

$$\begin{aligned}P_{UF} &= \pi_{UF}(\infty) \\ &= \frac{\lambda_{SU}}{2\lambda_H + \lambda_S} + \frac{2c\lambda_H(\bar{c}\lambda_H + \lambda_{SU})}{(2\lambda_H + \lambda_S)(\lambda_H + \lambda_S)}\end{aligned}\tag{8.127}$$

#

Example 8.41 (Conditional MTTF of a Fault-Tolerant System)

Consider the homogeneous CTMC models of three commonly used fault-tolerant system architectures. The simplex system S consists of a single processor. When a fault occurs in the processor at the rate λ , the fault can be detected and the system shuts down safely with probability c_s . On the other hand, with probability $1 - c_s$, the system does not detect the fault and it experiences an unsafe (hazardous) failure. The probability c_s is the coverage factor for a simplex system meaning the probability the system *detects* the fault once it has occurred. Figure 8.69a shows the Markov model of this system. In the operational state i , i ($i \geq 1$) processors are operating. The states *SF* and *UF* represent safe and unsafe failure states, respectively.

The duplex system (D) consists of two identical processors executing the same task in parallel. When a processor generates a fault at the rate λ , the fault is covered with probability c_d and the system is shutdown safely (Figure 8.69b). The coverage factor c_d in this case is the probability that the system *detects* the fault. After the execution of a task, the outputs from two processors are compared with each other.

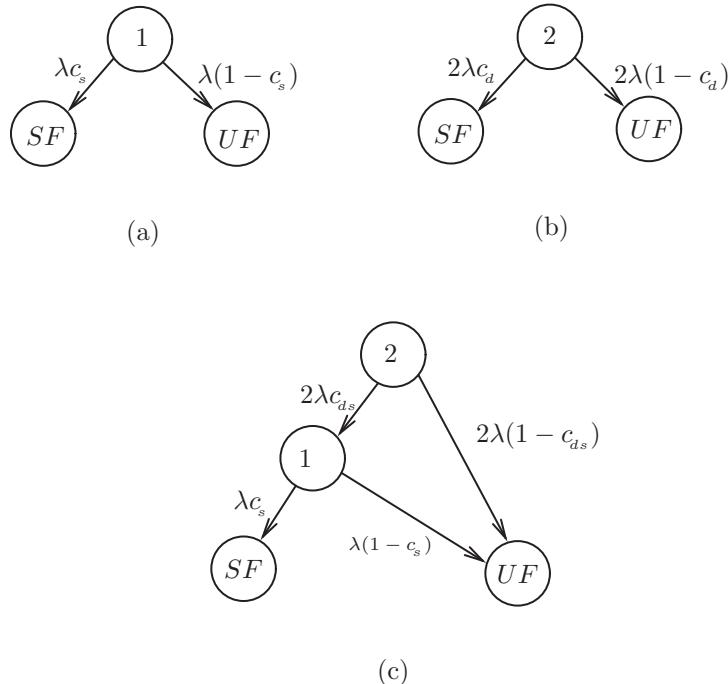


Figure 8.69. Architectures of fault-tolerant systems: (a) system S; (b) system D; (c) system DS

Because of the comparison, duplex system is more likely to detect a fault than the simplex system. Thus the probability c_d is naturally larger than c_s ($0 \leq c_s < c_d \leq 1$).

The duplex system reconfigurable to the simplex system (DS) also consists of two processors executing the same task in parallel. When a fault occurs, the system suffers unsafe failure with probability $1 - c_{ds}$. The coverage factor c_{ds} is the probability of *not only detecting the fault but also reconfiguring the system* in order to keep operating with the nonfaulty processor. Since c_{ds} is the probability that both the events are successful, c_{ds} is naturally smaller than c_d ($0 \leq c_{ds} < c_d \leq 1$).

We compare the three architectures with respect to the probability of unsafe failure, the mean time to failure (MTTF) of the system and the **conditional** MTTF to unsafe failure [CHWA 1998]. The conditional MTTF to A , $\text{MTTF}_{|A}$, is defined as the expected time until absorption to a state in set A given that the system is eventually absorbed into A , where A is a subset of the absorbing states of the Markov reliability model, which corresponds to a given failure condition.

First, we show a solution method for computing the conditional MTTF. Consider a CTMC with m ($m \geq 1$) absorbing states as in Figure 8.70. The set of all absorbing states of this CTMC is $S_A = \{a_1, a_2, \dots, a_m\}$. Let Y be a random variable representing the time for the CTMC to be absorbed into $A \in S_A$. Partition the infinitesimal generator matrix Q so that the transient states, labeled as T appear first, followed by states in A , and then followed by the remaining absorbing states,

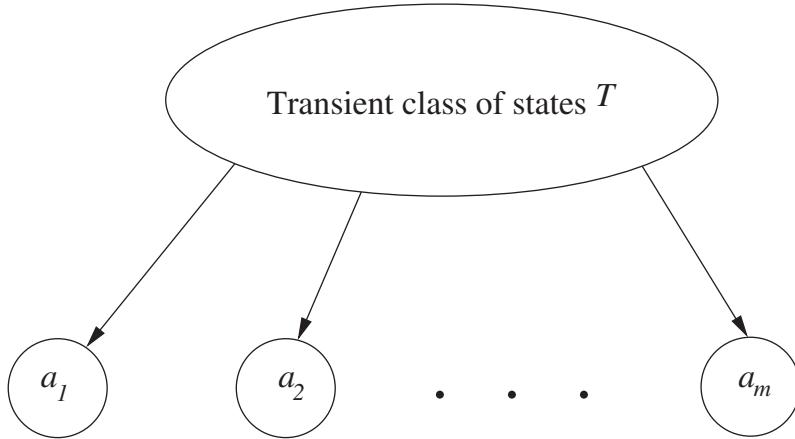


Figure 8.70. A CTMC with m absorbing states

labeled collectively as $B = S_A - A$. It is clear that for all j , $q_{ij} = 0$ for any state i in A or B :

$$Q = \begin{bmatrix} Q_{TT} & Q_{TA} & Q_{TB} \\ \mathbf{0}_{1 \times |T|} & 0 & \mathbf{0}_{1 \times |B|} \\ \mathbf{0}_{|B| \times |T|} & \mathbf{0}_{|B| \times 1} & \mathbf{0}_{|B| \times |B|} \end{bmatrix}.$$

Here Q_{TT} is the partition of the generator matrix consisting of the states in T , Q_{TA} has the transition rates from states in T to states in A and similarly Q_{TB} has the transition rates from states in T to states in B . We assume A to have a single state so that Q_{TA} is a $|T| \times 1$ matrix. In case A consists of multiple absorbing states, we can easily aggregate these states into a single state. Note that the aggregation does not imply an approximate solution in this case. Partition the state probability vector $\boldsymbol{\pi}(t) = [\boldsymbol{\pi}_T(t), \boldsymbol{\pi}_A(t), \boldsymbol{\pi}_B(t)]$, where $\boldsymbol{\pi}_T(t)$, $\boldsymbol{\pi}_A(t)$, $\boldsymbol{\pi}_B(t)$ are the state probability vectors for states in the sets T , A , and B respectively.

The **absorption probability** is the probability that the CTMC is absorbed into A with a given initial state pmf, namely, $P\{X(\infty) \in A\} = \boldsymbol{\pi}_A(\infty)$. The absorption probability to A is obtained by [CHWA 1998]:

$$\boldsymbol{\pi}_A(\infty) = \tau_T Q_{TA}, \quad (8.128)$$

where τ_T is the solution of the linear system:

$$\tau_T Q_{TT} = -\boldsymbol{\pi}_T(0). \quad (8.129)$$

Let \mathbf{e}_T be a column vector of size $|T|$ with all 1s, then recall that the unconditional MTTF is given by

$$\text{MTTF} = \tau_T \mathbf{e}_T. \quad (8.130)$$

TABLE 8.10. Dependability measures for the three architectures

Measures	Architecture S	Architecture D	Architecture DS
MTTF	$\frac{1}{\lambda}$	$\frac{1}{2\lambda}$	$\frac{1}{2\lambda} + \frac{c_{ds}}{\lambda}$
$\pi_{UF}(\infty)$	$1 - c_s$	$1 - c_d$	$1 - c_s c_{ds}$
$MTTF _{UF}$	$\frac{1}{\lambda}$	$\frac{1}{2\lambda}$	$\frac{1 + 2c_{ds} - 3c_s c_{ds}}{2\lambda(1 - c_s c_{ds})}$

The conditional MTTF to A is obtained by [CHWA 1998]:

$$MTTF|_A = \frac{E[Y \text{ and } Y < \infty]}{\pi_A(\infty)} = \frac{\boldsymbol{\theta}_T Q_{TA}}{\tau_T Q_{TA}}, \quad (8.131)$$

where $\boldsymbol{\theta}_T$ is the solution of the linear system:

$$\boldsymbol{\theta}_T Q_{TT} = -\tau_T.$$

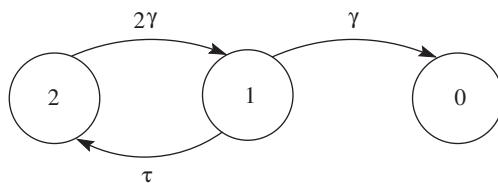
Using these methods, we compute the measures for the three architectures as shown in Table 8.10.

‡

Example 8.42

We return to the multiprocessor model of Example 8.32, but we now consider system failure state 0 as absorbing. We also simplify the model by using $n = 2$ and $c = 1$, and assuming instantaneous reconfiguration ($\delta = \infty$). Hence the system reliability model is as shown in Figure 8.71.

Since task arrivals occur at the rate λ and task service time is $EXP(\mu)$, when the reliability model is in state 2, the performance can be modeled by an $M/M/2/b$ queue. From equation (8.103), we have the buffer full probability $q_b(2)$. Similarly, in state 1 of the reliability model, we have the buffer full probability as $q_b(1)$. Furthermore, the probability of violating a deadline in state 2 is given by $P(R_b(2) > d)$,

**Figure 8.71.** Multiprocessor system reliability model

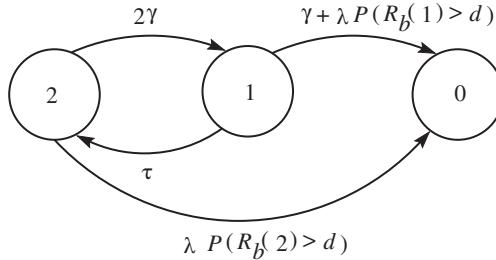


Figure 8.72. Multiprocessor system reliability model with hard-deadline failures

and that in state 1 is given by $P(R_b(1) > d)$. We make the following reward rate assignment to the states:

$$\begin{aligned} r_2 &= \lambda[1 - q_b(2)][P(R_b(2) \leq d)], \\ r_1 &= \lambda[1 - q_b(1)][P(R_b(1) \leq d)] \end{aligned}$$

and

$$r_0 = 0.$$

With this reward assignment, if we compute the expected accumulated reward until absorption, then we will obtain the approximate number of tasks successfully completed (that is, within deadline) until system failure:

$$E[Y(\infty)] = r_2\tau_2 + r_1\tau_1,$$

where τ_2 and τ_1 are from equation (8.118). This is a hierarchical performability model.

#

Example 8.43

In the previous example, deadlines were soft but now we consider a hard deadline so that if an accepted job fails to complete within the deadline, we will consider the system to have failed. To reflect this, we modify the reliability model of Figure 8.71 to introduce deadline (or dynamic) failures [SHIN 1986] as shown in Figure 8.72. Using the τ method, we can compute the values of τ_2 and τ_1 for the CTMC in the figure and hence get the system MTTF that includes the effect of dynamic failures: $MTTF = \tau_2 + \tau_1$.

#

Problems

- Assuming $\lambda = 10^{-4}$, and $\mu = 1$, compare the reliability of a two-unit parallel redundant system with repair (Example 8.34) with that of a two-unit parallel redundant system without repair. Also plot the two expressions on the same graph.

2. Compute the MTTF for Example 8.34 assuming that state 1 is the initial state. You may use the τ method. Next compute MTTF_{eq} of the model in Figure 8.27 using the technique of Example 8.11. Show that they are identical. Explain the reason for this equivalence.
3. Solve Example 8.37 using the τ method [equation (8.118)].
4. Modify the structure of Example 8.34 so that it is a two-unit standby redundant system rather than a parallel redundant system. Assume that the failure rates of online and standby units are respectively given by λ_1 and λ_2 , where $\lambda_1 \geq \lambda_2 \geq 0$. Repair times are exponentially distributed with mean $1/\mu$, and the system is considered to have failed on the failure of the second unit before the first unit is repaired. Obtain expressions for system reliability and system MTTF, assuming that the detection and switching mechanisms are fault-free.
5. Solve the system of equations (8.124) in Example 8.40 to derive the expressions for $R(t)$, $\pi_{SF}(\infty)$, and $\pi_{UF}(\infty)$.
6. * Consider a two-unit standby redundant system where the spare failure rate is identical to the failure rate of the active unit. The system is modeled using the homogeneous CTMC as shown in Figure 8.P.2. Here δ is the detection-reconfiguration rate and c is the coverage factor. Solve the system for its reliability $R(t)$, using the methods developed in this section. Next solve for $R(t)$ using the convolution-integral approach developed in problem 6 at the end of Section 8.1. Finally, solve for $R(t)$ using the matrix series approach developed in problem 1, Section 8.1.

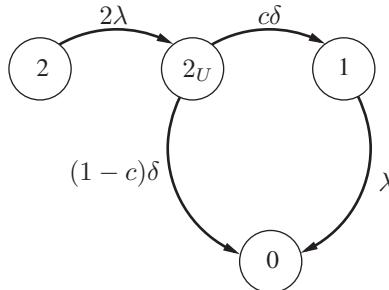


Figure 8.P.2. A two-unit system with nonzero detection latency

7. * Suppose that we wish to perform state aggregation on the state diagram of problem 6 above and reduce it to the state diagram shown in Figure 8.P.3. (Thus states 2 and 2_U of Figure 8.P.2 are aggregated into state $2'$.) Derive expressions for the transition parameters $\lambda_1(t)$ and $\lambda_2(t)$. Note that the reduced chain is a nonhomogeneous CTMC.
8. Consider the duplex system of Example 8.36. Now solve for system unreliability $\pi_{SF}(t) + \pi_{UF}(t)$ and system unsafety $\pi_{UF}(t)$ using the convolution integration approach (see problem 6 of Section 8.1).

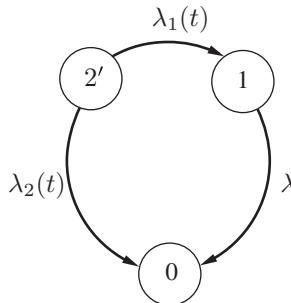


Figure 8.P.3. An aggregated version of Figure 8.P.2

9. Consider a two-component parallel redundant system with distinct failure rates λ_1 and λ_2 , respectively. On failure of a component, a repair process is invoked with the respective repair rates μ_1 and μ_2 . First assume that a failure of a component while another is being repaired causes a system failure. In this case set up the differential equations describing system behavior. Derive the reliability and the MTTF of the system.
10. Continuing with problem 9 above, assume that the failure of a component while another is being repaired is not catastrophic. Now compute the steady-state availability of the system.
11. Modify the reliability model of Example 8.37 to allow for a repair to occur from states $(3,0)$ and $(2,0)$ at a constant rate γ . State F is still assumed to be an absorbing state. Recompute the system MTTF.
12. Our assumption that the coverage probability is a given number is often unjustified in the modeling of fault-tolerant computers. In this problem we consider a “coverage model” of intermittent faults. (This is a simplified version of the model proposed by Stiffler [STIF 1980].) The model consists of five states as shown in Figure 8.P.4. In the active state A , the intermittent fault is capable of producing errors at the rate ρ and leading to the error state E . In the benign state B , the affected circuitry temporarily functions correctly. In state D the fault has been detected, and in the failure state F an undetected error has propagated so that we declare the system to have failed. Set up the differential equations for the five state probabilities. If we assume that all transition rates are greater than 0, then states A , B , and E are transient while states D and F are absorbing states. Given that the process starts in state A , it will eventually end up in either state D or state F . In the former case the fault is covered; in the latter it is not. We can, therefore, obtain an expression for coverage probability, $c = \lim_{t \rightarrow \infty} \pi_D(t)$. Using the final value theorem of Laplace transforms (see Appendix D), show that

$$c = \lim_{s \rightarrow 0} s\bar{\pi}_D(s) = \frac{\delta + \rho q}{\delta + \rho}.$$

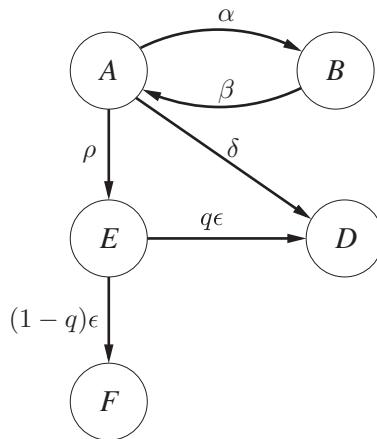


Figure 8.P.4. Stiffler's coverage model

Having obtained the value of c , we can then use an overall reliability model such as those in Examples 8.35 and 8.37. Such a decomposition is intuitively appealing, since the transition rates in the coverage model will be orders of magnitude larger than those in the overall reliability model. For a detailed study of this decomposition approach to reliability modeling, see the paper by Dugan and Trivedi [DUGA 1989].

13. Compute the absorption probabilities to state D and state F , starting from the initial state A at time 0 in problem 12 above using equation (8.131).
14. Modify the reliability model of Example 8.35 (Figure 8.59) to allow for repair from state 0 at a constant rate μ_1 . Now derive the expression for the steady-state availability of the system.
15. * Return to the concurrent program analyzed in problem 1, Section 8.4.2.1. First, derive the Laplace transform for the execution time for one iteration of the **repeat** statement using the methods of Section 8.5. Now, assume that the **repeat** clause in the program is changed so that it terminates when a Boolean expression B' is true. Assuming that the testing of this condition forms a sequence of Bernoulli trials with probability of success p , compute the mean and variance of the program execution time.

8.6 SOLUTION TECHNIQUES

In this section we describe different methods that can be used to obtain the solution to continuous-time Markov chains. Two kinds of solutions are of interest for a CTMC: transient and steady-state. The transient solution is obtained by solving the Kolmogorov differential equation (8.18), and the

steady-state solution is obtained by solving the linear system of equations (8.23). Closed-form analytical results are possible for either highly structured CTMCs (e.g., birth–death process) or very small CTMCs (e.g., Examples 8.19–8.27). In most other cases, we must resort to numerical solution techniques.

We divide this section into two subsections as follows. In the first subsection, we give steady-state solution methods for CTMCs and in the next subsection we discuss transient solution methods for CTMCs.

8.6.1 Methods for Steady-State Analysis

Direct methods such as Gaussian elimination are full-matrix methods and are memory-intensive. In practice, the Q matrix for practical CTMC problems is highly sparse. Hence sparse storage methods and sparsity preserving algorithms are needed. Iterative methods such as the power method, the Gauss–Seidel, and SOR belong to this class. Iterative methods require less storage and less computing resources. However, one major disadvantage of iterative methods is that they often require a very long time to converge to a desired solution. By contrast, for direct methods there is an upper bound on the time required to obtain the solution [STEW 1994]. Below we describe some commonly used iterative methods.

8.6.1.1 Power Method. The equation for steady-state probabilities (8.23) may be rewritten as

$$\boldsymbol{\pi} = \boldsymbol{\pi} \left(I + \frac{Q}{q} \right), \quad (8.132)$$

where $q \geq \max_i |q_{ii}|$. Note that

$$Q^* = I + \frac{Q}{q} \quad (8.133)$$

is a transition probability matrix of a DTMC (see problem 1 at the end of this section) and the above equation is the steady-state equation (7.18) for the DTMC. We can now set up an iteration by rewriting equation (8.132), such that

$$\boldsymbol{\pi}^{(i)} = \boldsymbol{\pi}^{(i-1)} Q^*, \quad (8.134)$$

where $\boldsymbol{\pi}^{(i)}$ is the value of $\boldsymbol{\pi}$ at the end of the i th step. We start off the iteration by initializing

$$\boldsymbol{\pi}^{(0)} = \boldsymbol{\pi}(0).$$

This method is referred to as the **power method**.

Example 8.44

Consider the two-state availability model of Figure 8.73. Assuming that the repair rate μ exceeds the failure rate λ , we choose $q = \max\{\mu, \lambda\} = \mu$. Then the matrix

$$Q^* = \begin{bmatrix} 1 - \rho & \rho \\ 1 & 0 \end{bmatrix},$$

where $\rho = \lambda/\mu$. If we let $\boldsymbol{\pi}^{(0)} = [1, 0]$, then, using Theorem 7.1, the result of the power method after n iterations will be

$$\boldsymbol{\pi}^{(n)} = \left[\frac{1 + \rho(-\rho)^n}{1 + \rho}, \frac{\rho - \rho(-\rho)^n}{1 + \rho} \right].$$

Therefore, as n approaches infinity, we will have

$$\boldsymbol{\pi} = \lim_{n \rightarrow \infty} \boldsymbol{\pi}^{(n)} = \left[\frac{1}{1 + \rho}, \frac{\rho}{1 + \rho} \right] = \left[\frac{\mu}{\lambda + \mu}, \frac{\lambda}{\lambda + \mu} \right].$$

‡

One difficulty that can arise in the use of the power method is that the Q^* matrix and the corresponding DTMC may be periodic. For instance, in Example 8.44, should $\lambda = \mu$, ρ will be 1, and hence the DTMC will be periodic, the power iteration (8.134) will not converge. In order to ensure convergence, we require that

$$q > \max_i |q_{ii}| \quad (8.135)$$

since this assures that the corresponding DTMC is aperiodic [GOYA 1987]. With this requirement, the power method is guaranteed to converge; however, the rate of convergence is found to be unacceptably low in practical problems. By contrast, the SOR method, to be discussed next, is found to be relatively fast in practice. Besides the advantage of guaranteed convergence, the power method can also be used on CTMCs with multiple absorbing states to obtain absorption probabilities.

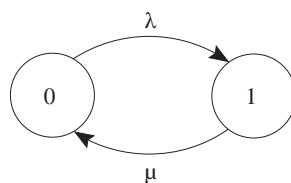


Figure 8.73. Two-state availability model of Example 8.44

Example 8.45

We return to the two-unit system DS of Example 8.41. The Q matrix is

$$Q = \begin{bmatrix} -2\lambda & 2\lambda c_{ds} & 0 & 2\lambda(1 - c_{ds}) \\ 0 & -\lambda & \lambda c_s & \lambda(1 - c_s) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Hence $q = 2\lambda$ and

$$Q^* = \begin{bmatrix} 0 & c_{ds} & 0 & 1 - c_{ds} \\ 0 & \frac{1}{2} & \frac{c_s}{2} & \frac{1-c_s}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The absorption probabilities can be obtained by applying equation (8.134). Applying the iteration to the current example, we have

$$\begin{aligned} \boldsymbol{\pi}^{(0)} &= [1, 0, 0, 0] \\ \boldsymbol{\pi}^{(1)} &= [0, c_{ds}, 0, 1 - c_{ds}] \\ \boldsymbol{\pi}^{(2)} &= \left[0, \frac{c_{ds}}{2}, \frac{c_{ds} \cdot c_s}{2}, 1 - \frac{c_{ds}}{2} - \frac{c_{ds} \cdot c_s}{2}\right] \\ \boldsymbol{\pi}^{(3)} &= \left[0, \frac{c_{ds}}{4}, \frac{3}{4}c_{ds} \cdot c_s, 1 - \frac{c_{ds}}{4} - \frac{3}{4}c_{ds} \cdot c_s\right] \\ &\vdots \quad \vdots \\ \boldsymbol{\pi}^{(k)} &= \left[0, \frac{c_{ds}}{2^{k-1}}, c_{ds} \cdot c_s \left(1 - \frac{1}{2^{k-1}}\right), 1 - \frac{c_{ds}}{2^{k-1}} - c_{ds} \cdot c_s \left(1 - \frac{1}{2^{k-1}}\right)\right]. \end{aligned}$$

At convergence, we have

$$\boldsymbol{\pi} = \lim_{k \rightarrow \infty} \boldsymbol{\pi}^{(k)} = [0, 0, c_{ds} \cdot c_s, 1 - c_{ds} \cdot c_s].$$

¶

8.6.1.2 Successive Overrelaxation (SOR). Iterative methods such as Gauss–Seidel or Successive overrelaxation (SOR) are preferable to direct methods (such as Gaussian elimination) to carry out steady-state analysis. The SOR method starts with an initial guess $\boldsymbol{\pi}^{(0)}$ and iterates using the following formula until some criteria for convergence are satisfied:

$$\boldsymbol{\pi}^{(k+1)} = \omega[\boldsymbol{\pi}^{(k+1)}U + \boldsymbol{\pi}^{(k)}L]D^{-1} + (1 - \omega)\boldsymbol{\pi}^{(k)}, \quad (8.136)$$

where $\boldsymbol{\pi}^{(k)}$ is the solution vector at the k th iteration, L is a lower triangular matrix, U is an upper triangular matrix, and D is a diagonal matrix such

that $Q = D - L - U$. Gauss–Seidel is a special case of SOR with $\omega = 1$. The choice of ω is discussed elsewhere [CIAR 1993]. Although in most problems SOR converges much faster than the power method, one can come up with fairly simple CTMCs on which SOR will diverge.

Example 8.46

Consider the four-state CTMC shown in Figure 8.74a with infinitesimal generator matrix

$$Q = \begin{bmatrix} -2.5 & 2.5 & 0 & 0 \\ 0 & -1.2 & 0 & 1.2 \\ 3.0 & 0 & -3.0 & 0 \\ 0 & 0 & 2.0 & -2.0 \end{bmatrix}.$$

This CTMC will not converge using the Gauss–Seidel or SOR iteration based on equation (8.136). However, when we use underrelaxation rather than overrelaxation, that is, $0 < \omega < 1$, the iteration will converge. Renumbering the states can also cause the iteration to converge. In Figure 8.74b, the swapping of state 2 and state 3 changes the infinitesimal generator matrix to

$$Q_1 = \begin{bmatrix} -2.5 & 2.5 & 0 & 0 \\ 0 & -1.2 & 1.2 & 0 \\ 0 & 0 & -2.0 & 2.0 \\ 3.0 & 0 & 0 & -3.0 \end{bmatrix}.$$

With this reordered CTMC, both SOR and Gauss–Seidel iterations will converge. ‡

Renumbering the states of the CTMC, switching to Gauss–Seidel, using underrelaxation (rather than overrelaxation), or switching to the power method are used as alternative solution methods in cases of diverging SOR as in the example above. Further discussion of numerical methods is available in the literature [BOLC 1998, STEW 1994, GRAS 2000].

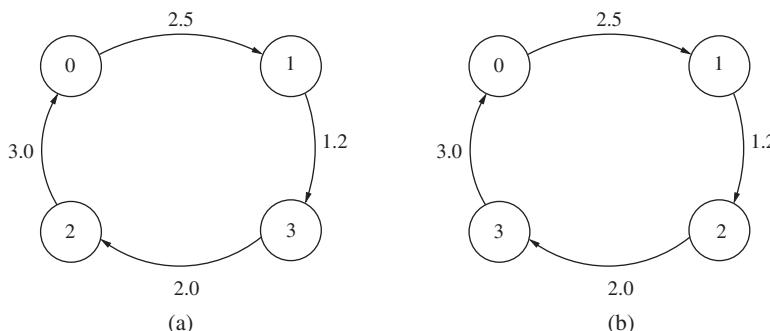


Figure 8.74. (a) A nonconvergent CTMC; (b) reordered convergent CTMC

Problems

1. Show that the matrix Q^* defined by equation (8.133) is a DTMC matrix.
2. Show that with condition (8.135), Q^* is an aperiodic matrix.
3. Perform steady-state analysis of the CTMC model in Example 8.23 (Figure 8.32) with the power method and then with the SOR method.
4. Apply the power method to the CTMC of Figure 8.P.4 to obtain an expression for the probability of absorption to state D .
5. Suppose that we are interested in computing the derivative $d\pi/d\lambda$ with respect to some parameter λ of the Q matrix. Derive an equation for computing this sensitivity vector [BLAK 1988].
6. * Show that the Gauss–Seidel iteration matrix for the CTMC of Figure 8.74a has eigenvalues $\{1, -1, 0, 0\}$ and that the SOR iteration matrix for $\omega > 1$ also has one eigenvalue whose magnitude is larger than 1. Finally, show that the iteration matrix with $\omega < 1$ has only one eigenvalue equal to 1 and the remaining eigenvalues are all less than 1 in magnitude.

8.6.2 Methods for Transient Analysis

8.6.2.1 Fully Symbolic Method. Taking the Laplace transform on both sides of the Kolmogorov differential equation (8.18), we have

$$s\bar{\pi}(s) - \pi(0) = \bar{\pi}(s)Q.$$

Rearranging the terms, we have

$$\bar{\pi}(s) = \pi(0)(sI - Q)^{-1}, \quad (8.137)$$

where I is the identity matrix. The transient state probability is obtained by computing the inverse Laplace transform of $\bar{\pi}(s)$. We have already used this approach in Sections 8.3 and 8.5.

The main advantage of the symbolic method is that the solution obtained will be closed-form and fully symbolic in both the system parameters and time t . However, in general, computing the inverse Laplace transform is extremely difficult, except for simple Markov chains. If we assume that entries in the Q matrix are all numerical but require the final solution $\pi(t)$ to be a symbolic function in time parameter t , we will have a semisymbolic (or seminumerical) solution. The semisymbolic method is simpler than the fully symbolic method and has been implemented in the SHARPE software package [RAME 1995, SAHN 1996]. In practice, the applicability of the semisymbolic method is limited since it requires full matrix storage, is computationally expensive, and is often numerically unstable. It is for these reasons that numerical solution methods are commonly used and implemented in software packages such as SHARPE and SPNP. We will next discuss numerical solution methods.

8.6.2.2 Numerical Methods. The Kolmogorov differential equation (8.18) is a set of ordinary differential equations (ODEs). Standard techniques for solving ODEs are discretization. These methods discretize the time interval into a finite number of subintervals and compute the solution step by step. Discretization methods can be classified into two categories: explicit and implicit. Implicit methods are superior to explicit methods because of their more stable nature in determining the step size and achieving high solution accuracy. For stiff Markov chains implicit methods, such as TR-BDF2 and the implicit Runge–Kutta method, are commonly used in seeking solutions [REIB 1988]. For nonstiff Markov chains, explicit methods such as Runge–Kutta work well.

ODE solvers are efficient in obtaining the evolution profile of the state probabilities. However, they may require a very small value of h to obtain accurate results. The most common method for transient analysis is the randomization method, which is also known as the **uniformization method** or **Jensen's method** [REIB 1988]. This method begins with the formal solution of equation (8.18) given by (see problem 2 of Section 8.1)

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)e^{Qt}, \quad (8.138)$$

where the matrix exponential is defined by the infinite series:

$$e^{Qt} = \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!}. \quad (8.139)$$

There are three practical problems in using this approach directly: (1) Q has both negative and positive entries and hence the preceding computation has both additions and subtractions (such computation will have poor numerical behavior); (2) raising the matrix Q to its powers is both costly and fills in zeros in the matrix—recall that in practice, Q will be very large yet sparse; and (3) the infinite series shown above will need to be truncated.

We solve the first problem using the following transformation. We use an integrating factor e^{qt} with $q \geq \max_i |q_{ii}|$ in equation (8.18), so that we let $\mathbf{y}(t) = e^{qt}\boldsymbol{\pi}(t)$. Then the Kolmogorov differential equation (8.18) can be transformed into

$$\frac{d\mathbf{y}}{dt} = qe^{qt}\boldsymbol{\pi}(t) + e^{qt}\frac{d\boldsymbol{\pi}}{dt} = \mathbf{y}(t)Q^*q$$

where $Q^* = Q/q + I$. So we have

$$\mathbf{y}(t) = e^{qt}\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)e^{Q^*qt}.$$

Hence

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0) \sum_{k=0}^{\infty} e^{-qt} \frac{(qt)^k}{k!} (Q^*)^k. \quad (8.140)$$

Matrix Q^* is a DTMC matrix, and hence it has no negative entries. This avoids the numerical problems as no subtractions are involved.

Example 8.47

Consider again the two-state homogeneous CTMC of Figure 8.73. The Kolmogorov differential equations become

$$\begin{aligned}\frac{d\pi_0(t)}{dt} &= -\lambda\pi_0(t) + \mu\pi_1(t), \\ \frac{d\pi_1(t)}{dt} &= \lambda\pi_0(t) - \mu\pi_1(t).\end{aligned}\tag{8.141}$$

Applying the fact that $\pi_0(t) + \pi_1(t) = 1$, we can rewrite equation (8.141) as

$$\frac{d\pi_0(t)}{dt} + (\mu + \lambda)\pi_0(t) = \mu.\tag{8.142}$$

This is a linear differential equation of first order. Assume that $\mu > \lambda$, and let $q = \mu$. Now define

$$y_0(t) = e^{\mu t}\pi_0(t).$$

By differentiating both sides we get

$$\frac{dy_0(t)}{dt} = \mu e^{\mu t}\pi_0(t) + e^{\mu t}\frac{d\pi_0(t)}{dt}.$$

From (8.142), we have

$$\begin{aligned}\frac{dy_0}{dt} &= \mu e^{\mu t} - \lambda e^{\mu t}\pi_0(t) \\ &= -\lambda y_0(t) + \mu e^{\mu t}.\end{aligned}$$

By solving the equation, we get

$$y_0(t) = y_0(0)e^{-\lambda t} + \frac{\mu}{\mu + \lambda}e^{\mu t}.$$

So we have

$$\begin{aligned}\pi_0(t) &= e^{-\mu t}y_0(t) \\ &= y_0(0)e^{-(\lambda+\mu)t} + \frac{\mu}{\mu + \lambda}.\end{aligned}$$

To determine the constant $y_0(0)$, we use the initial condition $\pi_0(0) = 1$. We have

$$1 = \pi_0(0) = \frac{\mu}{\mu + \lambda} + y_0(0), \quad y_0(0) = 1 - \frac{\mu}{\mu + \lambda} = \frac{\lambda}{\mu + \lambda}.$$

So we have

$$\pi_0(t) = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} e^{-(\mu+\lambda)t}.$$

In a similar way, we also have

$$\pi_1(t) = \frac{\lambda}{\mu + \lambda} - \frac{\lambda}{\mu + \lambda} e^{-(\mu+\lambda)t}.$$

#

To solve the second problem, we rewrite equation (8.140) as

$$\boldsymbol{\pi}(t) = \sum_{k=0}^{\infty} \boldsymbol{\theta}(k) e^{-qt} \frac{(qt)^k}{k!} \quad (8.143)$$

where $\boldsymbol{\theta}(0) = \boldsymbol{\pi}(0)$ and

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1)Q^*, \quad k = 1, 2, \dots \quad (8.144)$$

The computation of $\boldsymbol{\theta}(k), k = 1, 2, \dots$ avoids the problem of raising matrix Q to its powers. The term $\boldsymbol{\theta}(k)$ in (8.143) can be interpreted as the k th step state probability vector of a DTMC with transition probability matrix Q^* , while the term $e^{-qt}(qt)^k/k!$ is the Poisson pmf with parameter qt . Thus, the randomization method expresses the state probabilities of a CTMC in terms of the sum of the DTMC state probabilities of a series of steps weighted by a Poisson pmf.

Given a precision requirement (a truncation error tolerance ϵ), the infinite series can be left-/right-truncated (to solve the third problem):

$$\boldsymbol{\pi}(t) \approx \sum_{k=l}^r \boldsymbol{\theta}(k) e^{-qt} \frac{(qt)^k}{k!}. \quad (8.145)$$

The values of l and r can be determined from the specified truncation error tolerance ϵ by

$$\sum_{k=0}^{l-1} e^{-qt} \frac{(qt)^k}{k!} \leq \frac{\epsilon}{2}, \quad 1 - \sum_{k=0}^r e^{-qt} \frac{(qt)^k}{k!} \leq \frac{\epsilon}{2}.$$

For stiff Markov chains, qt is typically very large and the term e^{-qt} almost always runs into *underflow problems*. To avoid underflow, we use the method of Fox and Glynn [FOXG 1988] to compute l and r , the left and right truncation points. This method also computes the Poisson probabilities $e^{-qt}(qt)^k/k!$ for all $k = l, l+1, \dots, r-1, r$.

Note that the computational complexity of the randomization method rises linearly with q and t . A large value of qt also implies a large number of matrix

vector multiplications, which results in large roundoff errors. CTMCs with a large value of qt are hence said to be stiff. Observe that in equation (8.144), that is used to compute the probability vectors for the underlying DTMC, the DTMC matrix Q^* is identical to that used in the power method for computing the steady-state vector for the CTMC (see Section 8.6.1.1). If the convergence of the power iteration occurs before step r , we can terminate the iteration in equation (8.144) on attaining steady state. Recall that, in order to ensure convergence of the power iteration, we require

$$q > \max_k |q_{kk}|$$

since this assures that the DTMC described by Q^* is aperiodic.

Assume that convergence has been achieved at the S th iteration by observing the sequence $\boldsymbol{\theta}(k)$. Three cases arise [MUPP 1994]:

1. ($S > r$): In this case the steady-state detection has no effect and the probability vector is calculated using (8.145).
2. ($l < S \leq r$): Change equation (8.145) to

$$\boldsymbol{\pi}(t) \approx \sum_{k=l}^S \boldsymbol{\theta}(k) e^{-qt} \frac{(qt)^k}{k!} + \boldsymbol{\theta}(S) \left(1 - \sum_{k=0}^S e^{-qt} \frac{(qt)^k}{k!} \right).$$

3. ($S \leq l$): The DTMC reaches steady state before the left truncation point. In this case, no additional computation is necessary and $\boldsymbol{\pi}(t)$ is set to $\boldsymbol{\theta}(S)$.

Steady-state detection is used in the implementation of uniformization in SHARPE and SPNP software packages.

The computation of the cumulative probability vector $\mathbf{L}(t) = \int_0^t \boldsymbol{\pi}(u) du$ is similar to that of $\boldsymbol{\pi}(t)$ [CIAR 1993]. Integrating equation (8.143) with respect to t yields

$$\mathbf{L}(t) = \frac{1}{q} \sum_{k=0}^{\infty} \boldsymbol{\theta}(k) \left(1 - \sum_{j=0}^k e^{-qt} \frac{(qt)^j}{j!} \right). \quad (8.146)$$

This is again a summation of an infinite series that can be evaluated up to the first r significant terms resulting in

$$\mathbf{L}(t) \approx \frac{1}{q} \sum_{k=0}^r \boldsymbol{\theta}(k) \left(1 - \sum_{j=0}^k e^{-qt} \frac{(qt)^j}{j!} \right). \quad (8.147)$$

Given an error tolerance ϵ , the number of terms needed can be computed by

$$t \sum_{i=r}^{\infty} e^{-qt} \frac{(qt)^i}{i!} - \left(\frac{r+1}{q} \right) \sum_{i=r+1}^{\infty} e^{-qt} \frac{(qt)^i}{i!} < \epsilon. \quad (8.148)$$

The detection of steady state for the underlying DTMC also applies to equation (8.147). Two cases arise:

1. ($S > r$): Steady-state detection does not take place and $\mathbf{L}(t)$ is computed using equation (8.147).
2. ($S \leq r$): Equation (8.147) is modified as follows:

$$\begin{aligned}\mathbf{L}(t) \approx & \frac{1}{q} \sum_{k=0}^S \boldsymbol{\theta}(k) \left(1 - \sum_{j=0}^k e^{-qt} \frac{(qt)^j}{j!} \right) \\ & + \frac{1}{q} \boldsymbol{\theta}(S) \left(qt - \sum_{k=0}^S \left(1 - \sum_{j=0}^k e^{-qt} \frac{(qt)^j}{j!} \right) \right).\end{aligned}$$

Nonhomogeneous CTMCs, are useful for reliability models and to model performance of practical systems such as computer networks. Rindos et al. [RIND 1995] discussed a class of NHCTMCs that can be transformed into a homogeneous CTMC and then be solved by the techniques discussed above. For general NHCTMCs, solution techniques include the ODE solver, randomization [DIJK 1992], and time stepping. The ODE solver has been used successfully in obtaining numerical solution of NHCTMCs by discretization. Randomization of NHCTMC, as an extension of randomization of homogeneous CTMC, can provide numerical solutions, but the drawback is its complexity. The time-stepping method is simple and efficient, so it is frequently used in automated computation of NHCTMC. The basic idea is dividing the time axis into small intervals, and approximating the generator matrix to be time-independent within each interval [RAMA 2000]. Further studies of numerical solution techniques are available in the literature [GRAS 2000, MUPP 1992b, MUPP 1994, REIB 1988, REIB 1989, STEW 1994].

Problems

1. Consider an $M/M/\infty$ queuing system (refer to problem 4 in Section 8.2.2) and obtain the transient solution using the Laplace transform method.
2. Consider an $M/M/1/2$ queuing system and find the transient state probabilities using the Laplace transform method.
3. Find the transient state probabilities for Example 8.19 using the Laplace transform technique. Also find the instantaneous availability for this system and compare with that of Example 8.6. Show that they are not equal even though the steady-state availabilities are the same.
4. Compute the transient solution for Example 8.35 numerically using randomization.

8.7 AUTOMATED GENERATION

When we apply Markov chains to analyze the reliability, availability, and performance of a system, the primary procedure consists of the following steps: abstracting the physical system at first, constructing the Markov chain, and then setting up ordinary differential equations (8.18) (for transient solution) or linear equations (8.23) (for steady-state solution) manually, and finally writing a program for the numerical solution to the equations. It is a rather tedious and error-prone procedure, especially when the number of states becomes very large. There have been some efforts to develop software packages to solve Markov chains automatically using the numerical or seminumerical methods described in the previous section, while still requiring construction of the Markov chain by hand. Also note that the Markov model of a system is sometimes far removed from the shape and general feel of the system being modeled. System designers may have difficulty in directly translating their problems into a Markov chain. Since late 1980s, some researchers have been developing new modeling formalisms and software packages for the automated generation and solution of Markovian stochastic systems. The effort has led to the emergence of a popular formalism called **stochastic Petri nets** (SPNs), which is more concise in its specification and whose form is closer to a designer's intuition about what a model should look like. Some software packages such as SPNP [CIAR 1993], DSPNexpress [LIND 1998], GreatSPN [CHIO 1995], and SHARPE [SAHN 1996] are available, which can translate the SPN model into CTMC and then solve it automatically. These automated tools free system analysts from the painstaking construction and solution of Markov chains by hand and enable them to focus on the task of translating the dynamic behavior of the system into an SPN model. We begin by first describing ordinary Petri nets. Subsequently, we discuss various kinds of stochastic Petri nets and their applications.

8.7.1 Petri Nets

Petri nets were originally introduced by C. A. Petri in 1962. Formally, a Petri net (PN) is a 5-tuple $\text{PN} = (P, T, A, M, \mu_0)$, where

- $P = \{p_1, p_2, \dots, p_m\}$ is a finite set of *places* (drawn as circles).
- $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of *transitions* (drawn as bars).
- $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs connecting P and T .
- $M : A \rightarrow \{1, 2, 3, \dots\}$ is the multiplicity associated with the arcs in A .
- $\mu : P \rightarrow \{0, 1, 2, \dots\}$ is the *marking* that denotes the number of tokens (drawn as black dots or a positive integer) for each place in P . The initial marking is denoted as μ_0 .

Graphically, Petri net is a directed graph with two disjoint types of nodes: **places** and **transitions**. A directed arc connecting a place (transition) to a transition (place) is called an **input** (resp. **output**) **arc** of the transition. A positive integer called **multiplicity** can be associated with each arc. Places connected to a transition by input arcs are called the **input places** of this transition, and those connected by means of output arcs are called its **output places**.

Each place may contain zero or more tokens in a **marking**. Marking represents the state of the model at a particular instant. This concept is central to PNs. The notation $\#(p, \mu)$ is used to indicate the number of tokens in place p in marking μ . If the marking is clear from the context, the notation $\#p$ is used.

A transition is **enabled** when each of its input places has at least as many tokens as the multiplicity of the corresponding input arc. A transition may **fire** when it is enabled, and on firing, a number of tokens equal to the multiplicity of the input arc are removed from each of the input places, and a number of tokens equal to the multiplicity of the output arc are deposited in each of its output places. The sequencing of firing is an important issue in PNs. If two transitions are enabled in a PN marking, they cannot be fired “at the same time”: a choice must be made concerning which one to fire first, the other can only fire after that, if it is still enabled. The firing of a transition may transform a PN from one marking into another. With respect to a given initial marking μ_0 , the **reachability set** is defined as the set of all markings reachable through any possible firing sequences of transitions, starting from the initial marking. The evolution of a PN can be completely described by its **reachability graph**, in which each marking in the reachability set is a node in the graph, while the arcs describe the possible marking-to-marking transitions [CIAR 1993, MURA 1989]. Arcs are labeled with the name of the transition whose firing caused the associated changes in the marking.

Example 8.48 [BOLC 1998]

Consider a simple example of PN, shown in Figure 8.75. Part (a) shows the initial marking denoted by the vector $(2, 0, 0, 0)$, where only transition t_1 is enabled because place P_1 contains two tokens, and t_2 is disabled because P_2 is empty. When t_1 fires, one token is removed from its input place P_1 and one token is deposited in both its output places P_2 and P_3 [see the marking shown in part (b)]. In part (b), both transitions t_1 and t_2 are enabled. If t_1 fires first, the PN will reach a marking as shown in (d), while the marking (c) will be reached if t_2 fires first.

The reachability set in this example is given by $\{(2,0,0,0), (1,1,1,0), (0,0,1,1), (0,2,2,0)\}$. Figure 8.76 depicts the reachability graph of this PN.

#

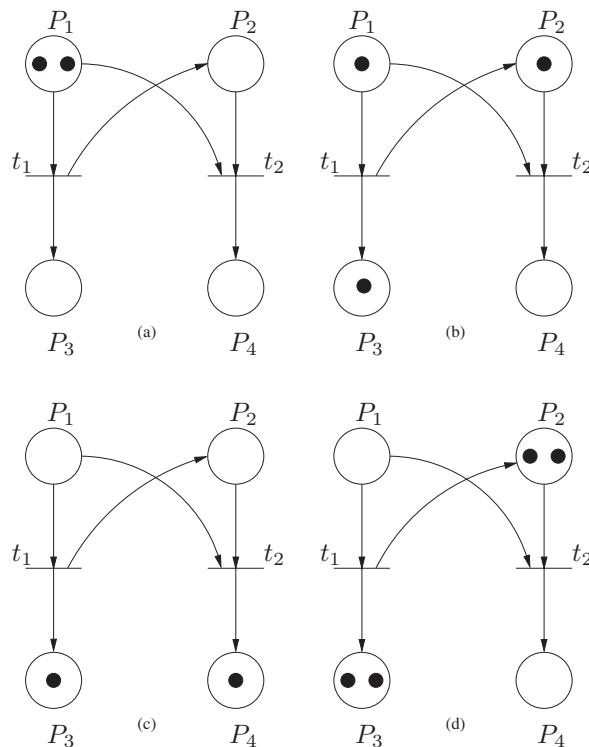


Figure 8.75. An example of a Petri net

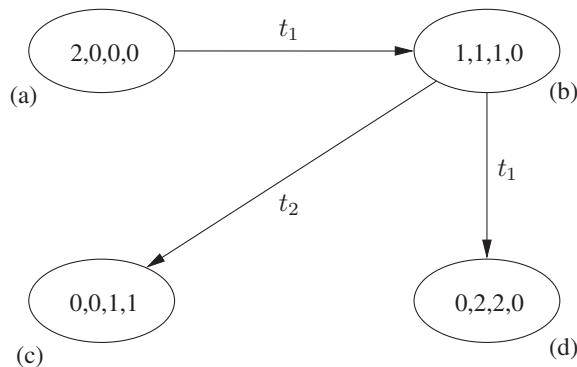


Figure 8.76. Reachability graph of the Petri net for Example 8.48

PN can be used to capture the behavior of many real-world situations, including sequencing, synchronization, concurrency, and conflict. In computer networks, they have been used to describe and verify the communication protocols [LAIJ 1998]. However, the concept of time is not explicitly given in the original definition of Petri nets, while for the performance and availability analysis of dynamical systems, it is necessary and useful to introduce time delays associated with transitions in the Petri net models. This intuition has led to the emergence of stochastic Petri nets.

8.7.2 Stochastic Petri Nets

Stochastic Petri nets are obtained by associating stochastic and timing information to Petri nets. We do this by attaching **firing time** to each transition, representing the time that must elapse from the instant that the transition is enabled until the instant it actually fires in isolation, that is, assuming that it is not affected by the firing of other transitions. If two or more transitions are enabled at the same time, the firing of transitions is determined by the **race policy**; that is, the transition whose firing time elapses first is chosen to fire next. If the firing times can have general distributions, SPN can be used to represent a wide range of well-known stochastic processes. However, choices about execution policy and memory policy, besides the firing time distributions, must be specified (see Ciardo et al. [CIAR 1993] and Choi et al. [CHOI 1994] for details). The firing times are often restricted to have an exponential distribution to avoid policy choices. A more important fact in this case, though, is that an SPN can be automatically transformed into a CTMC. In a graphical representation, transitions with exponentially distributed firing times are drawn as rectangular boxes.

When SPN is applied to performance analysis of computer networks, *places* can be used to denote the number of packets or cells in the buffer or the number of active users, or flows in the system, while the arrival and departure of packets, cells, users or flows can be represented by firing of *transitions*.

In the following, we present SPN models for simple queuing systems to illustrate the transformation from SPN into CTMC.

Example 8.49 (Poisson Process)

Consider an SPN model of a Poisson process as shown in Figure 8.77. The number of tokens in P_{queue} represents the number of customers that have arrived. Here, the customers arrive according to a Poisson stream with rate λ . This is captured through the transition T_{arrival} . Note that since transition T_{arrival} has no input arcs, it is always enabled. Furthermore, its firing time distribution is $\text{EXP}(\lambda)$.

Following the firing sequences, we can get the reachability graph (RG) of the SPN manually, then produce the underlying CTMC. Algorithms are available to explore the reachability graph in a recursive manner, in which the set of all possible markings is exhaustively sought by starting from the initial marking [CIAR 1993]. These algorithms have been implemented in automated SPN tools such as SHARPE [SAHN 1996] and SPNP [CIAR 1993].

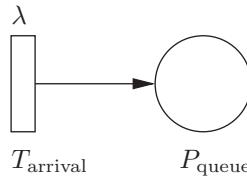


Figure 8.77. SPN model of a Poisson process

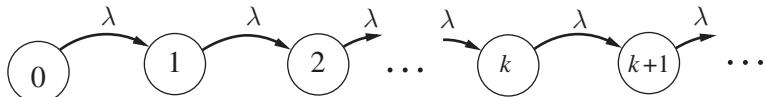


Figure 8.78. Reachability Graph of the Poisson process

We illustrate the reachability graph in Figure 8.78. A node in the RG is a marking of the SPN, represented by a circle labeled by the marking $\#P_{\text{queue}}$ inside. The arcs in the RG correspond to the firing of the transitions. The corresponding transition rates are labeled along with the arcs. This RG is the same as the state diagram of the Poisson process shown in Figure 8.21.

#

Example 8.50 ($M/M/1$ Queue)

In this example, we consider an SPN model of an $M/M/1$ queue as shown in Figure 8.79. This model can be seen as an extension of the previous model by including the transition T_{service} . The number of tokens in P_{queue} represents the number of customers in the system (including the one receiving service, if any). Whenever there is a customer (one or more tokens) in the system (in place P_{queue}), a customer may complete service when the transition T_{service} fires and the firing time is exponentially distributed with rate μ .

From the RG in Figure 8.80, we can easily recognize this as a simple birth–death process with an infinite state space (see Section 8.2.1). Measures such as system size, response time, and throughput can be computed by solving this CTMC.

#

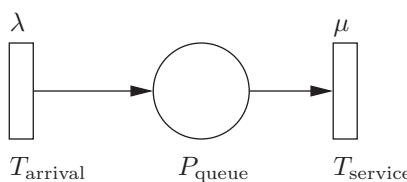


Figure 8.79. SPN model of $M/M/1$ queue

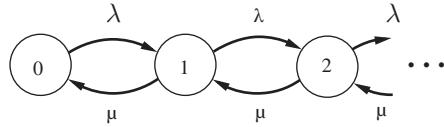


Figure 8.80. Reachability graph of the SPN model of $M/M/1$ queue

Example 8.51 ($M/M/1/n$ Queue)

An SPN model of an $M/M/1/n$ queue (Example 8.5) is shown in Figure 8.81. Comparing with the earlier example, place P_{queue} represents the number in the system while P_{vacancy} represents the vacancies in the buffer. Since the firing time of a timed transition is exponentially distributed, transition T_{arrival} is labeled λ . The finiteness of the queue is specified with the initial number of tokens in P_{vacancy} , which is equal to n . If $\#P_{\text{vacancy}} > 0$, that is, if the buffer is not full, the transition T_{arrival} is enabled. The firing of transition T_{arrival} represents a customer entering the queue. When the buffer is full, $\#P_{\text{vacancy}} = 0$, the transition T_{arrival} is disabled and any arriving customer is rejected. The firing of transition T_{service} represents the departure of a customer; the firing rate of the transition is μ . On its firing, one token (customer) will be removed from place P_{queue} through the input arc from place P_{queue} to T_{service} , while one token is deposited (one vacancy created) into place P_{vacancy} .

Now, we illustrate the reachability graph in Figure 8.82. A node in the RG is a marking of the SPN, represented by an oval labeled by the marking $(\#P_{\text{queue}}, \#P_{\text{vacancy}})$ as in earlier examples.

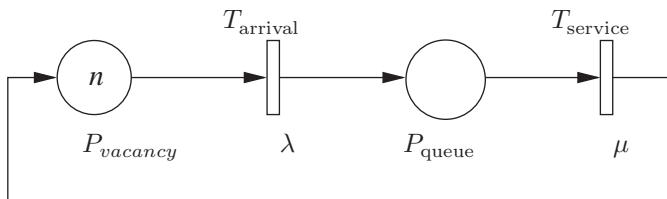


Figure 8.81. SPN model of $M/M/1/n$ queue

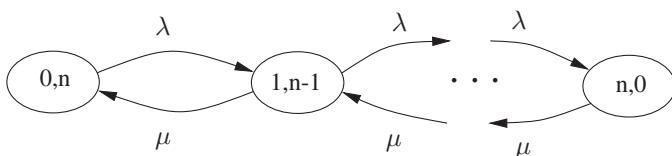


Figure 8.82. Reachability graph of the SPN model of $M/M/1/n$ queue

The reachability graph shown in Figure 8.82 can be easily recognized as a finite birth–death process. Except for the state labels, the reachability graph is the same as the CTMC state diagram of Figure 8.9. Subsequently all measures of interest can be obtained by solving this CTMC.

‡

Problems

1. Extend the SPN of the $M/M/1$ queue to introduce server failures and repairs. Assume that server failure rate is γ and the repair rate is τ . Draw the reachability graph.

8.7.3 Generalized Stochastic Petri Nets

In generalized stochastic Petri nets (GSPNs) [AJMO 1995], transitions are allowed to be either **timed** (exponentially distributed firing time, drawn as rectangular boxes) or **immediate** (zero firing time, represented by thin black bars). If both an immediate transition and a timed transition are enabled at the same instant, the immediate transition fires first. If several immediate transitions compete for firing, **firing probabilities**, usually specified as **weights** to be normalized, should be specified to resolve these conflicts.

Other extensions in GSPN include inhibitor arcs and transition priorities. **Inhibitor arcs** have small hollow circles instead of arrows at their terminating ends. A transition with an inhibitor arc cannot fire if the number of tokens that the input place of the inhibitor arc contains is equal to or more tokens than the multiplicity of the arc. **Transition priorities** are defined by assigning an integer priority level to each transition, which adds the constraint that a transition may be enabled in a marking only if no higher priority transition is enabled.

A marking of a GSPN is called **vanishing** if at least one immediate transition is enabled in the marking and **tangible** otherwise. It has been proved that exactly one CTMC corresponds to a given GSPN under the condition that only a finite number of transitions can fire in finite time with nonzero probability [AJMO 1995]. For finite reachability sets, the exception can only occur if a **vanishing loop** exists. This case is of little practical interest and is usually treated as a modeling error.

The GSPN analysis can be decomposed into four steps [CIAR 1993]:

- Generating the extended reachability graph, which contains the markings of the reachability set as nodes and some stochastic information attached to the arcs; thus all the markings are related to each other with stochastic information.
- Eliminating the vanishing markings with zero sojourn times and the corresponding transitions from the extended reachability graph. This procedure generates a homogeneous CTMC.

- Analyzing the steady-state, transient, or cumulative behavior of the CTMC.
- Determining the measures, such as the average number of tokens in a place and the throughput of a timed transition.

Now, we illustrate some examples of GSPN modeling of simple queuing systems.

Example 8.52 ($M/E_m/1/n + 1$ Queue)

Consider a GSPN model of an $M/E_m/1/n + 1$ queuing system shown in Figure 8.83 where the customers arrive according to a Poisson process and the service times are m -stage Erlang (E_m). Transition T_{arrival} with firing rate λ represents the arrival of a customer. An inhibitor arc with multiplicity n from place P_{queue} to T_{arrival} represents the capacity of the queue. The transition T_{arrival} is disabled when the number of tokens in place P_{queue} equals n . Note that one job will be in service in such a marking and hence the number of jobs in the system will be equal to $n + 1$. The immediate transition t_{quick} fires when place P_{queue} has one token and P_{service} is empty. m tokens will be deposited in place P_{service} after the firing of t_{quick} . The firing of transition T_{service} represents the completion of one stage of the m -stage Erlang distributed service time. Since the mean service time is to be $1/\mu$, the firing rate of transition T_{service} is $m\mu$.

Figure 8.84 shows the extended reachability graph of the GSPN model, where (i, j) represents the number of tokens in place P_{queue} and P_{service} . After the extended reachability graph is generated, the vanishing markings are eliminated, resulting in a CTMC (Figure 8.85). Then the CTMC can be solved using numerical methods discussed in Section 8.6. In this example, there are n vanishing markings, which are represented by dashed circles [i.e., $(1, 0), (2, 0), \dots, (n, 0)$].

‡

Example 8.53 ($M/M/i/n$ Queue)

Consider a GSPN model of an $M/M/i/n$ queuing system as shown in Figure 8.86. An inhibitor arc with multiplicity $n - i$ from place P_{queue} to T_{arrival} represents the maximum capacity of waiting customers in the system. The immediate transition t_{quick} will fire only if $\#P_{\text{server}} > 0$ and $\#P_{\text{queue}} > 0$ and upon its firing one token will be deposited in place P_{service} . Note that the symbol “#” is added next to arc from place P_{service} to transition T_{service} , which indicates that the firing rate of transition T_{service}

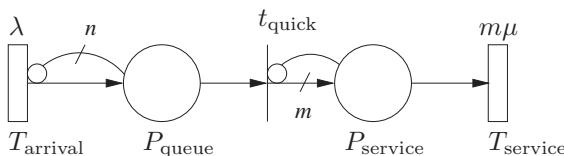


Figure 8.83. A GSPN model of $M/E_m/1/n + 1$ queue

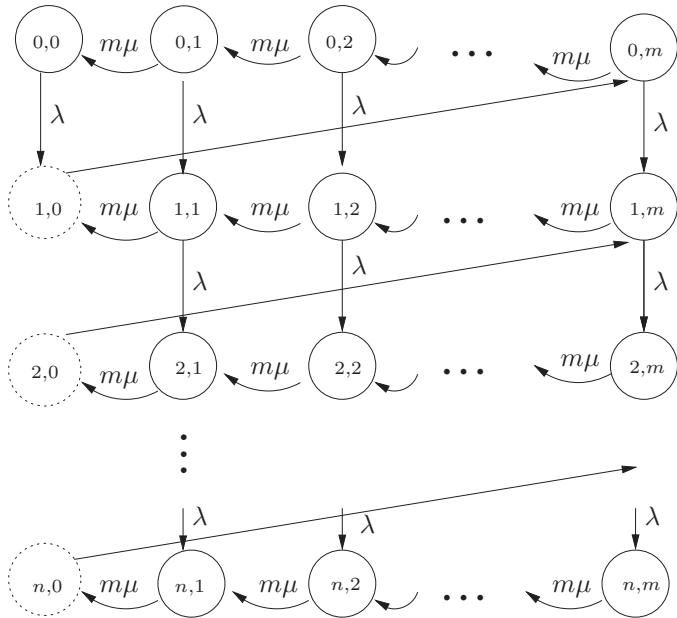


Figure 8.84. Extended reachability graph of the GSPN model of $M/E_m\backslash 1\backslash n+1$ queue

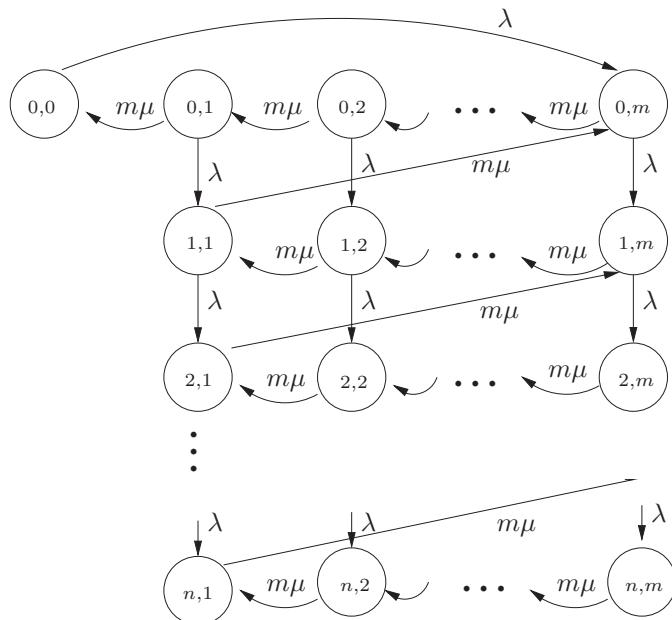


Figure 8.85. CTMC derived from the GSPN model of $M/E_m/1/n + 1$ queue

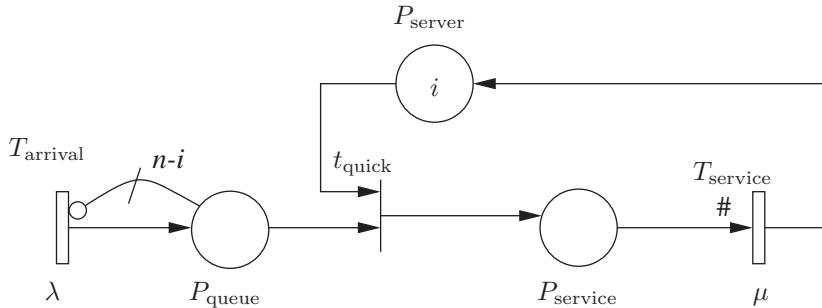


Figure 8.86. A GSPN model of $M/M/i/n$ queue

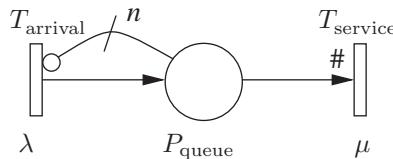


Figure 8.87. A GSPN model of $M/M/n/n$ queue

is marking-dependent, specifically, $\#P_{\text{service}} \cdot \mu$. On firing, one token will be put back in place P_{server} . Note that, in this GSPN model, the combined value of $\#P_{\text{queue}}$ and $\#P_{\text{service}}$ will give the number of customers in the system. The corresponding RG of this model can be easily drawn by the triple $(\#P_{\text{queue}}, \#P_{\text{service}}, \#P_{\text{server}})$. Measures of interest can be obtained from the corresponding CTMC.

‡

Example 8.54 ($M/M/n/n$ Queue)

Now let us revisit the $M/M/n/n$ queue, also known as the Erlang loss model (see problem 1 in Section 8.2.2) but now use the GSPN paradigm. The Erlang loss model is a special case of the $M/M/i/n$ queue for which $i = n$. Let us still use λ and μ as the arrival rate and the service rate, respectively. We depict the GSPN model in Figure 8.87.

Because of the existence of multiple servers, the service rate of the $M/M/n/n$ queue is the product of the number of customers in queue and the service rate, μ . Also notice that the multiplicity of the inhibitor arc from P_{queue} to T_{arrival} is n corresponding to the system capacity. The reachability graph is shown in Figure 8.88.

‡

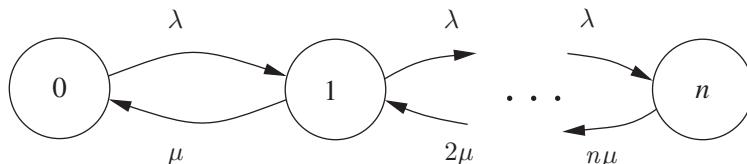


Figure 8.88. The reachability graph of Example 8.54 ($M/M/n/n$)

8.7.4 Stochastic Reward Nets

Stochastic reward nets (SRNs) are based on GSPN but extend it further [CIAR 1993]. In SRN, every tangible marking can be associated with a reward rate. It can be shown that an SRN can be mapped into a Markov reward model. Thus a variety of reward-based measures can be specified and calculated using a very convenient formalism. SRN also allows several other features that make specification more convenient:

- Each transition may have a **guard** (also called an **enabling function**) that is **marking-dependent**. A transition is enabled in a marking only if its guard (a Boolean condition) is satisfied, in addition to the constraints imposed by priority, input arcs, and inhibitor arcs. This feature provides a powerful means to simplify the graphical representation and to make SRNs easier to be understood.
- Marking-dependent arc multiplicities are allowed. This feature can be applied when the number of tokens to be transferred depends on the current marking. A common use of it is to allow a transition to flush all the tokens from a place with a single firing.
- Marking-dependent firing rates are allowed. This feature allows the firing rates of the transitions to be specified as a function of the current marking that can succinctly describe many complex behaviors. For example, service discipline in a queuing network can be represented in an SPN using appropriate marking-dependent firing rates. A common case is a transition whose transition rate is proportional to the number of tokens in its only input place. We denote this dependency by adding a “#” sign next to the transition. More general dependencies are often needed and hence allowed in the SRN formalism.
- Besides the traditional output measures obtained from a GSPN, such as throughput of a transition and the mean number of tokens in a place, more complex **reward functions** can be defined so that all the measures related to Markov reward models can be obtained.

Another important capability captured by SRN formalism is that of specifying the initial marking not as a single marking, but as a probability vector defined over a set of markings. This is often required in transient analysis, if the initial state of the system is uncertain. Initial probabilities are incorporated by adding a vanishing initial marking that transfers to all the markings immediately with the given probabilities [CIAR 1993]. If the number of possible initial markings is large, this method is not practical. However, software packages for SRN (such as SPNP [CIAR 1993]) provide simple *ad hoc* approaches for the specification of the initial probability vector.

SRN formalism has been widely used in performance, reliability, availability, and performability analysis of computer and communication systems. The following are several examples of modeling with SRN.

Example 8.55 (WFS Example)

Let us recall the WFS example with two workstations and one file server (see Examples 3.21, 4.16, 6.12, and 8.24). The SRN model is shown in Figure 8.89. Places P_{fsup} and P_{wsup} represent the working file servers and workstations, while the places P_{fsdn} and P_{wsdn} represent the failed file servers and workstations. Transitions T_{fsfl} and T_{wsfl} represent the failures of file servers and workstations. Note that transition T_{wsfl} has a marking-dependent firing rate. Transitions T_{fsrp} and T_{wsrp} represent the repair of the components. The inhibitor arc from P_{fsdn} to T_{wsup} guarantees that no workstations can be repaired in the event of file server failure. This gives a preemptive repair priority to file server over workstations.

Figure 8.90 is the reachability graph of the SRN, in which each node, a marking, is specified by a 4-tuple $(\#P_{wsup}, \#P_{wsdn}, \#P_{fsup}, \#P_{fsdn})$. The arcs are labeled with the corresponding transition rates.

We now set the reward rate value to 1 for any marking satisfying the condition $\#P_{wsup} > 0$ and $\#P_{fsup} > 0$ and set the reward rate to 0 for all other markings. With this reward rate assignment, the expected steady-state reward rate will give the steady-state availability of the system. Note that this measure is computed and specified at the SRN level and not at the level of the underlying CTMC. Thus the reward rate vector for all the states of the CTMC as well as the CTMC are automatically generated from the SRN. In other words, SRN is a formalism for automatic generation and solution of Markov reward models.

The SRN model discussed above can be easily extended to study the effect of imperfect coverage. Assume that, on a workstation failure, the probability of successfully detecting the failure is c . The failure is not detected with probability $(1 - c)$, leading to the corruption and the failure of the file server. Figure 8.91 shows the SRN model of the WFS example with imperfect coverage.

‡

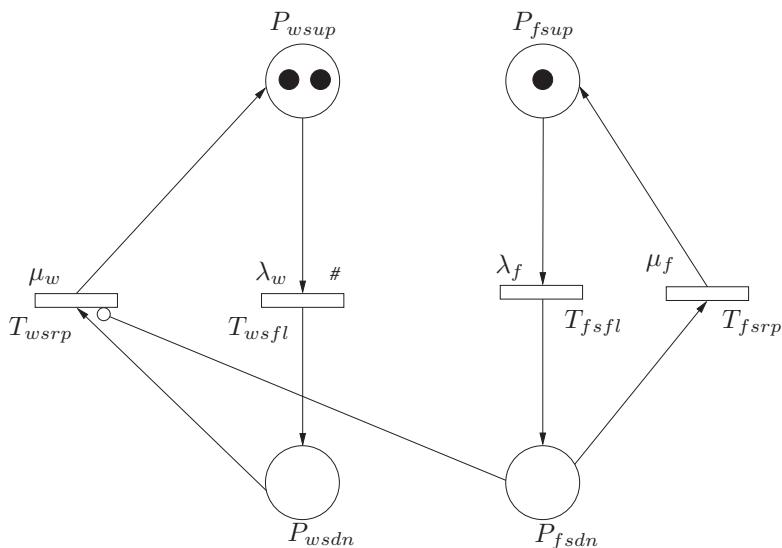


Figure 8.89. SRN for the WFS example

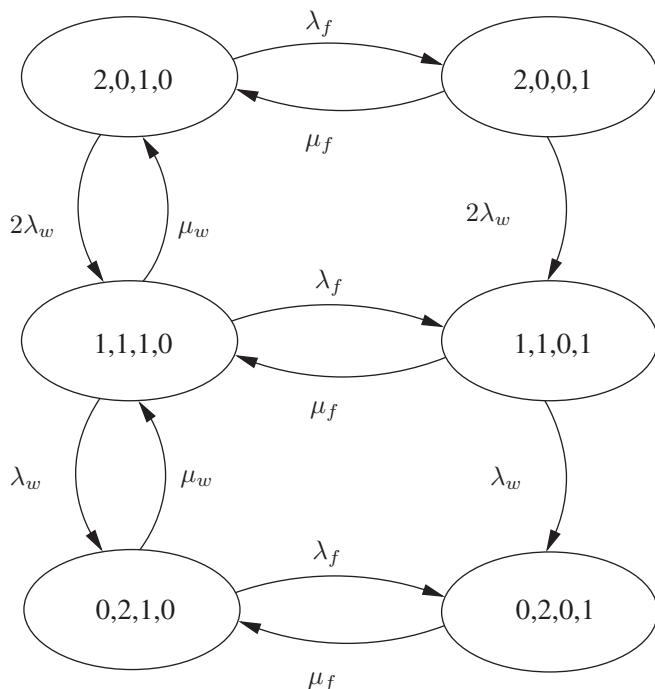


Figure 8.90. Reachability graph of Example 8.55

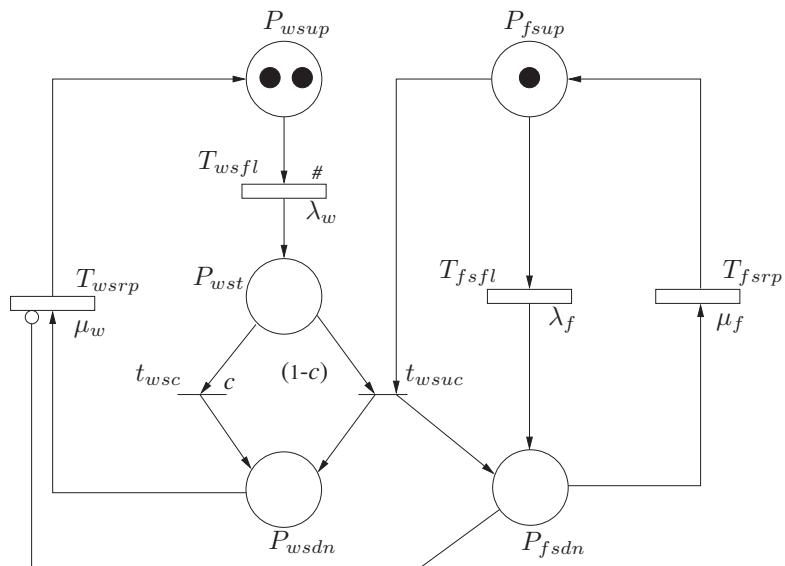


Figure 8.91. SRN for WFS example with imperfect coverage

Example 8.56 (WFS Example with Complex Failure and Repair Dependency)

As real systems generally have complex failure and repair dependencies, we consider a system with 50 workstations and 5 file servers in the WFS example described above. Assume that all the workstations will be automatically shut down if the number of working file servers is less than 5% of the working workstations. Also assume that the workstations cannot be restarted until four or more file servers are up. These constraints can be conveniently specified with guards and arcs with marking-dependent multiplicities. The SRN model for this system is shown in Figure 8.92. The “Z” symbol on the arc denotes the marking-dependent multiplicities, whose values are specified next to the symbol. In this case, both multiplicities are $\#P_{wsup}$, the number of tokens in P_{wsup} . Transition t_{sfl} represents the event of automatic shutdown. When t_{sfl} fires, all the tokens in P_{wsup} are flushed out and are deposited into P_{wsdn} , which means that all working workstations are forced to shutdown. The enabling of t_{sfl} is controlled by its guard function $[g_1] = (\#P_{wsup} \geq 20\#P_{fsup})$, which is a direct translation of the shutdown condition. Another guard $[g_2] = (\#P_{fsup} \geq 4)$ is attached to transition T_{wsrp} , which prevents the workstations from restarting until enough file servers are working.

‡

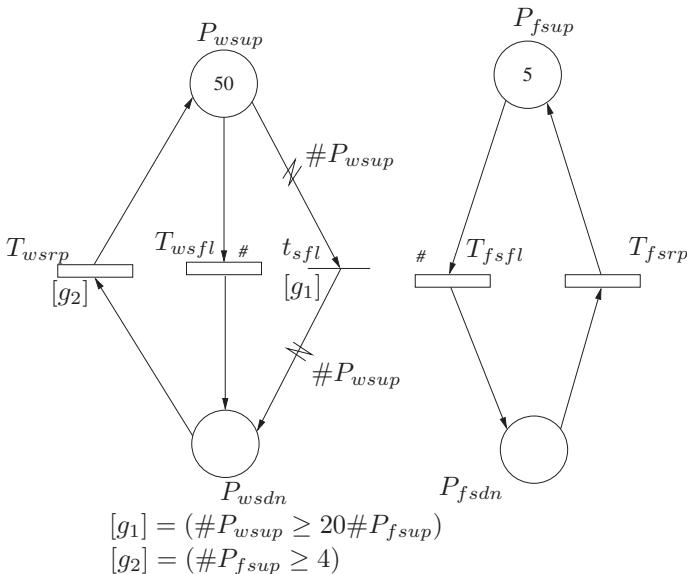


Figure 8.92. SRN for WFS with complex failure and repair dependency

Example 8.57 (Wireless Handoff Performance Model)

We now introduce an SRN model as shown in Figure 8.93, for the wireless handoff performance discussed in Section 8.2.3.2.

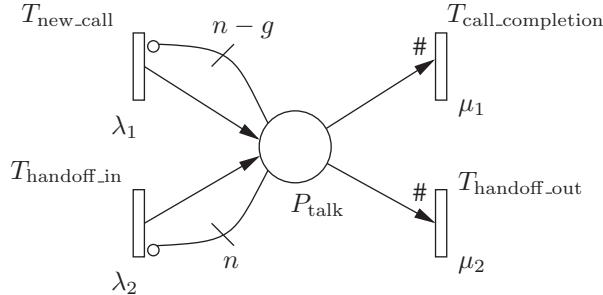


Figure 8.93. The SRN of wireless loss model

The number of tokens in place P_{talk} represents the number of channels that are occupied by either a new-call or a handoff call. The firing of transition $T_{\text{new_call}}$ represents the arrival of new calls and the firing of transition $T_{\text{handoff_in}}$ represents the arrival of a handoff call from neighboring cells. A handoff call will be dropped only when all channels are occupied (i.e., $\#P_{\text{talk}} = n$). This is realized by an inhibitor arc from place P_{talk} to $T_{\text{handoff_in}}$ with multiplicity n . A new call, however, will be blocked if there are no more than g idle channels. This is simply reflected in the SRN by the inhibitor arc from place P_{talk} to transition $T_{\text{new_call}}$ with multiplicity $n - g$. The firings of transition $T_{\text{call_completion}}$ and $T_{\text{handoff_out}}$ represent the completion of a call and the departure of an outgoing handoff call, respectively. The rates of transitions $T_{\text{call_completion}}$ and $T_{\text{handoff_out}}$ are marking-dependent, as indicated by the two “#” symbols next to the transitions. The underlying reachability graph of this SRN is the same as shown in Figure 8.20.

As discussed in Section 8.2.3.2, two steady-state measures are of great interest, namely, the new-call blocking probability, P_b , and the handoff-call dropping probability, P_d . We obtain these two measures by computing the expected steady-state reward rate for the SRN model with the proper assignment of reward rates to the markings. The reward rates to the marking i for the new-call blocking probabilities are

$$r_i = \begin{cases} 1, & \#P_{\text{talk}} \geq n - g \\ 0, & \#P_{\text{talk}} < n - g \end{cases}$$

and that for the handoff dropping probabilities are

$$r_i = \begin{cases} 1, & \#P_{\text{talk}} = n \\ 0, & \#P_{\text{talk}} < n. \end{cases}$$

#

Example 8.58 [CIAR 1992]

Consider a parallel program where data items produced by N_p producers are consumed by N_c consumers. The exchange of items between the N_p producer tasks and the N_c consumer tasks is performed using one additional buffer task. The buffer task stores the incoming items into an array having N_s positions. Producer tasks cannot pass items to the buffer task when the number of non-empty slots is equal to N_s

and consumer tasks cannot retrieve items from the buffer task when the number of non-empty slots is equal to 0. The number of produced items cannot then exceed the number of consumed items plus N_s . The mechanism by which two tasks synchronize and exchange data is the *rendezvous*. Whenever a producer task has an item ready to pass, it issues an *entry call* to the buffer task. If the buffer task accepts this entry call, the rendezvous takes place, the item is copied into the array; similarly, a rendezvous with a consumer retrieves an item from the array. Each *entry* has an associated queue, where tasks making an entry call wait for a rendezvous. The presence of *guards* [g_{put}] and [g_{get}] inhibits the rendezvous at the guarded entry if the boolean value of the guard is false (the guard is *closed*). There are five different policies discussed in [CIAR 1992] to make a choice between a rendezvous with the first producer or the first consumer based on three factors (presence of tasks in each of the two queues and the number of non-empty slots in the array). Here, we only take Consumer First (CF) policy into account: when both guards are open and their associated queues both contain at least one task. A rendezvous with the first consumer in their respective queues takes place immediately. In addition, we assume that a classic processor architecture, where all tasks share the same CPU, is employed.

The system just described is concisely modeled by the SRN shown in Figure 8.94. Tokens in places P_{plocal} , P_{clocal} , and P_{blocal} represent executing tasks (The number of the producers(S_p), the consumers(S_c) and the buffers(S_b), respectively), while token in places P_{pwait} , P_{cwait} , and P_{bwait} represent tasks waiting for a rendezvous at the t_{put} or t_{get} entries. Tokens in place P_{empty} and P_{full} count the number of empty and full slots in the array, respectively. Transitions T_{sp} , T_{sc} , and T_{sb} are assumed to have an exponentially distributed time duration, but they could be changed into a more

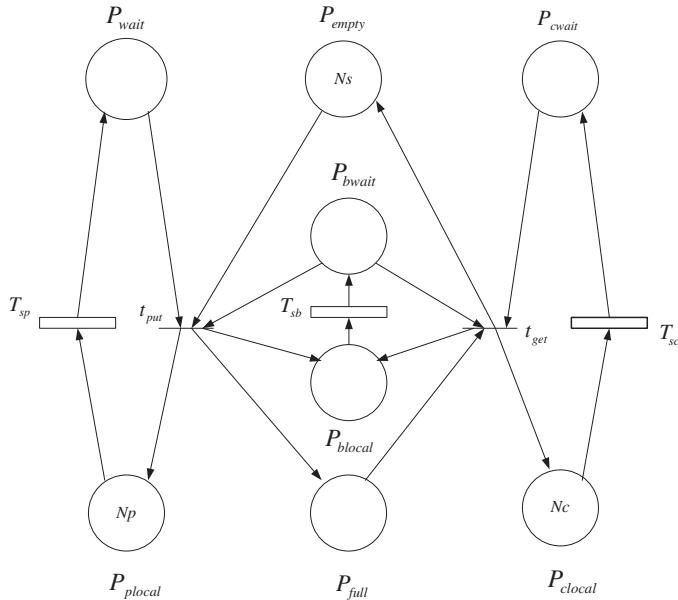


Figure 8.94. The SRN for the producer-consumer system

TABLE 8.11. Firing rates for transitions in the producer-consumer SRN

Transition	Firing rate (sec^{-1})
T_{sp}	$\lambda_p(\#P_{plocal}) / ((\#P_{plocal}) + (\#P_{clocal}) + (\#P_{blocal}))$
T_{sc}	$\lambda_c(\#P_{clocal}) / ((\#P_{plocal}) + (\#P_{clocal}) + (\#P_{blocal}))$
T_{sb}	$\lambda_b / ((\#P_{plocal}) + (\#P_{clocal}) + 1)$

detailed stage-type expansion (using a *subnet*) if more information were available about the actual nature of the distributions. Immediate transitions t_{put} and t_{get} correspond to the actions in the rendezvous, which are modeled as instantaneous, since the time spent for them is likely to be negligible compared with the other blocks of tasks. The enabling functions (or guards) associated with transitions t_{put} and t_{get} for CF policy, respectively: $[g_{put}] = ((\#P_{cwait}) == 0 \text{ or } (\#P_{full}) == 0)$ and $[g_{get}] = 1$. Assume perfect processor sharing with no context switch overhead, and the mean times required to execute blocks S_p , S_c , and S_b for a task running on a processor are λ_p^{-1} , λ_c^{-1} , and λ_b^{-1} , respectively. The specification of the firing rates for the three timed transitions are shown in Table 8.11.

With the SRN, the throughput of the producers, τ_p , can be easily computed. We leave the actual execution of this SRN via SHARPE or SPNP as an exercise.

#

Example 8.59 (The Multiprocessor Model)

Consider the SRN model of the multiprocessor system, discussed in Example 8.32 (in Section 8.4.3). Figure 8.95 shows the SRN availability model. The number of tokens in place P_{up} represents the number of nonfailed processors. The initial number of tokens in this place is n . The firing of transition T_{fail} represents the failure of one of the processors. The inhibitor arcs from the places P_{cov} and place P_{uncov} ensure that when the system is undergoing a reconfiguration or a reboot, no further failures can occur. The firing rate of transition T_{fail} is marking-dependant: $\text{Rate}(T_{fail}) = \gamma \#P_{up}$. When a token appears in place P_{fail} , the immediate transitions t_{cov} , t_{uncov} , and t_{quick} are enabled. If no token is in place P_{up} , then immediate transition t_{quick} will be enabled and will be assigned a higher priority than t_{cov} and t_{uncov} ; this is done to ensure that for the last processor to fail, there is no reconfiguration or reboot delay. A token will be deposited in place P_{rep} by firing immediate transition t_{quick} . Otherwise t_{cov} or t_{uncov} will fire with probabilities c and $1 - c$, respectively. In the case that t_{cov} fires, the system is reconfigured by firing the transition $T_{reconfig}$ whose firing time is $\text{EXP}(\delta)$. In the case that t_{uncov} fires, the system is rebooted by firing the transition T_{reboot} whose firing time is $\text{EXP}(\beta)$. The transition T_{rep} , with firing time $\text{EXP}(\tau)$, fires if at least one token is in place P_{rep} and upon firing it will deposit one token in place P_{up} .

Measures such as system availability, capacity-oriented availability, and total loss probability can be obtained by computing expected steady-state reward rate with proper choice of reward rates for the SRN model. To obtain the steady-state availability, we assign the reward rate r_i to marking i as

$$r_i = \begin{cases} 1, & \#P_{up} \geq 1 \text{ and } (\#P_{cov} + \#P_{uncov}) = 0 \\ 0, & \text{otherwise} \end{cases}.$$

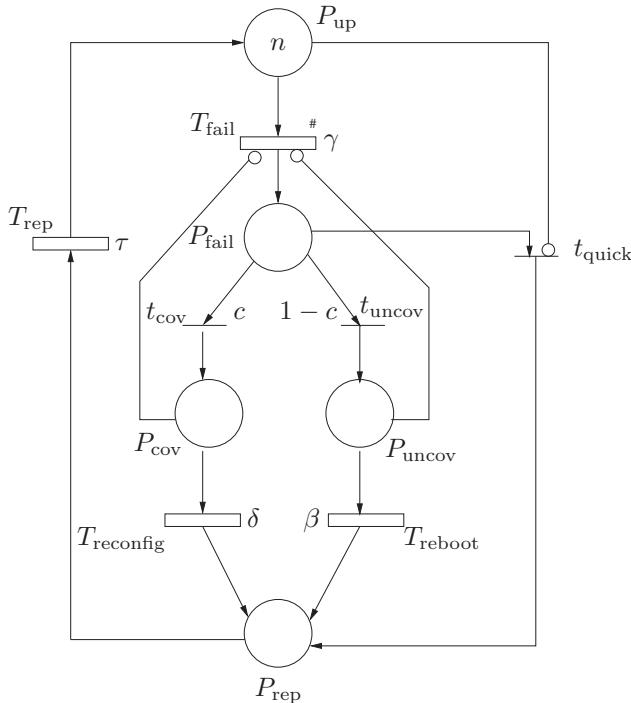


Figure 8.95. SRN of a multiprocessor model

Similarly, to get the scaled capacity-oriented availability, we choose the formula $r_i = (\#P_{\text{up}})/n$, for an up marking i .

The total loss probability of the multiprocessor system can be computed by combining the availability and the performance. Recall the performance model of the multiprocessor system (see Figure 8.51) and the state diagram shown in Figure 8.52. Using the values of $q_b(i)$ and $P(R_b(i) > d)$ (refer to problem 2 before the start of Section 8.2.3.1), we obtain the total loss probability by assigning the reward rates for marking i :

$$r_i = \begin{cases} 1, & \text{if } \#P_{\text{up}} = 0 \text{ or } \#P_{\text{cov}} = 1, \\ q_b(\#P_{\text{up}}) & \text{if } \#P_{\text{up}} > 0 \text{ and } \#P_{\text{cov}} = 0 \\ + [1 - q_b(\#P_{\text{up}})] [P(R_b(\#P_{\text{up}}) > d)], & \text{and } \#P_{\text{uncov}} = 0. \end{cases}$$

‡

Further studies of availability models using Petri nets are available in the literature [FRIC 1998, MALH 1995, MUPP 1992a, IBE 1989a], as are further examples of performance analysis [IBE 1990, IBE 1993, AJMO 1995] and of modeling software fault-tolerance [TOME 1994].

In the examples presented in this chapter, we have concentrated on SPN as well as SRN models with exponentially distributed firing time transitions. When the assumption of exponential distribution is relaxed, the underlying models can be solved using semi-Markov processes or Markov regenerative processes under certain specific conditions. Further literature on performance analysis of non-Markovian SPN models [GERM 2000, LOGO 1995, LIND 1998] and on reliability and performability analysis of non-Markovian SRN models [FRIC 1998a] is available. Bobbio *et al.* have provided a comprehensive account of the evolution of SPN [BOBB 1998].

Problems

1. Rewrite the SRN of the WFS example without using inhibitor arcs. Discuss the benefits of SRN modeling with inhibitor arcs as opposed to guard functions.
2. Draw the extended reachability graph of the WFS example with imperfect coverage, eliminating the vanishing markings to obtain the corresponding CTMC.
3. Draw the stochastic Petri net for the WFS example, considering nonpreemptive repair priority as opposed to preemptive repair priority used in Figure 8.89. Draw the corresponding reachability graph and CTMC.
4. Extend Example 8.54 to include failure and repair for the trunks. When a failure occurs, all the contents in the system are cleared. Assume that the failure rate γ is related with number in the system q_l as a function

$$\gamma = \begin{cases} \gamma_0, & \text{if } q_l < \frac{c}{2}, \\ \gamma_1 (q_l - \frac{c}{2}), & \text{if } q_l \geq \frac{c}{2}, \end{cases}$$

and the repair rate τ is a constant. Construct the SRN model and derive the corresponding CTMC. Vary c from 0 to 5.

5. For the multiprocessor model of Figure 8.95, assume that the processor failure rate γ is a function of its utilization u , where $u = \lambda/(\mu n)$. Following Iyer [IYER 1986], use the exponential function

$$\gamma(u) = ge^u. \quad (8.149)$$

We assume that the nominal failure rate is measured at $u = 0.7$ so that $1/g = 6000 \times e^{0.7}$. With these assumptions, calculate downtime, $D(n)$, in minutes per year as a function of n for the different values of λ in the SRN model using either the SHARPE or the SPNP package.

6. Draw the SRNs for the BTS system availability model and the system reliability model for Example 1.21. Define the reward functions for the measures. Now do a GSPN model for the same problem. Note the difficulty of solving such problems without having the capability to define reward rates at the net level [MALH 1995].

7. Return to the example of 2 control channels and 3 voice channels (problem 7 in Section 6.6). Construct an SRN availability model assuming a single repair person for the case that a control channel can also function as a voice channel. Define the reward function for the system availability and plot instantaneous availability as a function of time using SPNP. Modify the SRN model to compute system reliability both with and without repair.

REFERENCES

- [AJMO 1995] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, Wiley, New York, 1995.
- [AKIM 1993] H. Akimaru and K. Kawashima, *Teletraffic: Theory and Application*, Springer-Verlag, Heidelberg, 1993.
- [ARNO 1973] T. Arnold, “The concept of coverage and its effect on the reliability model of a repairable system,” *IEEE Trans. Comput.* **C-22**, 251–254 (March 1973).
- [BEUT 1980] F. J. Beutler, Sojourn Times in Markov Queuing Networks: Little’s Formula Revisited, Technical Report, Computer Information and Control Engineering Program, Univ. Michigan, Ann Arbor, 1980.
- [BHAT 1984] U. N. Bhat, *Elements of Applied Stochastic Processes*, Wiley, New York, 1984.
- [BLAK 1988] J. T. Blake, A. Reibman, and K. S. Trivedi, “Sensitivity analysis of reliability and performance measures for multiprocessor systems,” *Proc. ACM SIGMETRICS Conf. Measurement and Modeling of Computer Systems*, Santa Fe, NM, 1998, pp. 177–186.
- [BOBB 1998] A. Bobbio, A. Puliafito, M. Telek, and K. Trivedi, “Recent developments in stochastic Petri nets,” *J. of Circuits, Systems, Comput.* **8**(1), 119–158 (1998).
- [BOLC 1998] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains*, Wiley, New York, 1998.
- [CAO 2000] Y. Cao, H.-R. Sun, and K. S. Trivedi, “Performability analysis of TDMA cellular systems,” *Int. Conf. Performance and QoS of Next Generation Networking, P & Q Net 2000*, Nagoya, Japan, Nov. 2000.
- [CHIM 1993] P. F. Chimento and K. S. Trivedi, “The completion time of programs on processors subject to failure and repair,” *IEEE Trans. Comput.* **42**(10), 1184–1194 (1993).
- [CHIO 1995] G. Chiola, “GreatSPN 1.5 Software Architecture,” *Performance Evaluation*, 121–136 (1995).
- [CHOI 1994] H. Choi, V. G. Kulkarni, and K. S. Trivedi, “Markov regenerative stochastic Petri nets,” *Performance Evaluation* **20**(1–3), 337–357 (1994).

- [CHOI 1998] B. D. Choi, B.-C. Shin, K. B. Choi, D. H. Han, and J. S. Jang, “Priority queue with two-state Markov-Modulated arrivals,” *IEE Proc. Commun.* **145**(3), 152–158, 1998.
- [CHWA 1998] H. Choi, W. Wang and K. S. Trivedi, “Analysis of conditional MTTF of fault-tolerant systems,” *Microelectron. and Reliability* **38**(3):393–401 (1998).
- [CIAR 1992] G. Ciardo, J. Muppala, and K. S. Trivedi, “Analyzing concurrent and fault-tolerant software using stochastic Petri nets, *J. Parallel Distr. Comput.* **15**, 255–269–(1992).
- [CIAR 1993] G. Ciardo, A. Blakemore, P. Chimento, J. Muppala, and K. Trivedi, “Automated generation and analysis of Markov reward models using stochastic reward nets,” C. Meyer and R. J. Plemmons (eds.), *Linear Algebra, Markov Chains, and Queueing Models, IMA Volumes in Mathematics and its Applications*, Vol. 48, Springer-Verlag, Heidelberg, Germany, 1993, pp. 145–191.
- [CINL 1975] E. Cinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [DENN 1978] P. J. Denning and J. P. Buzen, “The operational analysis of queuing network models,” *ACM Comput. Surv.* **10**, 225–241 (1978).
- [DIJK 1992] N. M. Van Dijk, “Uniformization for nonhomogeneous Markov chains,” *Oper. Res. Lett.* **12**, 283–291–(1992).
- [DUGA 1989] J. B. Dugan and K. S. Trivedi, “Coverage modeling for dependability analysis of fault-tolerant systems,” *IEEE Trans. Comput.* **38**(6), 775–787–(1989).
- [FISC 1993] W. Fischer and K. Meier-Hellstern, “The Markov-modulated Poisson process (MMPP) cookbook,” *Performance Evaluation* **18**(2), 149–171 (1993).
- [FOGX 1988] B. L. Fox and P. W. Glynn, “Computing Poisson probabilities,” *Commun. ACM* **31**(4), 440–445 (1988).
- [FRIC 1998] R. Fricks and K. S. Trivedi, “Availability modeling of energy management systems,” *Microelectron. and Reliability* **38**, 727–743 (1998).
- [FRIC 1998a] R. Fricks, M. Telek, A. Puliafito, and K. S. Trivedi, “Markov renewal theory applied to performability evaluation,” in K. Bagchi and G. Zobrist (eds.), *State-of-the-Art in Performance Modeling and Simulation. Modeling and Simulation of Advanced Computer Systems: Applications and Systems*, Gordon and Breach, Newark, NJ, 1998, pp. 193–236.
- [FRIC 1999] R. Fricks, S. Garg, and K. S. Trivedi, “Modeling failure dependencies in real-time computer architectures,” *Proc. 10th INFORMS Applied Probability Conf.*, Univ. Ulm, Germany, July 26–28, 1999.
- [FULL 1975] S. H. Fuller, “Performance evaluation,” in H. S. Stone, (ed.), *Introduction to Computer Architecture*, Science Research Associates, Chicago, 1975, pp. 474–545.
- [GARG 1998] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, “Analysis of preventive maintenance in transactions based software systems,” *IEEE Trans. Comput.* **47**(1), 96–107 (1998).

- [GARG 1999] S. Garg, Y. Huang, C. M. Kintala, K. S. Trivedi, and S. Yajnik, "Performance and reliability evaluation of passive replication schemes in application level fault tolerance," in *Proc. 29th Annual Int. Symp. Fault Tolerant Computing (FTCS)*, Madison, Wisconsin, pp. 15-18-, June 15-18, 1999.
- [GEIS 1983] R. M. Geist and K. S. Trivedi, "The integration of user perception in the heterogeneous M/M/2 queue," in A. Agrawala and S.K. Tripathi (eds.), *PERFORMANCE '83*, North-Holland, 1983, pp. 203-216.
- [GERM 2000] R. German, *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*, Wiley, New York, 2000.
- [GOEL 1979] A. L. Goel and K. Okumoto, "A time dependent error detection rate model for software reliability and other performance measures," *IEEE Trans. Reliability* **R-28**(3), 206-211 (1979).
- [GOEL 1985] A. L. Goel, "Software reliability models: assumptions, limitations, and applicability," *IEEE Trans. Software Eng.* **SE-11**, 1411-1423 (1985).
- [GOKH 1998] S. Gokhale and K. S. Trivedi, "Log-logistic software reliability growth model," *Proc. 3rd IEEE Int. High Assurance Systems Engineering Symp., HASE98*, Washington DC, Nov. 1998.
- [GOYA 1987] A. Goyal, S. S. Lavenberg, and K. S. Trivedi, "Probabilistic modeling of computer system availability," *Ann. Oper. Res.* **8**, 285-306, 1987.
- [GRAS 2000] W. Grassman (ed.), *Computational Probability*, Kluwer Academic Publishers, Amsterdam, 2000.
- [GROS 1998] D. Gross and C. M. Harris, *Fundamentals of Queuing Theory*, 3rd ed., Wiley, New York, 1998.
- [HARI 2001] G. Haring, R. Marie, R. Puigjaner, and K. S. Trivedi, "Loss formulae and their optimization for cellular networks," *IEEE Trans. Vehic. Technol.* **VT-50**, 664-673, (2001).
- [HAVE 2001] B. R. Haverkort, R. Marie, K. S. Trivedi, and G. Rubino (eds.), *Performability Modelling Tools and Techniques*, Wiley, New York, 2001.
- [HEID 1996] P. Heidelberger, J. Muppala, and K. S. Trivedi, "Accelerating mean time to failure computations," *Performance Evaluation* **27 28**, 627-645-(1996).
- [HEIM 1990] D. Heimann, N. Mittal and K. S. Trivedi, "Availability and reliability modeling for computer systems," in M. Yovitts (ed.), *Advances in Computers*, Vol. 31, Academic Press, San Diego, 1990, pp. 175-233.
- [HONG 1986] D. Hong and S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized handoff procedures," *IEEE Trans. Vehic. Technol.* **VT-35**, 77-92 (1986).
- [HOSS 2000] M. M. Hosseini, R. M. Kerr and R. B. Randall, "An inspection model with minimal and major maintenance for a system with deterioration and Poisson failures," *IEEE Trans. Reliability* **49**(1), 88-98 (2000).
- [HUAN 1993] Y. Huang and C. M. Kintala, "Software implemented fault tolerance: technologies and experience," *Proc. 23rd Int. Symp. on Fault-Tolerant Computing (FTCS)*, Toulouse, France, 1993, pp. 2-9.

- [HUNT 1999] S. W. Hunter and W. E. Smith, "Availability modeling and analysis of a two node cluster," *Proc. 5th Int. Conf. on Information Systems, Analysis and Synthesis*, Orlando, FL, Oct. 1999.
- [IBE 1989a] O. Ibe, A. Sathaye, R. Howe, and K. S. Trivedi, "Stochastic Petri net modeling of VAXcluster availability," *Proc. Third Int. Workshop on Petri Nets and Performance Models (PNPM89)*, Kyoto, 1989, pp. 112–121.
- [IBE 1989b] O. Ibe, R. Howe, and K. S. Trivedi, "Approximate availability analysis of VAXCluster systems," *IEEE Trans. Reliability* **R-38**(1), 146–152 (1989).
- [IBE 1990] O. Ibe and K. S. Trivedi, "Stochastic Petri net models of polling systems," *IEEE J. Selected Areas Commun.* **8**(9), 1649–1657 (1990).
- [IBE 1993] O. Ibe, H. Choi and K. S. Trivedi, "Performance evaluation of client-server systems," *IEEE Trans. Parallel Distrib. Syst.* **4**(11), 1217–1229 (1993).
- [IYER 1986] R. K. Iyer, D. J. Rosetti, and M.-C. Hsueh, "Measurement and modeling of computer reliability as affected by system activity," *ACM Trans. Comput. Syst.* **4**, 214–237 (1986).
- [KANG 1997] S. H. Kang, C. Oh, and D. K. Sung, "A traffic measurement-based modeling of superposed ATM cell streams," *IEICE Trans. Commun.* **E80-B**(3), 434–440 (1997).
- [JELI 1972] Z. Jelinski and P. B. Moranda, "Software reliability research," in W. Freiberger (ed.), *Statistical Computer Performance Evaluation*, Academic Press, New York, 1972, pp. 485–502.
- [KLEI 1975] L. Kleinrock, *Queueing Systems*, Vol. I, *Theory*, Wiley, New York, 1975.
- [KLEI 1976] L. Kleinrock, *Queueing Systems*, Vol. II, Wiley, New York, 1976.
- [KOBA 1978] H. Kobayashi, *Modeling and Analysis*, Addison-Wesley, Reading, MA, 1978.
- [LAIJ 1998] R. Lai and A. Jirachieffattana, *Communication Protocol Specification and Verification*, Kluwer Academic Publishers, Amsterdam, 1998.
- [LAPR 1995] J.-C. Laprie, J. Arlat, C. Beounes, and K. Kanoun, "Architectural issues in software fault tolerance," in M. Lyu (ed.), *Software Fault Tolerance*, Wiley, New York, 1995.
- [LIND 1998] C. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*, Wiley, New York, 1998.
- [LOGO 1995] D. Logothetis, A. Puliafito, and K. S. Trivedi, "Markov regenerative models," *Proc. IEEE Int. Computer Performance and Dependability Symp.* Erlangen, Germany, 1995, pp. 134–143–.
- [LUCA 1990] D. Lucantoni, K. Meier-Hellstern, and M. F. Neuts, "A single-server queue with server vacations and a class of non-renewal arrival processes," *Adv. Appl. Probability* **22**, 676–705 (1990).
- [MALH 1995] M. Malhotra and K. S. Trivedi, "Dependability modeling using Petri nets," *IEEE Trans. Reliability* **44**(3), 428–440–(1995).

- [MORS 1958] P. M. Morse, *Queues, Inventories and Maintenance*, Wiley, New York, 1958.
- [MUPP 1992a] J. Muppala, A. Sathaye, R. Howe, and K. S. Trivedi, “Dependability modeling of a heterogeneous VAXcluster system using stochastic reward nets,” in D. Avresky (ed.), *Hardware and Software Fault Tolerance in Parallel Computing Systems*, Ellis Horwood, UK, 1992, pp. 33–59–.
- [MUPP 1992b] J. Muppala and K. S. Trivedi, “Numerical transient solution of finite Markovian queueing systems,” in U. N. Bhat and I. V. Basawa (eds.), *Queueing and Related Models*, Oxford Univ. Press, 1992, pp. 262–84.
- [MUPP 1994] J. Muppala, M. Malhotra, and K. S. Trivedi, “Stiffness-tolerant methods for transient analysis of stiff Markov chains,” *Microelectron. and Reliability* **34**(11), 1825–1841 (1994).
- [MUPP 1996] J. Muppala, M. Malhotra, and K. Trivedi, “Markov dependability models of complex systems: analysis techniques,” in S. Ozekici (ed.), *Reliability and Maintenance of Complex Systems*, Springer-Verlag, Berlin, 1996, pp. 442–486.
- [MURA 1989] T. Murata, “Petri nets: properties, analysis and applications,” *Proc. IEEE* **37**(4), 541–560 (1989).
- [MUSA 1983] J. D. Musa and K. Okumoto, “A logarithmic Poisson execution time models for software reliability measurement,” *Proc. 7th Int. Conf. Software Eng.*, Orlando, 1983, pp. 230–237.
- [NEUT 1978] M. F. Neuts, “Renewal processes of phase type,” *Naval Res. Logistics Quart.* **25**, 445–454 (1978).
- [NUTT 1997] G. Nutt, *Operating Systems, A Modern Perspective*, Addison-Wesley, Reading, MA, 1997.
- [ORTA 1999] R. Ortalo, Y. Deswart, and M. Kaâniche, “Experimenting with quantitative evaluation tools for monitoring operational security,” *IEEE Trans. Software Eng.* **25**(5), 633–650 (1999).
- [PARZ 1962] E. Parzen, *Stochastic Processes*, Holden-Day, San Francisco, CA, 1962.
- [RAMA 2000] S. Ramani, S. Gokhale, and K. S. Trivedi, “SREPT: Software reliability estimation and prediction tool,” *Performance Evaluation* **39**, 37–60 (2000).
- [RAME 1995] A. V. Ramesh and K. S. Trivedi, “Semi-numerical transient analysis of Markov models,” *Proc. 33rd ACM Southeast conf., Clemson*, SC, March 1995.
- [REIB 1988] A. Reibman and K. S. Trivedi, “Numerical transient analysis of Markov models,” *Comput. and Oper. Res.* **15**(1), 19–36 (1988).
- [REIB 1989] A. Reibman, R. Smith, and K. Trivedi, “Markov and Markov reward models: A survey of numerical approaches,” *Eur. J. Oper. Res.* **40**, 257–267 (1989).
- [RIND 1995] A. Rindos, S. Wolet, I. Viniotis, and K. S. Trivedi, “Exact methods for the transient analysis of nonhomogeneous continuous time Markov chains,” in W. J. Stewart (ed.), *2nd Int. Workshop on the Numerical Solution of Markov Chains*, Kluwer Academic Publishers, Boston, 1995.

- [ROSS 1970] S. M. Ross, *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, CA, 1970.
- [ROSS 1983] S. M. Ross, *Stochastic Processes*, Wiley, New York, 1983.
- [SAHN 1996] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer System: An Example-based approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.
- [SHIN 1986] K. Shin and C. Krishna, “New performance measures for design and evaluation of real-time multiprocessors,” *Comput. Sys. Sci. Eng.* **1**(4), 179–192 (1986).
- [STEW 1994] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton Univ. Press, Princeton, NJ, 1994.
- [STID 1974] S. Stidham, Jr., “A last word on $L = \lambda W$,” *Oper. Res.* **22**, 417–421 (1974).
- [STIF 1980] J. J. Stiffler, “Robust detection of intermittent faults,” *Proc. 10th Int. Symp. on Fault-Tolerant Comput.*, Kyoto, Japan, 1980, pp. 216–218.
- [SUN 1999] H.-R. Sun, Y. Cao, J. J. Han, and K. S. Trivedi, “Availability and performance evaluation for automatic protection switching in TDMA wireless system,” *Pacific Rim Dependability Conf.* 1999, pp. 15–22.
- [TOME 1991] L. A. Tomek and K. S. Trivedi, “Fixed point iteration in availability modeling,” in M. Dal Cin (ed.), *Proc. Fifth Int. GI/ITG/GMA Conf. on Fault-Tolerant Comput. Syst.*, Springer-Verlag, Berlin, 1991, pp. 229–240.
- [TOME 1994] L. A. Tomek and K. S. Trivedi, “Analyses using stochastic reward nets,” in M. Lyu (ed.), *Software Fault Tolerance*, Wiley, New York, 1994.
- [TOWS 1978] D. F. Towsley, J. C. Browne, and K. M. Chandy, “Models for parallel processing within programs: application to CPU: I/O and I/O: I/O overlap,” *CACM* **21**(10), 821–831 (Oct. 1978).
- [TRIV 1990] K. S. Trivedi, A. Sathaye and R. Howe, “Should I add a processor?” *23rd Annual Hawaii Conf. Sys. Sci.*, Jan. 1990, pp. 214–221.
- [TRIV 1992] K. Trivedi, J. Muppala, S. Woolet, and B. Havercort, “Composite performance and dependability analysis,” *Performance Eval.* **14**(3–4), 197–216 (1992).
- [WANG 1995] S. S. Wang and J. A. Silverster, “An approximate model for performance evaluation of real-time multimedia communication systems,” *Performance Eval.* **22**(3), 239–256 (1995).
- [WOLF 1982] R. Wolff, “Poisson arrivals see time averages,” *Oper. Res.* **30**, 223–231 (1982).
- [YOUS 1996] S. Y. Yousef and J. A. Schormans, “Performance, interarrival, and correlation analysis of four-phase MMPP model in ATM-based B-ISDN,” *IEE Proc. Commun.* **143**(6), 363–368 (1996).

Chapter 9

Networks of Queues

9.1 INTRODUCTION

We have studied continuous-time Markov chains of the birth–death type in Chapter 8. Such CTMCs are characterized by a simple product-form solution [equation (8.34)] and *a large number of applications*. When we remove the restriction of nearest-neighbor transitions only, we may not have the convenient product-form solution. It is logical to ask whether there is a class of Markov chains that subsumes birth–death processes and that possesses a product-form solution. One important generalization of the birth–death process that we consider is a network of queues. Such networks can model problems of contention that arise when a set of resources is shared. A node or a service center represents each resource. Thus, in a model for computer system performance analysis, we may have a service center for the CPU(s), a service center for each I/O channel, and possibly others. A service center may have one or more servers associated with it. If a job requesting service finds all the servers at the service center busy, it will join the queue associated with the center, and at a later point in time, when one of the servers becomes idle, a job from the queue will be selected for service according to some scheduling discipline. After completion of service at one service center, the job may move to another service center for further service, reenter the same service center, or leave the system.

We shall consider two types of networks: open and closed. An **open queuing network** is characterized by one or more sources of job arrivals and correspondingly one or more sinks that absorb jobs departing from the network. In a **closed queuing network**, on the other hand, jobs neither enter nor depart from the network.

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

The probabilities of transitions between service centers and the distribution of job service times at each center characterize the behavior of jobs within the network. For each center the number of servers, the scheduling discipline, and the size of the queue must be specified. Unless stated otherwise, we assume that the scheduling is FCFS and that each server has a queue of unlimited capacity. For an open network, a characterization of job arrival processes is needed, and for a closed network, the number of jobs in the network must be specified.

Queuing networks have been successfully used in performance modeling of computer and communication systems [BOLC 1998, HAVE 1998]. They are especially suited for representing resource contention and queuing for service. Most of the analysis techniques discussed in this chapter have concentrated on the evaluation of averages of various performance measures such as throughput, utilization, and response time using efficient algorithms such as convolution and mean-value analysis (MVA) [LAVE 1983]. For real-time systems, however, the knowledge of response time distributions is required in order to compute and/or minimize the probability of missing a deadline.

In open networks, the response time or sojourn time of a customer is defined as the time from its entry into the network until its exit from the network. In closed queuing networks, response time is defined as the time a customer requires to complete one cycle in the queuing network, starting from and returning to a particular node.

Closed-form solutions for response time distribution in queuing networks are available in only very few cases such as the $M/M/n$ FCFS queue. Methods for computing the Laplace transform of the response time distribution are available for queuing networks with special structure. Boxma and Daduna have provided an excellent survey of these methods [BOXM 1990]. As mentioned in the survey, it is very difficult to obtain closed-form solutions for queuing networks with a general structure. In the last section of this chapter, the problem of response time distribution in Markovian queuing networks is addressed.

Consider the two-stage tandem network shown in Figure 9.1. The system consists of two nodes with respective service rates, μ_0 and μ_1 . The external arrival rate is λ . The output of the node labeled 0 is the input to the node labeled 1. The service time distribution at both nodes is exponential, and the arrival process to the node labeled 0 is Poisson.

This system can be modeled as a stochastic process whose states are specified by pairs (k_0, k_1) , $k_0 \geq 0, k_1 \geq 0$, where k_i ($i = 0, 1$) is the number of jobs at server i in the steady state. The changes of state occur on a completion of

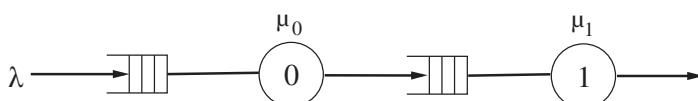


Figure 9.1. A two-stage tandem network

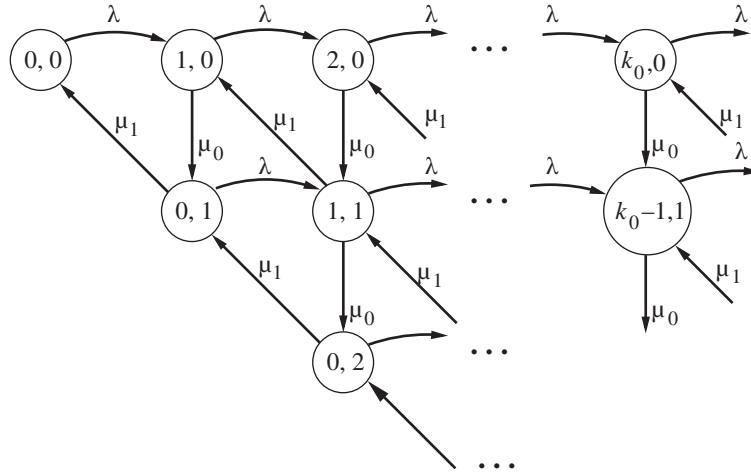


Figure 9.2. The state diagram for the two-stage tandem network

service at one of the two servers or on an external arrival. Since all interevent times are exponentially distributed (by our assumptions), it follows that the stochastic process is a homogeneous continuous-time Markov chain with the state diagram shown in Figure 9.2.

For $k_0, k_1 > 0$, the transitions into and out of that state are shown in Figure 9.3. Let $p(k_0, k_1)$ be the joint probability of k_0 jobs at node 0 and k_1 jobs at node 1 in the steady state. Equating the rates of flow into and out of the state, we obtain the following balance equations:

$$\begin{aligned} (\mu_0 + \mu_1 + \lambda)p(k_0, k_1) &= \mu_0 p(k_0 + 1, k_1 - 1) + \mu_1 p(k_0, k_1 + 1) \\ &\quad + \lambda p(k_0 - 1, k_1), k_0 > 0, k_1 > 0. \end{aligned} \quad (9.1)$$

For the boundary states, we have

$$\begin{aligned} (\mu_0 + \lambda)p(k_0, 0) &= \mu_1 p(k_0, 1) + \lambda p(k_0 - 1, 0), \quad k_0 > 0, \\ (\mu_1 + \lambda)p(0, k_1) &= \mu_0 p(1, k_1 - 1) + \mu_1 p(0, k_1 + 1), k_1 > 0, \\ \lambda p(0, 0) &= \mu_1 p(0, 1). \end{aligned}$$

The normalization is provided by

$$\sum_{k_0 \geq 0} \sum_{k_1 \geq 0} p(k_0, k_1) = 1.$$

It is easily shown by direct substitution that the following equation is the solution to the preceding balance equations:

$$p(k_0, k_1) = (1 - \rho_0)\rho_0^{k_0}(1 - \rho_1)\rho_1^{k_1}, \quad (9.2)$$

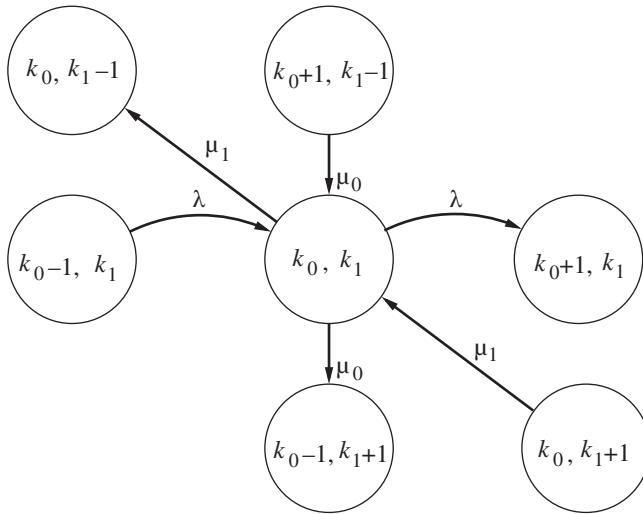


Figure 9.3. Portion of the state diagram for equation (9.1)

where $\rho_0 = \lambda/\mu_0$ and $\rho_1 = \lambda/\mu_1$. The condition for stability of the system is that both ρ_0 and ρ_1 are less than unity.

Equation (9.2) is a product-form solution similar to that of an $M/M/1$ queue. Observe that the node 0 in Figure 9.1 has a Poisson arrival source of rate λ and exponentially distributed service time. Therefore, the node labeled 0 is an $M/M/1$ queue. It follows that the pmf of the number of jobs N_0 at node 0 in the steady state is given by [see also equation (8.34)]

$$P(N_0 = k_0) = p_0(k_0) = (1 - \rho_0)\rho_0^{k_0}.$$

Burke [BURK 1956] has shown that the output of an $M/M/1$ queue is also Poisson with rate λ (you are asked to verify this result in problem 2 at the end of this section). Thus, the second queue in Figure 9.1 is also an $M/M/1$ queue with server utilization $\rho_1 = \lambda/\mu_1$ (assumed to be < 1). Hence, the steady-state pmf of the number of jobs N_1 at node 1 is given by

$$P(N_1 = k_1) = p_1(k_1) = (1 - \rho_1)\rho_1^{k_1}.$$

The joint probability of k_0 jobs at node 0 and k_1 jobs at node 1 is given by equation (9.2):

$$p(k_0, k_1) = (1 - \rho_0)\rho_0^{k_0}(1 - \rho_1)\rho_1^{k_1} = p_0(k_0)p_1(k_1).$$

Thus the joint probability $p(k_0, k_1)$ is the product of the marginal probabilities, $p_0(k_0)$ and $p_1(k_1)$; hence random variables N_0 and N_1 are independent in

the steady state. Therefore, the two queues are independent $M/M/1$ queues. As the arrival rate λ increases, the node with the larger value of ρ will introduce instability. Hence the node with the largest value of ρ is called the “bottleneck” of the system. The product form solution (9.2) can be generalized to an m -stage tandem queue.

Example 9.1

A repair facility shared by a large number of machines has two sequential stations with respective rates, 1 per hour and 2 per hour. The cumulative failure rate of all the machines is 0.5 per hour. Assuming that the system behavior may be approximated by the two-stage tandem queue of Figure 9.1, determine the average repair time.

Given $\lambda = 0.5$, $\mu_0 = 1$, $\mu_1 = 2$, we have $\rho_0 = 0.5$ and $\rho_1 = 0.25$. The average length of the queue at station i ($i = 0, 1$) is then given by [using formula (8.35)]

$$E[N_i] = \frac{\rho_i}{1 - \rho_i};$$

hence

$$E[N_0] = 1 \quad \text{and} \quad E[N_1] = \frac{1}{3}.$$

Using Little's formula, the repair delays at the two stations are respectively given by

$$E[R_0] = \frac{E[N_0]}{\lambda} = 2h \quad \text{and} \quad E[R_1] = \frac{E[N_1]}{\lambda} = \frac{2}{3} h.$$

Hence the average repair time is given by:

$$E[R] = E[R_0] + E[R_1] = \frac{8}{3} h.$$

This can be decomposed into waiting time at station 0 ($= 1$ h), the service time at station 0 ($= 1/\mu_0 = 1$ h), the waiting time at station 1 ($= \frac{1}{6}$ h), and the service time at station 1 ($1/\mu_1 = \frac{1}{2}$ h). The probability that both service stations are idle is given by

$$p(0,0) = (1 - \rho_0)(1 - \rho_1) = \frac{3}{8}.$$

Station 0 is the bottleneck of the repair facility.

#

Problems

1. In Chapter 8 we derived the distribution function of the response time of an isolated $M/M/1$ FCFS queue [see equation (8.40)]. Using this result, derive the distribution function of the response time for the tandem network of Figure 9.1. From this, obtain the variance of the response time.
2. * Consider an $M/G/1$ queue with FCFS scheduling . Let the random variables A , B , and D , respectively, denote the interarrival time, the service time, and the

interdeparture time. By conditioning on the number of jobs in the system and then using the theorem of total Laplace transforms, show that in the steady state

$$L_D(s) = \rho L_B(s) + (1 - \rho)L_A(s)L_B(s),$$

where ρ is the traffic intensity so that $\rho = E[B]/E[A]$.

Point out why the assumption of Poisson arrival stream is needed to derive this result. Then, specializing to the case of $M/M/1$ queue, show that

$$L_D(s) = L_A(s).$$

This verifies Burke's result that the output process of an $M/M/1$ FCFS queue is Poissonian. Note that the independence of successive interdeparture times needs to be shown in order to complete the proof.

3. Using the result of problem 2, show that in the $M/G/1$ case, the squared coefficient of variation of the interdeparture time is given by

$$C_D^2 = 1 + \rho^2(C_B^2 - 1),$$

where C_B^2 is the squared coefficient of variation of the service time distribution.

4. * Show that the interdeparture time distribution of an $M/M/m$ FCFS queue is exponential. To simplify the problem, first consider an $M/M/2$ queue. Let D_i denote the interdeparture time conditioned on the number of jobs in the system $N = i$. Then show that $D_i \sim \text{EXP}(2\mu)$ for $i \geq 2$. Next show that

$$L_{D_1}(s) = \frac{\lambda \cdot 2 \cdot \mu}{(s + \lambda + \mu)(s + 2\mu)} + \frac{\mu}{s + \lambda + \mu}$$

and

$$L_{D_0}(s) = \frac{\lambda}{s + \lambda} L_{D_1}(s)$$

(a tree diagram may be helpful here). Then obtain the required result using the theorem of total Laplace transforms. The generalization to the $M/M/m$ case proceeds in a similar fashion.

9.2 OPEN QUEUING NETWORKS

The argument given in the previous section for the product-form solution of tandem queues can be generalized to any feedforward network of Markovian queues (in which a job may not return to previously visited nodes) that is fed from independent Poisson sources. Jackson [JACK 1957] showed that the product-form solution also applies to open networks of Markovian queues with feedback. Besides requiring that the distributions of job interarrival times and service times at all nodes be exponential, assume that the scheduling discipline at each node is FCFS.

First we consider several examples illustrating Jackson's technique.

Example 9.2

Consider the simple model of a computer system shown in Figure 9.4a. Jackson's result is that two queues will behave like independent $M/M/1$ queues, and hence

$$p(k_0, k_1) = (1 - \rho_0)\rho_0^{k_0}(1 - \rho_1)\rho_1^{k_1}, \quad (9.3)$$

where $\lambda_0/\mu_0 = \rho_0$ and $\lambda_1/\mu_1 = \rho_1$. (where $\rho_0, \rho_1 < 1$ for stability.) To apply this result, we have to compute the average arrival rates λ_0 and λ_1 into the two nodes. Note that in the steady state the departure rates from the two nodes will also be λ_0 and λ_1 , respectively. Arrivals to the CPU node occur either from the outside world at the rate λ or from the I/O node at the rate λ_1 . The total arrival rate to the CPU node is therefore $\lambda_0 = \lambda + \lambda_1$. Given that a job just completed a CPU burst, it will next request I/O service with probability p_1 . Therefore, the average arrival rate to the I/O node is given by $\lambda_1 = \lambda_0 p_1$. Thus

$$\lambda_0 = \frac{\lambda}{1 - p_1} = \frac{\lambda}{p_0} \quad (9.4)$$

and

$$\lambda_1 = \frac{p_1 \lambda}{p_0}. \quad (9.5)$$

This implies that

$$\rho_0 = \frac{\lambda}{p_0 \mu_0} \quad \text{and} \quad \rho_1 = \frac{p_1 \lambda}{p_0 \mu_1}.$$

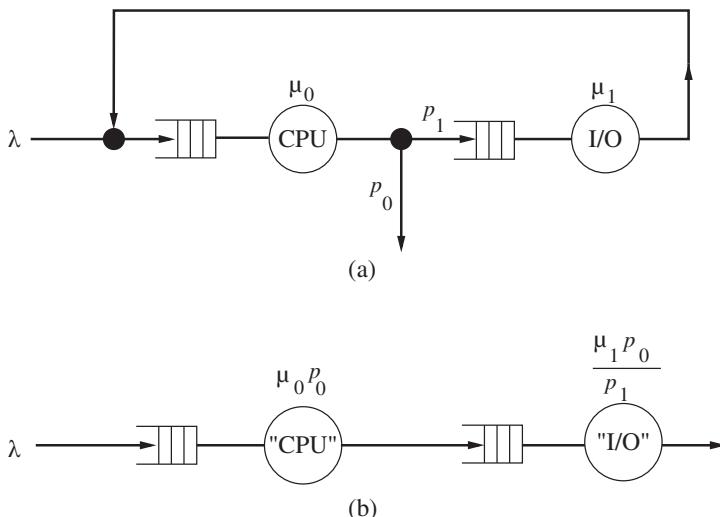


Figure 9.4. (a) An open network with feedback; (b) an “equivalent” network without feedback

If we let B_0 denote the total CPU service requirement of a program, then $E[B_0] = 1/(p_0\mu_0)$. Similarly, $E[B_1] = p_1/(p_0\mu_1)$ denotes the expected value of the total service time required on the I/O device for a typical program. If $\rho_0 > \rho_1$ (i.e., $E[B_0] > E[B_1]$), then the CPU is the bottleneck, in which case the system is said to be CPU-bound. Similarly, if $\rho_0 < \rho_1$, then the system is I/O-bound.

The average response time may be computed by summing the average number of jobs at the two nodes and then using Little's formula:

$$E[R] = \left(\frac{\rho_0}{1 - \rho_0} + \frac{\rho_1}{1 - \rho_1} \right) \frac{1}{\lambda}$$

or

$$\begin{aligned} E[R] &= \frac{1}{p_0\mu_0 - \lambda} + \frac{1}{\frac{p_0\mu_1}{p_1} - \lambda} \\ &= \frac{E[B_0]}{1 - \lambda E[B_0]} + \frac{E[B_1]}{1 - \lambda E[B_1]}. \end{aligned} \quad (9.6)$$

It is easily seen that this formula also gives the average response time of the “unfolded” tandem network shown in Figure 9.4b. The service rate of the “equivalent” CPU in this system is $\mu_0 p_0$. Thus, a job requests an uninterrupted average CPU time equal to $E[B_0] = 1/(\mu_0 p_0)$ in the equivalent system, while in the original system a job requires, on the average, $1/p_0$ CPU bursts of average time $1/\mu_0$ each. Therefore, to determine $E[R]$ and $E[N]$, it is sufficient to know only the aggregate resource requirements of a job; in particular, details of the pattern of resource usage are not important for computing these average values. We caution the reader that the equivalence between the networks of Figures 9.4a and 9.4b does not hold with respect to the distribution function $F_R(x)$ of the response time. Computation of response time distribution is difficult even for Jacksonian networks without feedback [SIMO 1979]. At the end of this chapter, we shall illustrate the computation of response time distribution for these two networks.

It is instructive to solve the network in Figure 9.4a by directly analyzing the stochastic process whose states are given by pairs (k_0, k_1) , $k_0 \geq 0, k_1 \geq 0$. By the assumption of exponentially distributed interevent times, the process is a homogeneous CTMC with the state diagram shown in Figure 9.5. For a state (k_0, k_1) with $k_0 > 0, k_1 > 0$, the steady state balance equation is obtained by equating the rates of flow into and out of the state:

$$\begin{aligned} (\lambda + \mu_0 + \mu_1)p(k_0, k_1) &= \lambda p(k_0 - 1, k_1) + \mu_0 p_1 p(k_0 + 1, k_1 - 1) \\ &\quad + \mu_0 p_0 p(k_0 + 1, k_1) + \mu_1 p(k_0 - 1, k_1 + 1). \end{aligned}$$

Similarly

$$\begin{aligned} (\lambda + \mu_0)p(k_0, 0) &= \lambda p(k_0 - 1, 0) + \mu_0 p_0 p(k_0 + 1, 0) + \mu_1 p(k_0 - 1, 1), k_0 > 0, \\ (\lambda + \mu_1)p(0, k_1) &= \mu_0 p_0 p(1, k_1) + \mu_0 p_1 p(1, k_1 - 1), k_1 > 0, \\ \lambda p(0, 0) &= \mu_0 p_0 p(1, 0). \end{aligned}$$

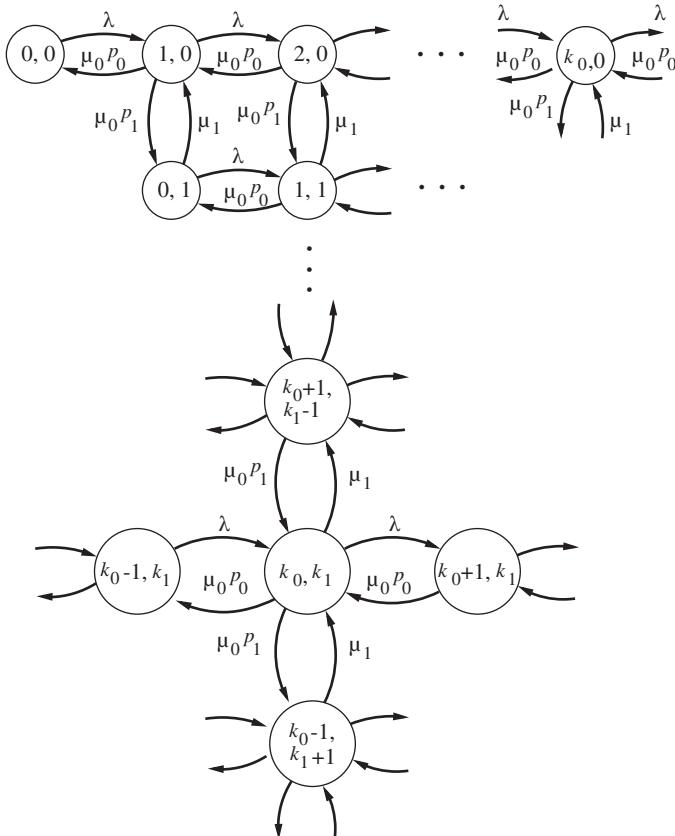


Figure 9.5. State diagram for the network in Figure 9.4a

Also

$$\sum_{k_0, k_1} p(k_0, k_1) = 1.$$

It may be verified by direct substitution that the solution (9.3) indeed satisfies these equations.

‡

Example 9.3

Consider the (open) central server queuing model of a computer system shown in Figure 9.6. Let us trace the path of a tagged program that just arrived. Temporarily ignoring queuing delays, the program will occupy one of $m + 1$ nodes at a time. Assume that the request for I/Oi occurs at the end of a CPU burst with probability p_i , independent of the past history of the tagged program. We can model the behavior of a tagged program as a homogeneous discrete-time Markov chain as in Example 7.20; the corresponding state diagram is given in Figure 7.25. [Note that we have

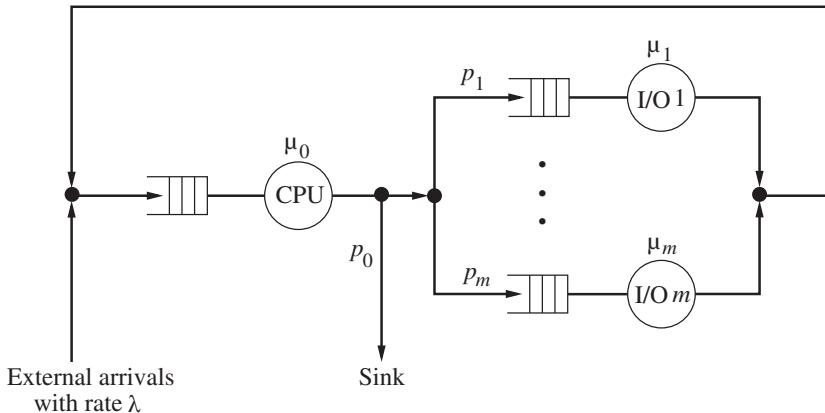


Figure 9.6. The open central server network

added an absorbing state, labeled $m + 1$ (or STOP).] The transition probability matrix of this DTMC is given by

$$P = \left[\begin{array}{cccc|c} 0 & p_1 & \cdots & p_m & p_0 \\ 1 & 0 & & 0 & 0 \\ 1 & 0 & & 0 & 0 \\ 1 & \cdot & & \cdot & \cdot \\ 1 & \cdot & & \cdot & \cdot \\ 1 & \cdot & & \cdot & \cdot \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 \end{array} \right].$$

As discussed in Chapter 7 (Section 7.9), the boxed portion of this matrix, denoted here by X , is of interest:

$$X = \left[\begin{array}{ccccc} 0 & p_1 & p_2 & \cdots & p_m \\ 1 & 0 & 0 & & 0 \\ 1 & \cdot & \cdot & & 0 \\ 1 & \cdot & \cdot & & 0 \\ 1 & \cdot & \cdot & & 0 \\ 1 & 0 & 0 & & 0 \end{array} \right].$$

X is known as the *routing matrix* of the queuing network of Figure 9.6. Analyzing the behavior of the tagged program, we are able to obtain the average number of visits (or visit counts) V_j as in Example 7.20 (we assume that $p_0 \neq 0$):

$$V_j = \begin{cases} \frac{1}{p_0}, & j = 0, \\ \frac{p_j}{p_0}, & j = 1, 2, \dots, m. \end{cases}$$

In other words, the average number of visits made to node j by a typical program is p_j/p_0 ($j \neq 0$) and $1/p_0$ ($j = 0$). Since λ programs per unit time enter the network on the average, the overall rate of arrivals, λ_j , to node j is then given by

$$\lambda_j = \begin{cases} \frac{1}{p_0} \lambda, & j = 0, \\ \frac{p_j}{p_0} \lambda, & j = 1, 2, \dots, m. \end{cases}$$

The utilization, ρ_j , of node j is given by $\rho_j = \lambda_j / \mu_j = \lambda V_j / \mu_j$ and we assume that $\rho_j < 1$ for all j . Jackson has shown that the steady-state joint probability of k_j customers at node j ($j = 0, 1, \dots, m$) is given by

$$p(k_0, k_1, k_2, \dots, k_m) = \prod_{j=0}^m p_j(k_j). \quad (9.7)$$

This formula implies that the queue lengths are mutually independent in the steady state, and the steady-state probability of k_j customers at node j is given by the $M/M/1$ formula:

$$p_j(k_j) = (1 - \rho_j) \rho_j^{k_j}.$$

The validity of this product-form solution can be established using the direct approach as in Examples 9.1 and 9.2. We leave this as an exercise.

The form of the joint probability (9.7) can mislead a reader to believe that the traffic along the arcs consists of Poisson processes. The reader is urged to solve problem 3 at the end of this section to realize that the input process to a service center in a network with feedback is not Poissonian in general [BEUT 1978, BURK 1976]. This is why Jackson's result is remarkable.

The average queue length, $E[N_j]$, and the average response time, $E[R_j]$, of node j (accumulated over all visits) are given by

$$E[N_j] = \frac{\rho_j}{1 - \rho_j} \quad \text{and} \quad E[R_j] = \frac{1}{\lambda} \frac{\rho_j}{1 - \rho_j}.$$

From these, the average number of jobs in the system and the average response time are computed to be

$$E[N] = \sum_{j=0}^m \frac{\rho_j}{1 - \rho_j}$$

and

$$\begin{aligned} E[R] &= \frac{1}{\lambda} \sum_{j=0}^m \frac{\rho_j}{1 - \rho_j} \\ &= \frac{1/(p_0 \mu_0)}{1 - \lambda/(p_0 \mu_0)} + \sum_{j=1}^m \frac{p_j / (p_0 \mu_j)}{1 - \lambda p_j / (\mu_j p_0)} \\ &= \frac{1}{\mu_0 p_0 - \lambda} + \sum_{j=1}^m \frac{1}{(\frac{p_0 \mu_j}{p_j} - \lambda)} \\ &= \frac{1}{\frac{\mu_0}{V_0} - \lambda} + \sum_{j=1}^m \frac{1}{\frac{\mu_j}{V_j} - \lambda} = \sum_{j=0}^m \frac{E[B_j]}{1 - \lambda E[B_j]}, \end{aligned} \quad (9.8)$$

where the total service requirement on device j , on the average, is given by $E[B_j] = V_j/\mu_j$. This last formula for the average response time is a generalized version of formula (9.6). It also affords an “unfolded” interpretation of the queuing network of Figure 9.6, in the same way as Figure 9.4b is the “unfolded” version of the network in Figure 9.4a. At the end of this chapter, we shall illustrate the computation of response time distribution for this example.

‡

Jackson’s result applies in even greater generality. Consider an open queuing network with $(m + 1)$ nodes, where the i th node consists of c_i exponential servers each with mean service time of $1/\mu_i$ seconds. External Poisson sources contribute γ_i jobs/second to the arrival rate of the i th node so that the total external arrival rate $\lambda = \sum_{i=0}^m \gamma_i$. If we let $q_i = \gamma_i/\lambda$, $i = 0, 1, \dots, m$ so that $\sum_{i=0}^m q_i = 1$, then a job will first enter the network at node i , with probability q_i . On service completion at node i , a job next requires service at node j with probability x_{ij} or completes execution with probability $1 - \sum_{j=0}^m x_{ij}$.

First, we analyze the behavior of a tagged job through the network. This behavior can be modeled as a homogeneous discrete-time Markov chain as in Example 7.20. Equation (7.79) is applicable here in computing V_i , the average number of visits made by the tagged program to node i . Therefore

$$V_i = q_i + \sum_{k=0}^m x_{ki} V_k, \quad i = 0, 1, \dots, m.$$

Now the average job arrival rate to node i is obtained by multiplying V_i by λ , the average job arrival rate to the network. Thus

$$\lambda_i = \lambda V_i = \lambda q_i + \sum_{k=0}^m x_{ki} \lambda V_k.$$

Noting that $\lambda q_i = \gamma_i$ and $\lambda V_k = \lambda_k$, this expression simplifies to

$$\lambda_i = \gamma_i + \sum_{k=0}^m \lambda_k x_{ki}, \quad i = 0, 1, \dots, m. \quad (9.9)$$

This system of equations (known as “traffic equations”) has a unique solution if we assume that there is at least one node j such that $\gamma_j > 0$ and that the matrix power series:

$$\sum_{k=0}^n X^k$$

converges as n approaches infinity (see Section 7.9). This implies that after a certain number of visits to various service centers there is a positive probability that a job will depart from the system. Jackson’s theorem states that

each node behaves like an independent $M/M/c_i$ queue and, therefore, the steady-state probability of k_i customers at node i , $i = 0, 1, \dots, m$ is given by the product form

$$p(k_0, k_1, \dots, k_m) = p_0(k_0) \cdots p_m(k_m), \quad (9.10)$$

where $p_i(k_i)$ is the steady-state probability of finding k_i jobs in an $M/M/c_i$ queue with arrival rate λ_i and average service time $1/\mu_i$ for each of the c_i servers. Thus, using equations (8.45) and (8.46), we have

$$p_i(k_i) = \begin{cases} \frac{(\lambda_i/\mu_i)^{k_i}}{k_i!} p_i(0), & 1 \leq k_i < c_i \\ \frac{(\lambda_i/\mu_i)^{k_i} p_i(0)}{c_i! c_i^{k_i - c_i}}, & k_i \geq c_i, \end{cases}$$

where

$$p_i(0) = \frac{1}{\sum_{k_i=0}^{c_i-1} \frac{(\lambda_i/\mu_i)^{k_i}}{k_i!} + \frac{(\lambda_i/\mu_i)^{c_i}}{c_i!} \frac{1}{1 - \frac{\lambda_i}{c_i \mu_i}}}.$$

Problems

- Consider the open central server queuing model with two I/O channels with a common service rate of 1.2 s^{-1} . The CPU service rate is 2 s^{-1} , and the arrival rate is $1/7$ job per second. The branching probabilities are given by $p_0 = 0.1$, $p_1 = 0.3$, and $p_2 = 0.6$. Determine steady-state probabilities, assuming that all service times are independent exponentially distributed random variables. Determine the queue length pmf at each node as well as the average response time from the source to the sink.
- Consider a variation of the queuing model of Figure 9.4a, where the CPU node consists of two parallel processors with a service rate of μ_0 each. Draw a state diagram for this system and proceed to solve the balance equations. Obtain an expression for the average response time $E[R]$ as a function of μ_0, μ_1, p_0 , and λ . Now compare your answer with that obtained using Jackson's result.
- *Refer to Burke [BURK 1976]. Consider the $M/M/1$ FCFS queue with feedback as shown in Figure 9.P.1. By Jackson's theorem, it is easy to derive the steady-state pmf of the number of jobs N in the system:

$$P(N = i) = \left(1 - \frac{\lambda}{\mu p}\right) \left(\frac{\lambda}{\mu p}\right)^i, \quad i = 0, 1, \dots$$

We wish to show that the actual input process (which is a merger of the external arrival process and the feedback process) is not Poissonian, even though the

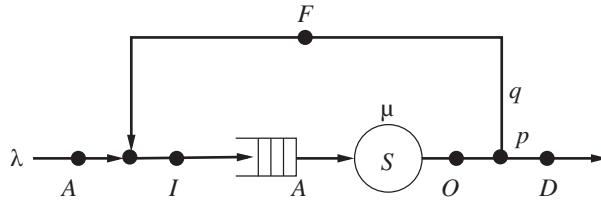


Figure 9.P.1. $M/M/1$ queue with Bernoulli feedback

exogenous arrival process and the departure process are both Poissonian with rate λ . Proceed by first observing that the complementary distribution function of the interinput times is given by

$$R_I(t) = e^{-\lambda t} R_Y(t),$$

where Y is the time to the next feedback as measured from the time of the last input to the queue-server pair. Next obtain the density of Y as

$$f_Y(t) = \mu q e^{-(\mu - \lambda)t};$$

hence, show that

$$R_Y(t) = \frac{\mu q}{\mu - \lambda} e^{-(\mu - \lambda)t} + \frac{\mu p - \lambda}{\mu - \lambda}.$$

From this, conclude that I is hyperexponentially distributed. In order to derive the Laplace-Stieltjes transform of the interdeparture time D , proceed by computing the conditional LST $L_{D|N_d=i}(s)$, where N_d is the number of jobs left in the system by a departing job. Using the result of problem 5 in Section 7.7, show that

$$P(N_d = i) = P(N = i),$$

and hence the unconditional LST of D is obtained as:

$$L_D(s) = \frac{\lambda}{s + \lambda}.$$

This verifies that the departure process is Poissonian.

9.3 CLOSED QUEUING NETWORKS

One of the implicit assumptions behind the model of Example 9.3 is that immediately on its arrival, a job is scheduled into main memory and is able to compete for active resources such as the CPU and the I/O channels. In practice, the number of main-memory partitions will be limited, which implies the existence of an additional queue, called the **job-scheduler queue** (see Figure 9.7). However, such a network is said to involve multiple resource holding. This is because a job can simultaneously hold main memory and

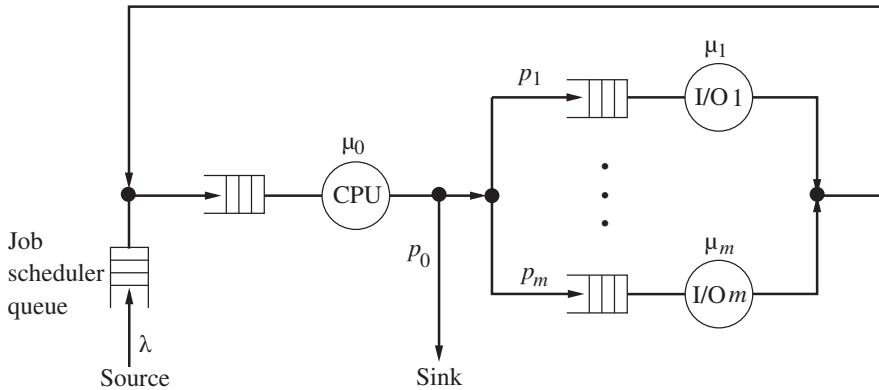


Figure 9.7. An open central server network with a blocking

an active device. Such a network cannot be solved by product-form methods [CHAN 1978]. Nevertheless, an approximate solution to such a network is provided by the methods of the last section. If the external arrival rate λ is low, then the probability that a job has to wait in the scheduler queue will be low, and we expect the solution of Example 9.3 to be quite good. Thus, the model of Example 9.3 is a light-load approximation to the model of Figure 9.7. Let us now take the other extreme and assume a large value of λ , so that the probability that there is at least one customer in the job-scheduler queue is very high.

We may then assume that the departure of a job from the active set immediately triggers the scheduling of an already waiting job into main memory. Thus, the closed network of Figure 9.8 will be a “good” approximation to the system of Figure 9.7, under heavy-load conditions. Each job circulating in this closed network is said to be an active job and must be allocated a partition

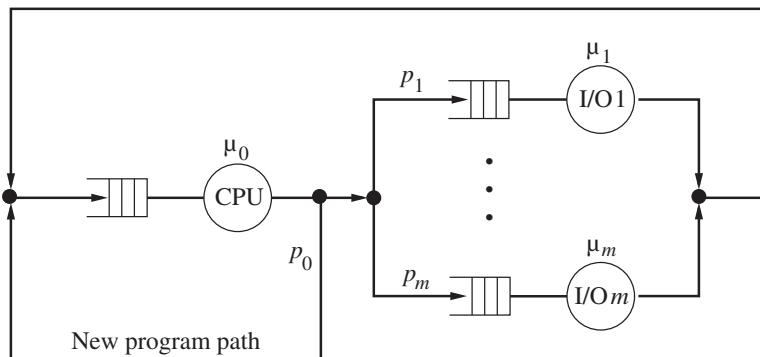


Figure 9.8. The (closed) central server model

of main memory. The total number of active jobs is called **the degree (or level) of multiprogramming**.

Many efficient algorithms for calculating performance measures of product-form closed queuing networks have been developed. Two most important algorithms, namely, the convolution algorithm and mean-value analysis (MVA) algorithm, are described in this section. The former algorithm is an efficient iterative technique for calculating the normalization constant. Once the normalization constant is computed, the system performance measures of interest can be easily derived. In the following examples, we introduce the normalization constant and then give an efficient algorithm to compute it.

Example 9.4

Let us return to the cyclic queuing model studied in the last chapter (Example 8.8), which is shown in Figure 9.9. Here, we choose to represent the state of the system by a pair (k_0, k_1) , where k_i denotes the number of jobs at node i ($i = 0, 1$). Recall that $k_0 + k_1 = n$, the degree of multiprogramming. Unlike the two-node open queuing network (Figure 9.4a), the state space in this case is finite. The dot pattern on the (k_0, k_1) plane of Figure 9.10 represents the infinite state space of the open network (Figure 9.4a), while the dot pattern on the line $k_0 + k_1 = n$ is the finite state space of the cyclic (closed) queuing network studied here.

The state diagram for the cyclic queuing model is shown in Figure 9.11. The steady state balance equations are given by

$$\begin{aligned} (\mu_1 + \mu_0 p_1)p(k_0, k_1) &= \mu_0 p_1 p(k_0 + 1, k_1 - 1) + \mu_1 p(k_0 - 1, k_1 + 1), \quad k_0, k_1 > 0, \\ \mu_1 p(0, n) &= \mu_0 p_1 p(1, n - 1), \\ \mu_0 p_1 p(n, 0) &= \mu_1 p(n - 1, 1). \end{aligned}$$

If we let $\rho_0 = a/\mu_0$ and $\rho_1 = ap_1/\mu_1$, where a is an arbitrary constant, then we can verify by direct substitution that the steady-state probability $p(k_0, k_1)$ has the following product form:

$$p(k_0, k_1) = \frac{1}{C(n)} \rho_0^{k_0} \rho_1^{k_1}.$$

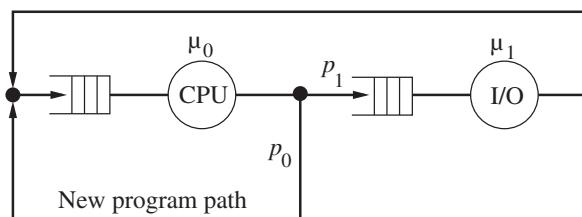


Figure 9.9. The closed cyclic queuing model

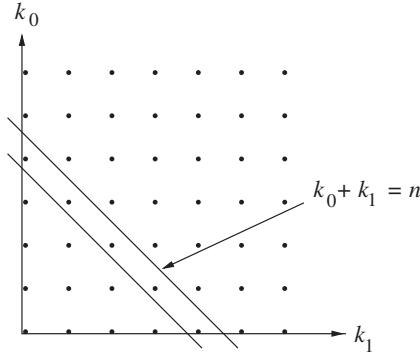


Figure 9.10. State spaces for two-node networks

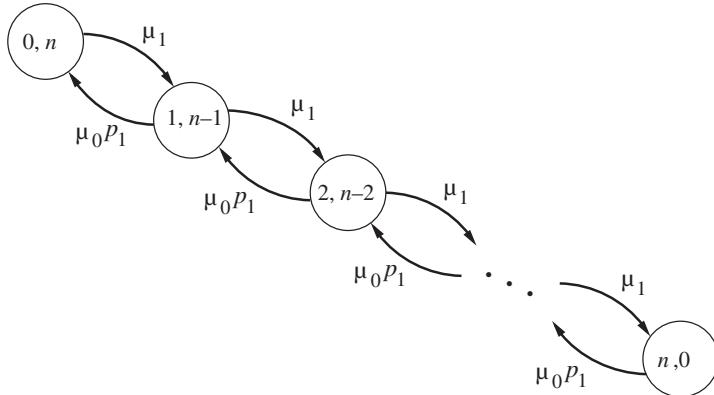


Figure 9.11. The state diagram for the closed cyclic queuing model

The normalizing constant $C(n)$ is chosen so that

$$\sum_{\substack{k_0+k_1=n \\ k_0, k_1 \geq 0}} p(k_0, k_1) = 1.$$

The choice of the constant a is quite arbitrary in that the value of $p(k_0, k_1)$ will not change with a , although the intermediate values ρ_0, ρ_1 , and $C(n)$ will depend on a . If we define $\lambda_0 = a$ and $\lambda_1 = ap_1$, we may interpret the vector (λ_0, λ_1) as the relative throughputs of the corresponding nodes. Then $\rho_0 = (\lambda_0/\mu_0)$ and $\rho_1 = (\lambda_1/\mu_1)$ are interpreted as relative utilizations. Two popular choices of the constant a are $a = 1$ and $a = \mu_0$. Choosing $a = \mu_0$, we have $\rho_0 = 1$ and $\rho_1 = \mu_0 p_1 / \mu_1$. Also

$$p(k_0, k_1) = \frac{1}{C(n)} \rho_1^{k_1}.$$

Using the normalization condition, we get

$$1 = \frac{1}{C(n)} \sum_{k_1=0}^n \rho_1^{k_1} = \frac{1}{C(n)} \frac{1 - \rho_1^{n+1}}{1 - \rho_1}$$

or

$$C(n) = \begin{cases} \frac{1 - \rho_1^{n+1}}{1 - \rho_1}, & \rho_1 \neq 1 \\ n + 1, & \rho_1 = 1. \end{cases}$$

Now the CPU utilization U_0 may be expressed as

$$U_0 = 1 - p(0, n) = 1 - \frac{\rho_1^n}{C(n)},$$

$$U_0 = \begin{cases} \frac{1 - \rho_1^n}{1 - \rho_1^{n+1}}, & \rho_1 \neq 1 \\ \frac{n}{n+1}, & \rho_1 = 1. \end{cases}$$

This agrees with the solution obtained in the last chapter [equation (8.62)]. The average throughput is given by

$$E[T] = \mu_0 U_0 p_0.$$

#

Example 9.5

Consider the (closed) central server network shown in Figure 9.8. The state of the network is given by an $(m + 1)$ -tuple, (k_0, k_1, \dots, k_m) where $k_i \geq 0$ is the number of jobs at server i (including any in service). Since the number of jobs in a closed network is fixed, we must further impose the constraint $\sum_{i=0}^m k_i = n$ on every state. Thus the state space of the network is finite, as the number of states is equal to the number of partitions of n objects among $m + 1$ cells. You were asked to compute this number in problem 5 in section 1.12:

$$\binom{n+m}{m} = \frac{(n+m)!}{n! m!}. \quad (9.11)$$

If we assume that service times at all servers are exponentially distributed, the stochastic process modeling the behavior of the network is a finite-state homogeneous continuous-time Markov chain, which can be shown to be irreducible and recurrent nonnull (assuming that $0 < p_i < 1, i = 0, 1, \dots, m$). In principle, therefore, we can write down the steady-state balance equations and obtain the unique steady-state probabilities. However, the number of equations in this system will be equal to the number of states given by expression (9.11). This is a formidable number of states, even for relatively small values of n and m . Fortunately, Gordon and Newell [GORD 1967] have shown that such Markovian closed networks possess relatively simple product-form solutions.

In order to use their technique, we first analyze the behavior of a tagged program, ignoring all queues in the network. The movement of a tagged program through the network can be modeled by a homogeneous discrete-time Markov chain with $m + 1$ states. The transition probability matrix X of this Markov chain is given by

$$X = \begin{bmatrix} p_0 & p_1 & \cdots & p_m \\ 1 & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

The DTMC is finite, and if we assume that $0 < p_i < 1$ for all i , then it can be shown to be irreducible and aperiodic. Then the unique steady-state probability vector $\mathbf{v} = (v_0, v_1, \dots, v_m)$ can be obtained by solving the system of equations

$$\mathbf{v} = \mathbf{v}X \quad (9.12)$$

and

$$\sum_{i=0}^m v_i = 1. \quad (9.13)$$

If we observe the system for a real-time interval of duration τ , then $v_i\tau$ can be interpreted to be the average number of visits to node i in the interval. If we remove the normalization condition (9.13), then the v_i terms cannot be interpreted as probabilities, but $a v_i \tau$ will still yield the average number of visits to node i ($i = 0, 1, \dots, m$) for some fixed constant a . In this sense, v_i can be thought of as the relative visit count for node i , and thus v_i is sometimes called the *relative arrival rate* or the *relative throughput* of node i .

For the central server network, equation (9.12) becomes

$$\begin{aligned} v_0 &= v_0 p_0 + \sum_{i=1}^m v_i, \\ v_i &= v_0 p_i, \quad i = 1, 2, \dots, m. \end{aligned} \quad (9.14)$$

Only m out of these $(m + 1)$ equations are independent; therefore [in absence of the normalization condition (9.13)], v_0 can be chosen as any real value that will aid us in our computations. The usual choices of v_0 are $1/p_0$, μ_0 , and 1.

If we choose $v_0 = 1/p_0$, then from (9.14), we have $v_i = p_i/p_0$ ($i = 1, 2, \dots, m$). Bear in mind that the closed central server model is intended to be an approximation to the open model of Figure 9.7. It follows from our analysis of a tagged program for the open network of Figure 9.6 that with this choice of v_0 , $v_i = V_i$, where V_i ($i = 0, 1, \dots, m$) is the average number of visits a typical program makes to node i in order to complete its execution. Let the relative utilization of device i be given by $\rho_i = v_i/\mu_i$. If we let B_i be the total service requirement of a program on device i , then in this case ρ_i equals the expected value of the total service requirement $E[B_i]$ on device i . If we choose $v_0 = \mu_0$, then $\rho_0 = 1$; hence all device utilizations are scaled by the CPU utilization. This choice is often more convenient computationally.

Gordon and Newell [GORD 1967] have shown that the steady-state probability $p(k_0, k_1, \dots, k_m)$ of finding k_i jobs at nodes $i, i = 0, 1, \dots, m$, is given by

$$\begin{aligned} p(k_0, k_1, \dots, k_m) &= \frac{1}{C(n)} \prod_{i=0}^m \rho_i^{k_i} \\ &= \frac{1}{C(n)} \prod_{i=0}^m \left(\frac{v_i}{\mu_i} \right)^{k_i}. \end{aligned} \quad (9.15)$$

Here, the normalization constant $C(n)$ is evaluated using the condition:

$$\sum_{k \in I} p(k_0, k_1, \dots, k_m) = 1,$$

where the state space

$$I = \{(k_0, k_1, \dots, k_m) \mid k_i \geq 0 \text{ for all } i \text{ and } \sum_{i=0}^m k_i = n\}$$

contains $\binom{n+m}{m}$ states.

#

More generally, consider an arbitrary closed queuing network having exponentially distributed service time with respective rates μ_i ($i = 0, 1, \dots, m$) and the routing matrix $X = [x_{ij}]$. As in Example 9.5, we first remove all queues and model the behavior of a tagged program through the network. Since our only concern at this point is to count the average number of visits to device i , the behavior of the tagged program is then captured by a homogeneous, finite-state DTMC with transition probability matrix X . We will assume that this chain is irreducible and aperiodic. Therefore, unlike the case of an open network, the routing matrix of a closed network is a stochastic matrix. As in Example 9.5, we can obtain the relative throughputs, v_i terms, by solving the system of linear equations [analogous to equation (9.12)]:

$$v_i = \sum_{j=0}^m v_j x_{ji}, i = 0, 1, \dots, m. \quad (9.16)$$

Again, since X is a stochastic matrix, only m out of the above $m + 1$ equations are independent, and the system of equations has a unique solution, up to a multiplying constant. Therefore, one of the components of v can be chosen arbitrarily. As before, common choices for v_0 are V_0 , μ_0 , and 1.

If v_0 is chosen to be equal to the average number of visits V_0 to node 0 per program, then the relative utilization ρ_i is equal to $V_i/\mu_i = E[B_i]$, the expected value of the total service requirement imposed by a typical program on the i th node. As we shall see below, U_i , the real utilization

of node i , is a function only of the relative utilizations $\rho_0, \rho_1, \dots, \rho_m$, and from the real utilization of device i , the average system throughput is computed as $U_i/E[B_i]$. Alternatively, we will see that in this case average system throughput equals the ratio $C(n-1)/C(n)$, which again depends only on $\rho_i, i = 0, 1, \dots, m$. Thus, measures of system performance such as device utilizations and average system throughput can be obtained from a specification of the $(m+1)$ service requirements, $E[B_0], E[B_1], \dots, E[B_m]$. In particular, the topology of the network, the routing probabilities x_{ij} , and the individual service rates μ_i need not be specified. Also, equation (9.16) need not be solved for the relative throughputs v_i . Such an interpretation is convenient since the quantities $E[B_i]$ are readily estimated from measured data [DENN 1978].

The reader is urged to verify in problems 1–3 at the end of this section that the choice of v_0 will not affect the performance measures of interest, although it will affect the values of intermediate quantities such as ρ_i and $C(n)$.

Continuing with our analysis of the general closed network, we see that since there are $n > 1$ programs circulating through the network, the state of the network at any time will be denoted by (k_0, k_1, \dots, k_m) , where k_i is the number of jobs at node i . If $k_i = 0$, then device i is idle. If $k_i = 1$, a job is being processed by device i . If $k_i > 1$, a job is being processed by device i and $k_i - 1$ jobs are waiting to be served on device i . Since there are exactly n jobs in the system, we must further impose the restriction that $\sum_{i=0}^m k_i = n$. This implies, as before, that the state space, I , of the network contains $\binom{n+m}{m}$ states, specifically:

$$I = \{(k_0, k_1, \dots, k_m) \mid k_i \geq 0, \sum_{i=0}^m k_i = n\}.$$

By assumption of exponentially distributed service times, all interevent times are exponentially distributed, and thus the network can be modeled by a homogeneous CTMC. Since the chain is finite, if we assume that it is irreducible, then a unique steady-state probability vector exists that can be obtained as a solution of steady-state balance equations:

For any state $s = (k_0, k_1, \dots, k_m)$, the probability, $p(s)$, of being in that state times the rate of transition from that state has to be equal to the sum over all states t of $p(t)$ times the rate of transition from t to s . Therefore we have

$$\begin{aligned} & \sum_{j|k_j>0} \mu_j p(k_0, k_1, \dots, k_m) \\ &= \sum_{j|k_j>0} \sum_i x_{ij} \mu_i p(k_0, \dots, k_i + 1, \dots, k_j - 1, \dots, k_m). \end{aligned} \quad (9.17)$$

In other words, in steady state, the rate of flow out of a state must equal the rate of flow into that state.

As in Example 9.5, equation (9.17) has the following product-form solution [GORD 1967]:

$$p(k_0, k_1, \dots, k_m) = \frac{1}{C(n)} \prod_{i=0}^m \rho_i^{k_i}. \quad (9.18)$$

The normalization constant $C(n)$ can be computed using the fact that the probabilities sum to unity,

$$C(n) = \sum_{s \in I} \prod_{i=0}^m \rho_i^{k_i}, \quad (9.19)$$

where $s = (k_0, k_1, \dots, k_m)$.

Since the number of states of the network grows exponentially with the number of customers and the number of service centers, it is not feasible to evaluate $C(n)$ by direct summation as in equation (9.19), because the computation would be too expensive and perhaps numerically unstable. Nevertheless, it is possible to derive stable and efficient computational algorithms to obtain the value of the normalization constant $C(n)$ [BUZE 1973]. These algorithms also yield simple expressions for performance measures such as the average queue length, $E[N_i]$ and the utilization U_i of the i th server.

Consider the following polynomial in z [WILL 1976]:

$$\begin{aligned} G(z) &= \prod_{i=0}^m \frac{1}{1 - \rho_i z} \\ &= (1 + \rho_0 z + \rho_0^2 z^2 + \dots)(1 + \rho_1 z + \rho_1^2 z^2 + \dots) \dots \\ &\quad (1 + \rho_m z + \rho_m^2 z^2 + \dots). \end{aligned} \quad (9.20)$$

It is clear that the coefficient of z^n in $G(z)$ is equal to the normalization constant $C(n)$, since the coefficient is just the sum of all the terms of the form $\rho_0^{k_0} \rho_1^{k_1} \dots \rho_m^{k_m}$ with $\sum_{i=0}^m k_i = n$. In other words, $G(z)$ is the generating function of the sequence $C(1), C(2), \dots$

$$G(z) = \sum_{n=0}^{\infty} C(n) z^n, \quad (9.21)$$

where $C(0)$ is defined to be equal to unity. It should be noted that since $C(n)$ is not a probability, $G(z)$ is not a probability generating function and hence $G(1)$ is not necessarily equal to unity. In order to derive a recursive relation for computing $C(n)$, define

$$G_i(z) = \prod_{k=0}^i \frac{1}{1 - \rho_k z}, \quad i = 0, 1, \dots, m, \quad (9.22)$$

so that $G_m(z) = G(z)$. Also define $C_i(j)$ by

$$G_i(z) = \sum_{j=0}^{\infty} C_i(j)z^j, \quad i = 0, 1, \dots, m,$$

so that $C_m(j) = C(j)$. Observe that

$$G_0(z) = \frac{1}{1 - \rho_0 z} \tag{9.23}$$

and

$$G_i(z) = G_{i-1}(z) \frac{1}{1 - \rho_i z}, \quad i = 1, 2, \dots, m.$$

This last equation can be rewritten as

$$G_i(z)[1 - \rho_i z] = G_{i-1}(z)$$

or

$$G_i(z) = \rho_i z G_i(z) + G_{i-1}(z)$$

or

$$\sum_{j=0}^{\infty} C_i(j)z^j = \sum_{j=0}^{\infty} \rho_i z C_i(j)z^j + \sum_{j=0}^{\infty} C_{i-1}(j)z^j.$$

Equating the coefficients of z^j on both sides, we have a recursive formula for the computation of the normalization constant:

$$\begin{aligned} C_i(j) &= C_{i-1}(j) + \rho_i C_i(j-1), & i &= 1, 2, \dots, m, \\ j &= 1, 2, \dots, n. \end{aligned} \tag{9.24}$$

The initialization is obtained using (9.23) as

$$C_0(j) = \rho_0^j, \quad j = 0, 1, 2, \dots, n.$$

Also from (9.22), we have that the coefficient of z^0 in $G_i(z)$ is unity; hence

$$C_i(0) = 1, \quad i = 0, 1, \dots, m.$$

The convolution method for computing the normalization constant $C(n)$ is fully defined by equation (9.24). The computation of $C(n) = C_m(n)$ is illustrated in Table 9.1.

TABLE 9.1. Computation of the Normalization Constant $C_i(j)$

$j \backslash i$	0	1	2	...	$i - 1$	i	...	m
0	1	1	1	...	1	1	1	1
.	ρ_0	$\rho_0 + \rho_1$	$\rho_0 + \rho_1 + \rho_2$...		$\sum_{k=0}^i \rho_k$...	$C_m(1)$
.
$j - 1$	ρ_0^{j-1}	...				$C_i(j - 1)$		
							$\downarrow {}^* \rho_i$	
j	ρ_0^j	...			$C_{i-1}(j) \rightarrow$	$C_i(j)$		
.	.	.						
n	ρ_0^n	...						$C_m(n)$

As we will see shortly [formula (9.25)], only the last column of the $C_i(j)$ matrix of Table 9.1 is needed for the computation of the device utilizations. It is possible to avoid the storage of the $(n + 1) \times (m + 1)$ matrix suggested in the table. Because the matrix can be computed one column at a time, we need only store the column currently under computation. Assume a one-dimensional array $C[0 .. n]$ initialized to contain all zeros, except for $C[0]$, which is initialized to 1, and representing the current column of the $C_i(j)$ matrix. Also let $\rho[0 .. m]$ denote the vector of relative utilizations. Then the set of all $C(j)$ values may be computed using the following program segment.

Program 9.1 (Convolution Algorithm)

```
{initialize} C[0] := 1; for j := 1 to n do C[j] := 0;
for i := 0 to m do
  for j := 1 to n do
    C[j] := C[j] + rho[i] * C[j - 1].
```

Next, let us derive an expression for $U_i(n)$, the utilization of the i th device. Consider a slight modification to the generating function $G(z)$, denoted by $H_i(z)$:

$$\begin{aligned}
H_i(z) &= \left(\prod_{\substack{j=0 \\ j \neq i}}^m \frac{1}{1 - \rho_j z} \right) \left(\frac{1}{1 - \rho_i z} - 1 \right) \\
&= (1 + \rho_0 z + \rho_0^2 z^2 + \dots) \cdots (1 + \rho_{i-1} z + \rho_{i-1}^2 z^2 + \dots) \\
&\quad \cdot (\rho_i z + \rho_i^2 z^2 + \dots) (1 + \rho_{i+1} z + \rho_{i+1}^2 z^2 + \dots) \\
&\quad \cdots (1 + \rho_m z + \rho_m^2 z^2 + \dots).
\end{aligned}$$

The difference between $H(z)$ and $G(z)$ is that we have omitted the first term in the factor corresponding to the i th device. As a result, the coefficient of z^n in $H(z)$ will be the sum of all terms

$$\rho_0^{k_0} \cdots \rho_i^{k_i} \cdots \rho_m^{k_m}$$

such that $k_i \geq 1$. From (9.18) we then see that the coefficient of z^n in $H(z)$ divided by the coefficient of z^n in $G(z)$ must yield the marginal probability $P(N_i \geq 1)$, which is exactly the utilization $U_i(n)$. Now

$$H(z) = G(z) \frac{\frac{1}{1 - \rho_i z} - 1}{\frac{1}{1 - \rho_i z}} = G(z) \rho_i z.$$

Thus, the coefficient of z^n in $H(z)$ is simply ρ_i times the coefficient of z^{n-1} in $G(z)$. Therefore, we get

$$U_i(n) = \frac{\rho_i C(n-1)}{C(n)}. \quad (9.25)$$

From this formula we see that $U_i(n)/U_j(n) = \rho_i/\rho_j$, which explains the reason for calling ρ_i “relative utilizations.”

By a similar argument, we can obtain an expression for the probability that there are k or more jobs at node i :

$$P(N_i \geq k) = \frac{\rho_i^k C(n-k)}{C(n)}. \quad (9.26)$$

To get an expression for the average queue length at node i , as a function of the degree of multiprogramming n , observe that

$$\begin{aligned}
E[N_i(n)] &= \sum_{k=1}^n kP(N_i = k) \\
&= \sum_{k=1}^n k[P(N_i \geq k) - P(N_i \geq k+1)] \\
&= \sum_{k=1}^n kP(N_i \geq k) - \sum_{k=1}^n kP(N_i \geq k+1) \\
&= \sum_{k=1}^n kP(N_i \geq k) - \sum_{j=2}^{n+1} (j-1)P(N_i \geq j) \\
&= \sum_{k=1}^n kP(N_i \geq k) - \sum_{j=2}^{n+1} jP(N_i \geq j) + \sum_{j=2}^{n+1} P(N_i \geq j) \\
&= P(N_i \geq 1) - (n+1)P(N_i \geq n+1) + \sum_{j=2}^{n+1} P(N_i \geq j) \\
&= \sum_{j=1}^n P(N_i \geq j)
\end{aligned}$$

since $P(N_i \geq n+1) = 0$. Now, using the expression (9.26), we have

$$E[N_i(n)] = \frac{1}{C(n)} \sum_{j=1}^n \rho_i^j C(n-j). \quad (9.27)$$

Once again, in order to compute the average queue lengths, only the last column of the $C_j(i)$ matrix is needed.

Formula (9.27) leads us to an alternative recursive formula for the computation of $C(n)$. Observe that

$$\sum_{i=0}^m E[N_i(n)] = n,$$

so from (9.27) we get

$$n = \frac{1}{C(n)} \sum_{i=0}^m \sum_{j=1}^n \rho_i^j C(n-j)$$

and

$$C(n) = \frac{1}{n} \sum_{j=1}^n C(n-j) \left[\sum_{i=0}^m \rho_i^j \right] \quad (9.28)$$

with the initial condition $C(0) = 1$.

Formula (9.28) requires somewhat more arithmetic operations for its evaluation than does formula (9.24), but when many devices have equal ρ_i values and $m > n$, it will be more efficient to precompute the factors $(\sum_{i=0}^m \rho_i^j)$ above for each j and use formula (9.28).

Example 9.6

Consider a numerical instance of the central server model (Example 9.5) with the parameters as shown in Table 9.2.

We choose the relative throughput $v_0 = \mu_0 = \frac{1}{20}$ (per millisecond). Then from (9.14) we have, $v_1 = \mu_0 p_1$, and hence $v_1 = \frac{0.667}{20}$ and $v_2 = \frac{0.233}{20}$. The relative utilizations are then computed to be $\rho_0 = 1$, $\rho_1 = 1$, and $\rho_2 = 0.5$. The values of $C(1), C(2), \dots, C(10)$ are shown in Table 9.3. We compute the utilizations of the three nodes, using equation (9.25). Finally the average system throughput is computed by (in jobs per second):

$$E[T(n)] = \mu_0 p_0 U_0(n) = \frac{\mu_0 p_0 \rho_0 C(n-1)}{C(n)} = 5 \cdot \frac{C(n-1)}{C(n)}.$$

The average system throughput as a function of the degree of multiprogramming (degmul) is shown in Table 9.4 and is plotted in Figure 9.12.

‡

Figure 9.12 shows that as the degree of multiprogramming increases, the average throughput also increases. An increase in the degree of multiprogramming generally implies that additional main memory must be purchased. In the case of systems employing paged virtual memory, the degree of multiprogramming is not inherently limited by the size of the main memory, since a program is allowed to execute with only part of its address space in the main memory. For a fixed size of main memory, an increase in the degree of multiprogramming then implies a reduction in the page allotment per program;

TABLE 9.2. Parameters of Example 9.6

Parameter	Symbol	Value
Number of I/O channels	m	2
Degree of multiprogramming	n	1–10
Mean CPU time per burst	$1/\mu_0$	20 ms
Mean drum time per visit	$1/\mu_1$	30 ms
Mean disk time per visit	$1/\mu_2$	42.918 ms
Drum branching probability	p_1	0.667
Disk branching probability	p_2	0.233
Probability of job completion	p_0	0.1

TABLE 9.3. Computation of the Normalization Constant $C_i(j)$

$j \backslash i$	0	1	2
1	1	2	2.5
2	1	3	4.25
3	1	4	6.125
4	1	5	8.062
5	1	6	10.031
6	1	7	12.016
7	1	8	14.008
8	1	9	16.004
9	1	10	18.002
10	1	11	20.001

TABLE 9.4. Average System Throughput (Jobs Completed per Second)

<i>Degree of Multiprogramming n</i>	<i>Average system throughput (Jobs/s)</i>
1	2.0
2	2.9412
3	3.4694
4	3.7985
5	4.0187
6	4.1743
7	4.2889
8	4.3764
9	4.4450
10	4.5003

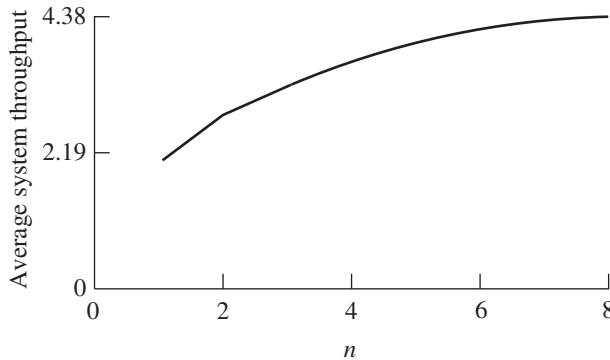


Figure 9.12. Average system throughput versus degree of multiprogramming

hence an increased frequency of page faults, which will tend to reduce average system throughput. On the other hand, an increased degree of multiprogramming implies that there is a greater chance of finding a job ready to run (on the CPU) whenever the currently executing job incurs a page fault. This will tend to increase the average throughput. The combined effect of these two conflicting factors on average system throughput is investigated in the next example.

Example 9.7

Consider a central server network with $m = 2$ I/O channels. The channel labeled 1 is a paging disk and the channel labeled 2 is a disk used for file I/O. Other parameters are specified in Table 9.5.

In this problem we set v_0 , the relative throughput of the CPU, to be equal to the average number of visits V_0 to the CPU per job. Then ρ_0 , the relative utilization

TABLE 9.5. Parameters for Example 9.7

Parameter	Symbol	Value
Number of I/O channels	m	2
Degree of multiprogramming	n	1–10
Mean total CPU time per job	$E[B_0]$	0.06667 s
Mean paging disk service time	$1/\mu_1$	10–100 ms
Mean disk service time	$1/\mu_2$	50 ms
Average number of paging disk requests (page faults) per job	$V_1(n)$	$0.1e^{(0.415n)}$
Average number of file I/O disk requests per job	V_2	5

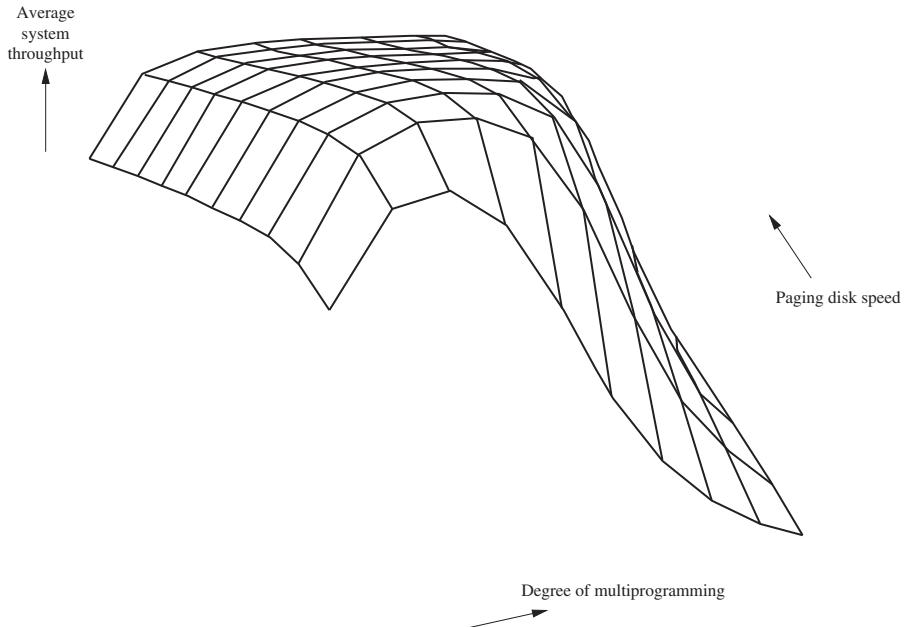


Figure 9.13. Demonstration of the thrashing phenomenon

of the CPU, equals $E[B_0]$, the average CPU service requirement per job. Note that V_0 is not even specified here. Also

$$\rho_1(n) = E[B_1] = \frac{V_1(n)}{\mu_1}$$

is an increasing function of n and

$$\rho_2 = E[B_2] = \frac{V_2}{\mu_2} = 0.25 \text{ s/job.}$$

Figure 9.13 is a three-dimensional plot showing the variation in average system throughput $E[T]$ with the degree of multiprogramming n and with the paging disk service rate μ_1 . For a fixed value of μ_1 , $E[T]$ first increases with n and, after reaching a maximum, starts to drop rather sharply. When n becomes rather large, the dramatic reduction in average system throughput, known as “thrashing”, occurs. The reduction in average system throughput can be compensated by an increase in the paging disk speed, which causes an increase in μ_1 , as shown in Figure 9.13. Alternative methods to control thrashing are to purchase more main memory or to improve program locality so that programs will page fault less frequently at a given allotment of main memory.

Example 9.8

The central server network discussed in the Example 9.5 may be interpreted from another viewpoint. Consider a system with n components, each with a powered failure rate μ_0 . Standby redundancy is used so that at most one component is in powered status while the remaining $n - 1$ components are either in a powered-off standby status or waiting to be repaired. The failure rate of a powered-off spare is assumed to be zero. The failures are classified into $m + 1$ distinct classes with a conditional probability p_i for class i . For failures of classes $1, 2, \dots, m$, the average repair time is $1/\mu_i$, and failures of each class possess a dedicated repair facility. Failures of class 0 are transient, so the corresponding machine is returned immediately to the queue of good standby machines. The probability that at least one machine is available equals the steady-state availability and is given by formula (9.25):

$$A_0 = \rho_0 \frac{C(n-1)}{C(n)}. \quad \sharp$$

This treatment can be generalized to the case of a closed queuing network with c_i servers of rate μ_i at node i ($i = 0, 1, \dots, m$). If we compute v_i and ρ_i as before, then the joint probability of k_i jobs at node i ($i = 0, 1, \dots, m$) is given by [KLEI 1975]

$$p(k_0, k_1, \dots, k_m) = \frac{1}{C(n)} \prod_{i=0}^m \frac{\rho_i^{k_i}}{\beta_i(k_i)}, \quad (9.29)$$

where

$$\beta_i(k_i) = \begin{cases} k_i!, & k_i < c_i, \\ c_i! c_i^{k_i - c_i}, & k_i \geq c_i, \end{cases}$$

and

$$C(n) = \sum_{s \in I} \prod_{i=0}^m \frac{\rho_i^{k_i}}{\beta_i(k_i)},$$

where $I = \{(k_0, k_1, \dots, k_m) \mid k_i \geq 0 \text{ and } \sum_{i=0}^m k_i = n\}$. The computation of $C(n) = C_m(n)$ may be performed using the following recursive scheme [WILL 1976]. For $i = 0, 1, \dots, m$, let

$$r_i(k) = \begin{cases} \frac{\rho_i^k}{\beta_i(k)}, & k \neq 0, \\ 1, & k = 0. \end{cases}$$

Then, for $j = 1, 2, \dots, n$, let

$$C_i(j) = \begin{cases} r_0(j), & i = 0, \\ \sum_{k=0}^j C_{i-1}(j-k)r_i(k), & i \neq 0, \end{cases} \quad (9.30)$$

TABLE 9.6. Normalization Constant for Load-dependent Servers $C_i(j)$

$j \backslash i$	0	1	2	\dots	$i - 1$	i	\dots	m
0	1	1	1	\dots	$C_{i-1}(0) * r_i(n)$	$\rightarrow +$	\dots	
1	$r_0(1)$			\dots	$C_{i-1}(1) * r_i(n-1)$	$\rightarrow +$		
2	$r_0(2)$							
3						$\rightarrow +$		
.	.					$\rightarrow +$.	
.	.					$\rightarrow +$.	
.	.					$\rightarrow +$.	
n	$r_0(n)$		\dots		$C_{i-1}(n-1) * r_i(1)$	$\rightarrow +$	$C_i(n)$	\dots
					$C_{i-1}(n)$	\rightarrow		$C_m(n)$

with the initialization, $C_i(0) = 1$ for all i . The computation of $C_i(n)$ is depicted in Table 9.6. A comparison of this table with Table 9.1 illustrates the greater complexity of the load-dependent case. Also note that, unlike the previous case, in this case it is necessary to save all the values in column $i - 1$ while computing elements of column i . Thus two columns of the matrix, rather than just one, need to be stored.

A program to compute $C_i(j) = C(j), j = 1, 2, \dots, n$ can be easily written. Assume that a two-dimensional array $C[0..n, 0..1]$ is declared and two binary variables PREV and CUR are also declared. The $r_i(k)$ values as specified above are assumed to be precomputed and stored in the two-dimensional array $r[0..m, 0..n]$. Program 9.2 (below) computes the desired value $C(n) = C[n, \text{PREV}]$.

Program 9.2 (Normalization Constant Computation for a Closed Queuing Network with a Single Job Type and Load-Dependent Servers)

```

begin
{initialize}
  C[0,0] := 1; C[0, 1] := 1;
  for j := 1 to n do
    C[j, 0] := 0;
  PREV := 0; CUR := 1;
{recursion}
  for i := 0 to m do
    begin
      for j := 1 to n do
        begin
          C[j,CUR] := 0;
          for k := 0 to j do
            C[j,CUR] := C[j,CUR] + C[j - k, PREV] * r[i, k]
          end;
        end;
    end;
end;

```

```

    PREV := 1-PREV;
    CUR := 1-CUR
  end
end.

```

The expression for the utilization of node i is a bit more complex in this general case, but for the node with a single server (load-independent service), the formula

$$U_i(n) = \frac{\rho_i C(n-1)}{C(n)}$$

holds even though the nodes other than the i th node may give load-dependent service [BUZE 1973, WILL 1976].

Example 9.9

Consider a closed queuing network with two nodes. The CPU node is a multiple server node with the number of processors, c , varying from one to five. The total service requirement of a typical program on the CPU node is $E[B_0] = 10$ s. The I/O node is a single server node with the total service requirement $E[B_1] = 1$ s. The degree of multiprogramming $n = 5$.

Defining $\rho_0 = E[B_0]$ and $\rho_1 = E[B_1]$ we solve for average system throughput, $C(4)/C(5)$, as a function of the number of processors as shown in Table 9.7. Since programs are CPU-bound, increasing the number of processors improves average throughput substantially.

#

Example 9.10 ([TRIV 1978])

Consider the terminal-oriented distributed computing system shown in Figure 9.14. We use the following abbreviations:

T: the set of terminals

F: front-end interface processor

TABLE 9.7. Results for Example 9.9

Average throughput $E[T(c)]$	Number of processors c
0.0999991	1
0.1998464	2
0.2972297	3
0.3821669	4
0.4360478	5

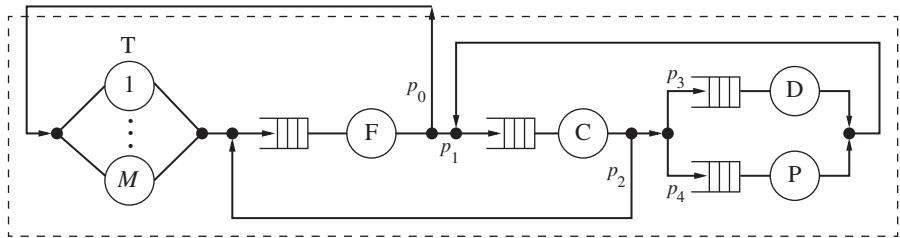


Figure 9.14. Queuing model of Example 9.10

C: communication processor

D: database management processor

P: principal element processor

The average think time of a terminal user is assumed to be $1/\lambda$. The routing matrix X is given by

$$X = \begin{bmatrix} & T & F & C & D & P \\ T & 0 & 1 & 0 & 0 & 0 \\ F & p_0 & 0 & p_1 & 0 & 0 \\ C & 0 & p_2 & 0 & p_3 & p_4 \\ D & 0 & 0 & 1 & 0 & 0 \\ P & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

where

$$p_0 + p_1 = 1, \quad p_2 + p_3 + p_4 = 1.$$

Solving for relative throughputs, we get

$$\begin{aligned} v_T &= v_F p_0 \\ v_F &= v_T + v_C p_2 \\ v_C &= v_F p_1 + v_D + v_P \\ v_D &= v_C p_3 \\ v_P &= v_C p_4. \end{aligned}$$

Choose $v_T = 1$ and then $v_F = 1/p_0$, $v_C = (1 - p_0)/(p_0 p_2)$, $v_D = [(1 - p_0)p_3]/(p_0 p_2)$, and $v_P = [(1 - p_0)p_4]/(p_0 p_2)$. Noting that the “service rate” of a terminal, μ_T , is given by λ , the device relative utilizations are given by

$$\begin{aligned} \rho_T &= \frac{1}{\lambda}, \quad \rho_F = \frac{1}{p_0 \mu_F}, \quad \rho_C = \frac{1 - p_0}{p_0 p_2 \mu_C}, \\ \rho_D &= \frac{(1 - p_0)p_3}{p_0 p_2 \mu_D}, \quad \rho_P = \frac{(1 - p_0)p_4}{p_0 p_2 \mu_P}. \end{aligned}$$

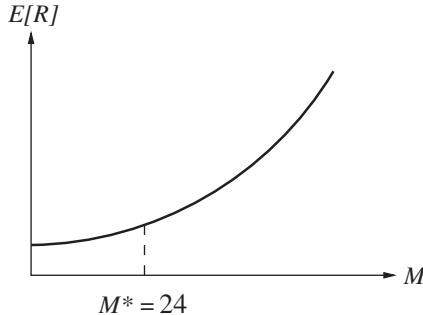


Figure 9.15. Average response time versus number of terminals

(Note that with this choice of v_T, ρ_i equals the average service time $E[B_i]$ per terminal request on device i .) Then, using formula (9.29), we have

$$p(k_T, k_F, k_C, k_D, k_P) = \frac{1}{C(M)} \frac{(\rho_T)^{k_T}}{k_T!} (\rho_F)^{k_F} (\rho_C)^{k_C} (\rho_D)^{k_D} (\rho_P)^{k_P}.$$

Note that the only node with load-dependent service is the terminal for which $\beta_T(k) = k!$, since necessarily $k \leq M$, the number of terminals. We can compute the $C(i)$ values using formula (9.30). Now, since the F node has only one server, we have

$$U_F = \rho_F \frac{C(M-1)}{C(M)} = \frac{1}{p_0 \mu_F} \frac{C(M-1)}{C(M)},$$

and the average throughput $E[T]$ or the rate of request completion is

$$E[T] = \mu_F p_0 U_F = \frac{C(M-1)}{C(M)}.$$

The average response time $E[R]$ can then be found from Little's formula, applied to the subsystem enclosed by dashed lines as $(E[R] + 1/\lambda)E[T] = M$, to be

$$E[R] = \frac{M}{E[T]} - \frac{1}{\lambda} = \frac{M \cdot C(M)}{C(M-1)} - \frac{1}{\lambda}. \quad (9.31)$$

As a numerical example, let $p_0 = 0.8$, $p_1 = 0.2$, $p_2 = 0.45833$, $p_3 = 0.33334$, and $p_4 = 0.20833$. Let $\mu_F = 1.5$, $\mu_C = 1.0$, $\mu_D = 0.2$, and $\mu_P = 0.2$. Let the average think time $1/\lambda = 15$ s (or $\lambda = 0.06667$). For this case, the average response time is plotted as a function of the number of terminals, M , in Figure 9.15.

#

Example 9.11 [MENA 1998]

Consider the HTTP Web browsing model shown in Figure 9.16, which models the traffic behavior of HTTP requests generated by M clients browsing the Internet. The

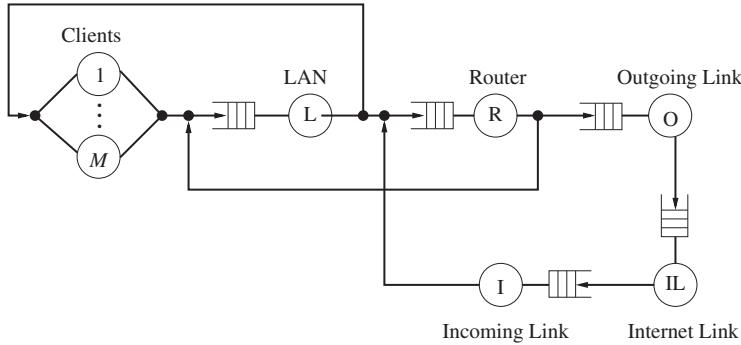


Figure 9.16. Queuing model of Example 9.11

thought process of the clients is represented by node C . When a request is generated, it first goes through the local area network (LAN) represented by node L , then to the router represented by node R . The outgoing link modeled by node O connects the router and the Internet service provider (ISP). Since the link connecting the router and the ISP is full duplex, a separate incoming link node I is used for the incoming traffic. The delay at ISP, the Internet, and the remote server is represented by node IL . The performance measure of interest is the mean response time for a user to receive the data he/she has asked for.

Suppose that the service requirement at each station could be estimated from real Internet measurement data as follows:

$$\begin{aligned} E[B_C] &= 3.33333 \text{ s}, & E[B_L] &= 0.01884 \text{ s}, & E[B_R] &= 0.00115 \text{ s}, \\ E[B_O] &= 0.04185 \text{ s}, & E[B_{IL}] &= 1.2615 \text{ s}, & E[B_I] &= 3.18129 \text{ s}. \end{aligned}$$

If v_C is chosen to be equal to the average number of visits V_C to node C , then the relative utilization ρ_i is equal to $V_i/\mu_i = E[B_i]$, the expected service requirement imposed by a typical client request on the i th node.

Then, the joint probability is given by

$$p(k_C, k_L, k_R, k_O, k_{IL}, k_I) = \frac{1}{C(M)} \frac{(\rho_C)^{k_C}}{k_C!} (\rho_L)^{k_L} (\rho_R)^{k_R} (\rho_O)^{k_O} (\rho_{IL})^{k_{IL}} (\rho_I)^{k_I}$$

The utilization at the LAN is

$$U_L = \rho_L \frac{C(M-1)}{C(M)},$$

and the average throughput $E[T]$ or the rate of request completion is

$$E[T] = \frac{C(M-1)}{C(M)}.$$

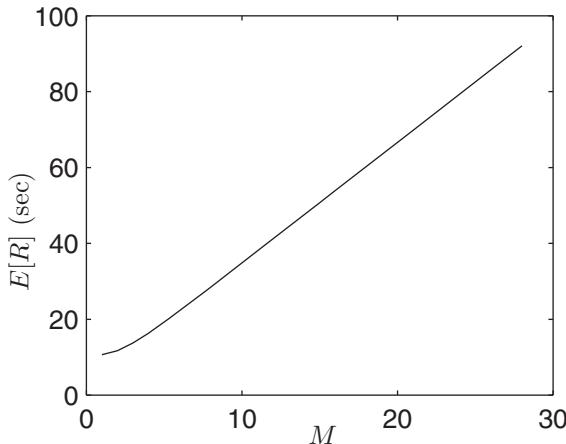


Figure 9.17. Average response time versus number of clients

The average response time $E[R]$ can then be found from equation (9.31), which is

$$E[R] = \frac{M}{E[T]} - \frac{1}{\lambda} = \frac{MC(M)}{C(M-1)} - \frac{1}{\lambda}. \quad (9.32)$$

The average response time $E[R]$, as a function of the number of clients M , is plotted in Figure 9.17. ‡

More details about the convolution algorithm are given in Bruell and Balbo [BRUE 1980]. Because the computation of the normalization constant can cause numerical problems, other techniques were developed that allow the calculation of the performance measures without using the normalization constant. One of the most efficient algorithms for calculating performance measures of closed product-form queuing networks is mean-value analysis (MVA), developed by Reiser and Lavenberg [REIS 1980]. It is an iterative technique where the mean values of the performance measures such as the mean waiting time, throughput, and the mean number of jobs at each node can be computed directly without computing the normalization constant. It is the algorithm of choice for networks with only IS (infinite server) and (load-independent) single server nodes.

The solution involves two simple laws: Little's formula applied to the overall system and the theorem of the distribution at arrival time. The arrival theorem says that in a closed product-form queuing network, the pmf of the number of jobs seen at the time of arrival to node i when there are n jobs in the network is equal to the pmf of the number of jobs at this node with one less job in the network [SEVC 1981, LAVE 1980].

We discuss the MVA for product-form single-class closed queuing network consisting of $(m + 1)$ nodes with n number of total jobs.

The key to MVA is the ability to give a recursive expression for the mean response time of a node in terms of measures for the network with one less job. For single server nodes with an FCFS strategy, the average response time per visit to node i , assuming there are j jobs in the network, is given by

$$E[R_i(j)] = \frac{1}{\mu_i} [1 + E[N_i(j - 1)]], i = 0, 1, \dots, m. \quad (9.33)$$

where $N_i(j - 1)$ is the number of jobs at node i , assuming that there are $j - 1$ jobs in the network. It is clear for node i that the mean response time of a job in a network with j jobs is given by the mean service time of that job plus the sum of the mean service times of all jobs that are ahead of this job in the queue. Equation (9.33) clearly follows from the use of arrival theorem but it can also be derived without using the arrival theorem. For this purpose, we use equation (9.25) for computing the utilization, and equation (9.27) for computing the mean number of jobs. Then

$$\begin{aligned} E[R_i(j)] &= \frac{E[N_i(j)]}{E[T_i(j)]} \\ &= \frac{E[N_i(j)]}{U_i(j)\mu_i} \\ &= \frac{1}{\mu_i} \left(1 + \frac{\sum_{k=2}^j \rho_i^{k-1} C(j-k)}{C(j-1)} \right) \end{aligned}$$

Now let $k = l + 1$,

$$\begin{aligned} E[R_i(j)] &= \frac{1}{\mu_i} \left(1 + \frac{\sum_{l=1}^{j-1} \rho_i^l C(j-l-1)}{C(j-1)} \right) \\ &= \frac{1}{\mu_i} (1 + E[N_i(j - 1)]). \end{aligned}$$

In the MVA technique, we actually use iteration instead of recursion over number of jobs starting at 1 and stopping at n . The following two equations would allow the iteration to proceed. First we would obtain the throughputs by applying Little's formula to the mean cycle time, namely, the mean time between a customer's arrivals at a queue.

$$E[T(j)] = \frac{j}{\sum_i v_i E[R_i(j)]}. \quad (9.34)$$

We would then determine the mean number of jobs at the i th node by using Little's formula again:

$$E[N_i(j)] = E[R_i(j)]v_iE[T(j)], \quad i = 0, 1, \dots, m. \quad (9.35)$$

The three equations, (9.33), (9.34), and (9.35), can be applied iteratively to compute $E[R_i(j)]$, $E[T(j)]$, and $E[N_i(j)]$, respectively, for any value of j starting at $j = 1$ and using the initial condition $E[N_i(0)] = 0$. Equation (9.33) is valid for FCFS single-server nodes, PS(processor sharing) nodes, and LCFS-PR (last come, first served, preemptive resume) nodes. Note that in the case of IS nodes, equation (9.33) can be easily modified and is given by

$$E[R_i(j)] = \frac{1}{\mu_i}, \quad i = 0, 1, \dots, m. \quad (9.36)$$

For the case of multiserver nodes ($c_i > 1$), it is necessary, however, to compute the marginal probabilities. Let us use $\pi_i(k-1|j-1)$ for the probability that $N_i = k-1$ in a network with $j-1$ jobs. Then we have

$$E[R_i(j)] = \sum_{k=1}^j \frac{k}{\mu_i(k)} \pi_i(k-1|j-1), \quad i = 0, 1, \dots, m, \quad (9.37)$$

where $\mu_i(k)$ denotes the service rate of node i given k jobs. To obtain the queue length pmf, we can use

$$\begin{aligned} \pi_i(k|j) &= \frac{E[T_i(j)]}{\mu_i(k)} \pi_i(k-1|j-1), \\ \pi_i(0|0) &= 1, \\ \pi_i(0|j) &= 1 - \sum_{k=1}^j \pi_i(k|j). \end{aligned} \quad (9.38)$$

Therefore, the mean response time at the i th node can be obtained by induction and rearrangement of the equations and is given by

$$E[R_i(j)] = \frac{1}{c_i \mu_i} \left[1 + E[N_i(j-1)] + \sum_{k=0}^{c_i-2} (c_i - k - 1) \pi_i(k|j-1) \right], \quad (9.39)$$

where

$$\begin{aligned} \pi_i(0|j) &= 1 - \frac{1}{c_i} \left[\frac{E[T_i(j)]}{\mu_i} + \sum_{k=1}^{c_i-1} (c_i - k) \pi_i(k|j) \right], \\ \pi_i(k|j) &= \frac{E[T_i(j)]}{\mu_i(k)} \pi_i(k-1|j-1). \end{aligned}$$

The throughput of each node can be computed using the following equation:

$$E[T_i(j)] = v_i E[T(j)]. \quad (9.40)$$

We can derive the other performance measures, such as utilization, mean waiting time, and mean queue length, from the calculated measures using the well-known equations.

Now, we shall describe the procedure of computing the state probabilities using the MVA. Akyildiz and Bolch [AKYI 1983] have extended the MVA for computing the normalization constant and for computing the state probabilities.

It is known that the expression for the throughput of node i in the load-dependent or load-independent case is given by

$$E[T_i(j)] = v_i \frac{C(j-1)}{C(j)}, \quad (9.41)$$

and the expression for the throughput of the network in the load-dependent or load-independent case is given by

$$E[T(j)] = \frac{C(j-1)}{C(j)} \quad (9.42)$$

Substituting the value of $E[T(j)]$, which is computed from equation (9.34), in this equation, we can find the normalization constant

$$C(j) = \frac{C(j-1)}{E[T(j)]}, \quad (9.43)$$

with the initial condition $C(0) = 1$. Once the iteration stops, we have the normalization constant $C(n)$ that can be used to compute the state probabilities using equation (9.15).

Example 9.12

Consider the closed queuing network model shown in Fig 9.18 with $N = 2$ jobs. The average think time of a terminal (T) is assumed to be $1/\mu_0 = 1$ s. At the second node (F) we have $c_1 = 2$ identical processors with service rate $\mu_1 = 2$. Nodes C and D have exponentially distributed service times with means $1/\mu_2 = 0.6$ s and $1/\mu_3 = 0.8$ s, respectively.

The routing matrix X is given by

$$X = \begin{matrix} & T & F & C & D \\ T & 0 & 1 & 0 & 0 \\ F & p_0 & 0 & p_1 & p_2 \\ C & 0 & 1 & 0 & 0 \\ D & 0 & 1 & 0 & 0 \end{matrix}.$$

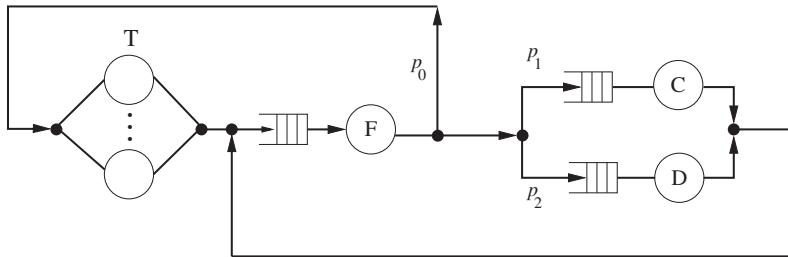


Figure 9.18. Closed queuing network model of Example 9.12

The routing probabilities are as follows:

$$p_0 = 0.1, \quad p_1 = 0.4, \quad p_2 = 0.5.$$

We get the relative throughputs by solving the equation (9.12):

$$v_0 = 1, \quad v_1 = 10, \quad v_2 = 4, \quad v_3 = 5.$$

We compute the performance measures and normalization constant with the help of MVA in two steps as follows:

Step 1: Initialization. For $i = 0, 1, 2, 3$, we obtain

$$E[N_i(0)] = 0, \quad \pi_1(0|0) = 1, \quad \pi_1(1|0) = 0.$$

Step 2: Iteration over the number of jobs $n = 1$ and 2. For $n = 1$, we compute

$$\begin{aligned} E[R_0(1)] &= 1, \quad E[R_1(1)] = \frac{1}{c_1 \mu_1} [1 + E[N_1(0)] + \pi_1(0|0)] = 0.5, \\ E[R_2(1)] &= \frac{1}{\mu_2} [1 + E[N_2(0)]] = 0.6, \quad E[R_3(1)] = \frac{1}{\mu_3} [1 + E[N_3(0)]] = 0.8. \end{aligned}$$

From (9.34), we find

$$E[T(1)] = \frac{1}{\sum_{i=0}^3 v_i E[R_i(1)]} = 0.0806.$$

Substituting the results $E[T(1)]$ and $E[R_i(1)]$ in equation (9.35), we get $E[N_i(1)]$:

$$E[N_0(1)] = E[R_0(1)]v_0E[T(1)] = 0.0806,$$

$$E[N_1(1)] = E[R_1(1)]v_1E[T(1)] = 0.43,$$

$$E[N_2(1)] = E[R_2(1)]v_2E[T(1)] = 0.1934,$$

$$E[N_3(1)] = E[R_3(1)]v_3E[T(1)] = 0.3224.$$

Similarly, we can find $E[T(n)]$, $E[R_i(n)]$ and $E[N_i(n)]$ for $n = 2$ iteratively. Note that we compute $E[R_1(n)]$, $n = 1, 2$ by using the equation (9.39). Mean response times are given by

$$\begin{aligned} E[R_0(2)] &= 1, & E[R_1(2)] &= 0.5068, \\ E[R_2(2)] &= 0.7160, & E[R_3(2)] &= 1.0579, \end{aligned}$$

and throughput is $E[T(2)] = 0.0703$. Finally, the mean number of jobs in each node is given by

$$\begin{aligned} E[N_0(2)] &= 0.0703, & E[N_1(2)] &= 0.3563, \\ E[N_2(2)] &= 0.2013, & E[N_3(2)] &= 0.3719. \end{aligned}$$

Substituting the value of $E[T(n)]$, $n = 1$ and 2 in equation (9.43), we get the normalization constant of the network and is given by $C(2) = 176.4857$. With equation (9.15), we can compute the steady-state probabilities.

‡

The MVA algorithm for computing the performance measures of single-class closed queueing networks can easily be extended to the multiclass case [REIS 1980]. MVA has been extended to the analysis of mixed product-form queueing networks [ZAHO 1981]. However, even for product-form queueing networks, the computational cost of an exact solution becomes prohibitively expensive as the number of classes, jobs, and nodes grows. Bard [BARD 1979] first introduced an approximate MVA algorithm, which is a variant of the exact MVA algorithm for product-form queueing networks that yields approximate answers with reduced computational expense. Starting from Bard's work, numerous approximate MVA algorithms have been derived for product-form queueing networks [BOLC 1998, WANG 2000].

Problems

1. Solve Example 9.6, choosing the relative throughput $v_0 = 1/p_0 = 10$. Compare the values of $C(n)$, U_0 , and $E[T]$ with those obtained in the text. Next use MVA to calculate performance measures and compare with the answers obtained using the convolution algorithm.
2. Let us write the normalization "constant" $C(n)$ as a function of the relative utilization vector $\rho = (\rho_0, \rho_1, \dots, \rho_m)$ —that is, as $C_n(\rho)$. Show that

$$C_n(a\rho) = a^n C_n(\rho).$$

3. Recall that while dealing with closed queueing networks, the relative throughputs (and hence, relative utilizations) can be computed within a multiplying constant. The relative throughput v_0 can be chosen arbitrarily. Of course, different choices

of v_0 will result in different values of C_n, C_{n-1}, \dots . Assume that the first choice is $v_0 = \mu_0$, with the corresponding normalization constants C_n, C_{n-1}, \dots . Now let $v_0 = 1$ and denote the corresponding sequence of normalization constants by M_n, M_{n-1}, \dots . Show that although C_k may not be equal to M_k , the utilization U_i for each node i has the same final value regardless of what choice of v_0 was made. Do the same for the expressions of average system throughput.

4. Derive a closed-form expression for average system throughput for a closed queuing network under monoprogramming (i.e., $n = 1$).
5. Given a closed queuing network with a single server at each node and relative utilizations that are independent of n , show that in the limit as n approaches infinity, average system throughput is given by

$$\lim_{n \rightarrow \infty} E[T(n)] = \frac{v_j}{V_j} \min_i \left\{ \frac{1}{\rho_i} \right\} = \min_i \left\{ \frac{1}{E[B_i]} \right\},$$

and it is for this reason that the network node with the largest value of $E[B_i]$ (or the largest relative utilization ρ_i) is called the “bottleneck” node. Show that in the limit as n approaches infinity, the real utilization for node i is given by

$$\lim_{n \rightarrow \infty} U_i(n) = \frac{\rho_i}{\rho_b},$$

where b is the index of the bottleneck node.

6. Show that for a closed queuing network model of a terminal-oriented system with all other nodes in the subsystem having single servers, the heavy-load asymptote to the response time $E[R]$ is given by

$$E[R](\text{heavy load}) = ME[B_b] - \frac{1}{\lambda},$$

where b is the index of the bottleneck node. Now show that the saturation number M^* is given by

$$M^* = \frac{\sum E[B_i] + \frac{1}{\lambda}}{E[B_b]}.$$

7. Consider a central server network with two I/O channels and three CPUs. The average CPU time per program $E[B_0]$ is 500 ms, and the average time per program on the I/O devices is 175 ms and 100 ms, respectively. Compute the average system throughput for the degree of multiprogramming n varying from 1 to 5. First perform your computations manually, using the structure as in Table 9.6 and next using Program 9.2. Now vary the number of CPU's from one to three and study the effect on the average throughput.
8. For a closed network of Example 9.10 with a single server at each node and a total of three jobs, recalculate the performance measures using mean value analysis.

9. An interactive system workload measurement showed that the average CPU time $E[B_0]$ per terminal request was 4.6 s and the average disk time per request was 4.0 s [SEVC 1980]. Since the response time was found to be very large, two alternative systems were considered for purchase. Compared to the existing system (denoted **ex**), one of the proposed systems (denoted **tr1**) had a CPU 0.9 times as fast and the disk was twice as fast. The alternative system (denoted **tr2**) had a CPU 1.5 times as fast as that in **ex** and a disk twice as fast. Decide whether a change from **ex** to **tr1** can be recommended (assuming first that **tr2** is intolerably expensive), first on the basis of asymptotic bounds analysis as in problem 6 and then by exactly computing the response times for the two cases as functions of the number of terminals. Next, assuming that **tr2** is affordable, show how much reduction in response time is possible as a function of the number of terminals.

9.4 GENERAL SERVICE DISTRIBUTION AND MULTIPLE JOB TYPES

More general queuing network models that overcome many of the restrictions of the models mentioned earlier have been formulated and have been found to have a product-form solution [AKYI 1987, BOLC 1998, CHAN 1977, CHAN 1980, CHAN 1983]. For example, any differentiable service time distribution can be allowed at a node, provided that the scheduling discipline at the node is PS (processor sharing) or LCFS-PR (last come, first served, preemptive resume). Any differentiable service time distribution can also be allowed at a node with ample servers (so that no queuing is needed). Networks with multiple job types can also be analyzed. We will consider several examples of these more general models. For additional details, consult the references cited in this section.

Example 9.13

Consider an example of the cyclic queuing model of Figure 9.9 in which $p_0 = 0$. We assume that the I/O service times are exponentially distributed with parameter λ , but the CPU service times are hyperexponentially distributed with two phases so that its pdf is

$$f(t) = \alpha_1 \mu_1 e^{-\mu_1 t} + \alpha_2 \mu_2 e^{-\mu_2 t}. \quad (9.44)$$

Let k , the number of jobs in the CPU node, denote the state of the system so that the state space is $\{0, 1, \dots, n\}$. The corresponding stochastic process is not Markovian, since the future behavior of the process depends not only on the current state but also on the time spent on the CPU by the job undergoing service (assuming that $k > 0$). If this time is denoted by τ , then we need to consider (k, τ) as the state of the system, and the corresponding state space is nondenumerable.

We can avoid the difficulty, however, by observing that the job at the CPU will be in one of two alternative phases, as shown in Figure 9.19, where a job scheduled for the CPU chooses phase i with probability α_i . We further simplify this discussion by considering only two jobs in the network, and we assume that the CPU scheduling discipline is processor sharing (PS). This discipline is a limiting case of the

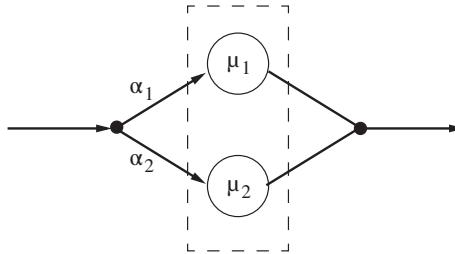


Figure 9.19. Two-phase hyperexponential CPU service time distribution

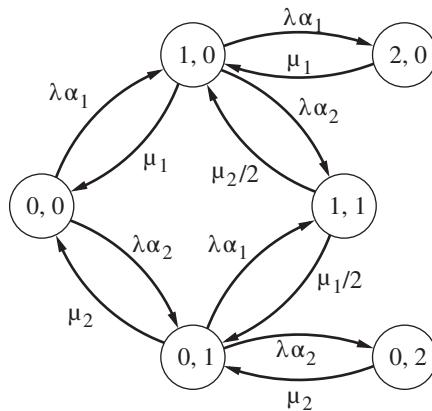


Figure 9.20. State diagram for the system of Example 9.13

quantum-oriented RR (round robin) discipline, where the quantum size is allowed to approach zero. As a result, the CPU is equally shared among all the jobs in the CPU queue. Thus, if there are k jobs in the queue, each job perceives the CPU to be slower by a factor k .

If the state of the system is (i, j) then i jobs are in the first phase of the CPU execution and j jobs in the second. Clearly, $i \geq 0, j \geq 0$ and $i + j \leq 2$. The state space is thus $\{(0, 0), (1, 0), (0, 1), (2, 0), (0, 2), (1, 1)\}$. The state diagram is shown in Figure 9.20. Since all interevent times are now exponentially distributed, we have a homogeneous CTMC. When the job finishes execution on the I/O device, it selects one of the two CPU phases with respective probabilities α_1 and α_2 . When the system is in state $(1, 1)$, two jobs are in the CPU queue, each of which perceives a CPU of half the speed. Thus the job in phase 1 completes its CPU burst requirement with rate $\mu_1/2$, and similarly for the other job.

Balance equations in the steady state are written as

$$\lambda p(0, 0) = \mu_1 p(1, 0) + \mu_2 p(0, 1),$$

$$(\mu_1 + \lambda)p(1, 0) = \mu_1 p(2, 0) + \alpha_1 p(0, 0) + \frac{\mu_2}{2} p(1, 1),$$

$$\begin{aligned}
(\mu_2 + \lambda)p(0, 1) &= \mu_2 p(0, 2) + \alpha_2 p(0, 0) + \frac{\mu_1}{2} p(1, 1), \\
\mu_1 p(2, 0) &= \lambda \alpha_1 p(1, 0), \\
\mu_2 p(0, 2) &= \lambda \alpha_2 p(0, 1), \\
\frac{\mu_1 + \mu_2}{2} p(1, 1) &= \lambda \alpha_1 p(0, 1) + \lambda \alpha_2 p(1, 0).
\end{aligned}$$

The reader should verify by substitution that a parametric solution of these equations is

$$\begin{aligned}
p(1, 0) &= \frac{\lambda \alpha_1}{\mu_1} p(0, 0), \\
p(0, 1) &= \frac{\lambda \alpha_2}{\mu_2} p(0, 0), \\
p(0, 2) &= \left(\frac{\lambda \alpha_2}{\mu_2}\right)^2 p(0, 0), \\
p(2, 0) &= \left(\frac{\lambda \alpha_1}{\mu_1}\right)^2 p(0, 0), \\
p(1, 1) &= \frac{2(\lambda \alpha_1)(\lambda \alpha_2)}{\mu_1 \mu_2} p(0, 0).
\end{aligned} \tag{9.45}$$

Now, using the normalization condition

$$p(0, 0) + p(1, 0) + p(0, 1) + p(2, 0) + p(0, 2) + p(1, 1) = 1,$$

we can compute $p(0, 0)$.

Suppose that we want to pursue a reduced description of the system in which state 1 of the reduced version corresponds to the union of original states $(1, 0)$ and $(0, 1)$, and state 2 is the union of original states $(2, 0)$, $(0, 2)$ and $(1, 1)$. Therefore:

$$\begin{aligned}
p(0) &= p(0, 0), \\
p(1) &= p(1, 0) + p(0, 1),
\end{aligned}$$

and

$$p(2) = p(2, 0) + p(0, 2) + p(1, 1).$$

From equations (9.45) we get

$$\begin{aligned}
p(1) &= \lambda \left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2} \right) p(0), \\
p(2) &= \lambda^2 \left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2} \right)^2 p(0).
\end{aligned} \tag{9.46}$$

TABLE 9.8. Networks with Product-form Solutions

<i>Scheduling discipline at node i</i>	<i>Service time distribution at node i</i>
FCFS	Exponential
PS	Coxian phase type
LCFS-PR	Coxian phase type
IS	Coxian phase type

Let $1/\mu = \alpha_1/\mu_1 + \alpha_2/\mu_2$ be the average CPU service time per visit to the CPU. Then the preceding equations are the special case of birth-death recursions, and the system has the well-known product-form solution.

‡

The argument presented in Example 9.13 above can be generalized to show that the product-form solution of Sections 9.2 and 9.3 is valid for a queuing network under the conditions shown in Table 9.8. Chandy and others [CHAN 1977] have further generalized these cases.

Now that we have considered an example of a network with nonexponential service time distribution, we turn our attention to queuing network models that support multiple job classes.

Example 9.14

Consider a central server network with three nodes (one CPU node labeled 0 and two I/O nodes labeled 1 and 2). Let the number of jobs in the network be $n = 2$. These jobs are labeled 1 and 2. Job 1 does not access I/O node 2, and job 2 does not access I/O node 1. The mean service time of job 1 on CPU is $1/\mu_1$, and that of job 2 is $1/\mu_2$. The mean I/O service time of job 1 on device 1 is $1/\lambda_1$, and that of job 2 on device 2 is $1/\lambda_2$. For simplicity we assume that there is no new program path, so that a job completing a CPU burst enters its respective I/O node with probability 1. Assume that the CPU scheduling discipline is PS.

Define the state of the system as a triple (k_0, k_1, k_2) where for $i = 0, 1, 2$, k_i is the number of jobs at node i . The state diagram is shown in Figure 9.21. From the state diagram we obtain the following balance equations:

$$\begin{aligned} \frac{\mu_1 + \mu_2}{2} p(2, 0, 0) &= \lambda_1 p(1, 1, 0) + \lambda_2 p(1, 0, 1), \\ (\lambda_1 + \mu_2)p(1, 1, 0) &= \lambda_2 p(0, 1, 1) + \frac{\mu_1}{2} p(2, 0, 0), \\ (\lambda_2 + \mu_1)p(1, 0, 1) &= \lambda_1 p(0, 1, 1) + \frac{\mu_2}{2} p(2, 0, 0), \\ (\lambda_1 + \lambda_2)p(0, 1, 1) &= \mu_1 p(1, 0, 1) + \mu_2 p(1, 1, 0). \end{aligned} \tag{9.47}$$

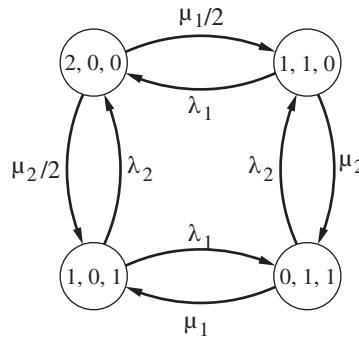


Figure 9.21. The state diagram for the central server network with two job types

The solution to this system of linear equations is easily seen to be

$$\begin{aligned}
 p(2, 0, 0) &= \frac{1}{C} \frac{2}{\mu_1 \mu_2}, \\
 p(1, 1, 0) &= \frac{1}{C} \frac{1}{\lambda_1 \mu_2}, \\
 p(1, 0, 1) &= \frac{1}{C} \frac{1}{\lambda_2 \mu_1}, \\
 p(0, 1, 1) &= \frac{1}{C} \frac{1}{\lambda_1 \lambda_2},
 \end{aligned} \tag{9.48}$$

where the normalization constant C is evaluated by using the condition:

$$p(2, 0, 0) + p(1, 1, 0) + p(1, 0, 1) + p(0, 1, 1) = 1$$

as

$$C = \frac{2}{\mu_1 \mu_2} + \frac{1}{\lambda_1 \mu_2} + \frac{1}{\mu_1 \lambda_2} + \frac{1}{\lambda_1 \lambda_2}. \tag{9.49}$$

The utilization of I/O device 1 is

$$U_1 = p(1, 1, 0) + p(0, 1, 1) = \frac{1}{C} \frac{1}{\lambda_1} \left[\frac{1}{\mu_2} + \frac{1}{\lambda_2} \right]$$

and that of I/O device 2 is

$$U_2 = p(1, 0, 1) + p(0, 1, 1) = \frac{1}{C} \frac{1}{\lambda_2} \left[\frac{1}{\mu_1} + \frac{1}{\lambda_1} \right].$$

The average throughput of type 1 jobs is therefore

$$E[T_1] = U_1 \lambda_1 = \frac{1}{C} \left[\frac{1}{\mu_2} + \frac{1}{\lambda_2} \right]$$

and that of type 2 jobs is

$$E[T_2] = U_2 \lambda_2 = \frac{1}{C} \left[\frac{1}{\mu_1} + \frac{1}{\lambda_1} \right].$$

‡

We can generalize this simple example by considering a closed network with r classes of customers [BASK 1975, CHAN 1977]. A class t customer has a routing matrix X_t , and its service rate at node i is denoted by μ_{it} . First we solve for the visit counts v_{it} by solving the traffic equations (9.16) for each $t = 1, 2, \dots, r$.

We admit four different types of service at a node. Node i is said to be a type 1 node if it has a single server with exponentially distributed service times, FCFS scheduling, and identical service rates for all job types (i.e., $\mu_{it} = \mu_i$ for all t). A node is said to be of type 2 if it has a single server, PS scheduling, and a service time distribution that is differentiable. Each job type may have a distinct service time distribution. Node i is said to be a type 3 node if it has an ample number of servers so that no queue ever forms at the node. Any differentiable service time distribution is allowed, and each job type may have a distinct service time distribution. Finally, a node is said to be of type 4 if it has a single server with LCFS-PR scheduling. Any differentiable service time distribution is allowed, and each job type may have a distinct service time distribution.

Let k_{it} be the number of jobs of type t at node i . Assume that there are n_t jobs of type t in the network so that we have

$$\sum_{i=0}^m k_{it} = n_t \quad \text{for each } t.$$

Define vector \mathbf{Y}_i by

$$\mathbf{Y}_i = (k_{i1}, k_{i2}, \dots, k_{ir}),$$

so that $(\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_m)$ is a state of the system. Let $k_i = \sum_{t=1}^r k_{it}$ be the total number of jobs at node i . The steady-state joint probability of such a state is given by [BASK 1975, CHAN 1977]

$$p(\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_m) = \frac{1}{C(n_1, n_2, \dots, n_r)} \prod_{i=0}^m g_i(\mathbf{Y}_i), \quad (9.50)$$

where

$$g_i(\mathbf{Y}_i) = \begin{cases} k_i! \prod_{t=1}^r \frac{1}{(k_{it})!} (v_{it})^{k_{it}} \left(\frac{1}{\mu_i}\right)^{k_i}, & \text{node } i \text{ is type 1,} \\ k_i! \prod_{t=1}^r \frac{1}{(k_{it})!} \left[\frac{v_{it}}{\mu_{it}}\right]^{k_{it}}, & \text{node } i \text{ is type 2 or 4,} \\ \prod_{t=1}^r \frac{1}{(k_{it})!} \left[\frac{v_{it}}{\mu_{it}}\right]^{k_{it}}, & \text{node } i \text{ is type 3.} \end{cases}$$

Define the relative utilization of node i due to jobs of type t by

$$\rho_{it} = \frac{v_{it}}{\mu_{it}}. \quad (9.51)$$

Then the real utilization of node i (of type 1, 2, or 4) due to jobs of type t is given by [WILL 1976]

$$U_{it} = \frac{\rho_{it} C(n_1, n_2, \dots, n_{t-1}, n_t - 1, n_{t+1}, \dots, n_r)}{C(n_1, n_2, \dots, n_{t-1}, n_t, n_{t+1}, \dots, n_r)}, \quad (9.52)$$

and the utilization of node i is given by

$$U_i = \sum_{t=1}^r U_{it}. \quad (9.53)$$

Assuming that all nodes are of type 1, 2, or 4, a recursive formula for the computation of the normalization constant C is derived in a way analogous to the single-job type case:

$$C(n_1, n_2, \dots, n_r) = C_m(n_1, n_2, \dots, n_r),$$

where for $i = 1, 2, \dots, m$, and for $j_t = 1, 2, \dots, n_t$:

$$\begin{aligned} C_i(j_1, j_2, \dots, j_r) &= C_{i-1}(j_1, j_2, \dots, j_r) \\ &+ \sum_{\substack{t=1 \\ j_t \neq 0}}^r \rho_{it} C_i(j_1, j_2, \dots, j_{t-1}, j_t - 1, j_{t+1}, \dots, j_r) \end{aligned} \quad (9.54)$$

with the initial conditions

$$C_0(j_1, j_2, \dots, j_r) = \frac{(j_1 + j_2 + \dots + j_r)!}{j_1! j_2! \dots j_r!} \prod_{t=1}^r \rho_{0t}^{j_t}$$

and

$$C_i(0, 0, \dots, 0) = 1.$$

Computational techniques for product-form networks have been discussed in the literature [BOLC 1998, CONW 1989], and reports on further theoretical development are also available [LAM 1977, KELL 1979, VAND 1993].

As with a single-job type, we can define the relative utilizations $\rho_{it} = E[B_{it}]$, the total service requirement of type t job on server i . In this case, the routing matrices X_t need not be specified and visit counts need not be computed.

Example 9.15

A database server processes three types of jobs. Jobs of type 1 are I/O-bound; in order to complete execution, they need 1 s of CPU time (i.e., $E[B_{01}] = 1$), 10 s of I/O time, and one unit of main memory. Jobs of type 2 are balanced; they need 10 s each of CPU and I/O, and two units of main memory. Jobs of type 3 are CPU-bound; they consume 100 s of CPU time, 10 s of I/O time, and five units of main memory.

The total main memory available for user allocation is 10 units. Therefore, we can admit either (one job of type 1, two jobs of type 2, and one job of type 3) or (three jobs of type 1, one job of type 2, and one job of type 3) in the active set. Evaluate the effects of the two choices.

We can let

$$\begin{aligned}\rho_{01} &= E[B_{01}] = 1, & \rho_{11} &= E[B_{11}] = 10, \\ \rho_{02} &= E[B_{02}] = 10, & \rho_{12} &= E[B_{12}] = 10, \\ \rho_{03} &= E[B_{03}] = 100, & \rho_{13} &= E[B_{13}] = 10.\end{aligned}$$

For the first case with $n_1 = 1$, $n_2 = 2$, and $n_3 = 1$, we compute the following using formula (9.54):

$$\begin{aligned}C_1(1, 2, 1) &= 1,410,000, \\ C_1(0, 2, 1) &= 66,000, \\ C_1(1, 1, 1) &= 56,400, \\ C_1(1, 2, 0) &= 6,600.\end{aligned}$$

Now the average throughputs by class in jobs per second are computed to be

$$\begin{aligned}E[T_1] &= \frac{C_1(0, 2, 1)}{C_1(1, 2, 1)} = 0.04681, \\ E[T_2] &= \frac{C_1(1, 1, 1)}{C_1(1, 2, 1)} = 0.04, \\ E[T_3] &= \frac{C_1(1, 2, 0)}{C_1(1, 2, 1)} = 0.004681.\end{aligned}$$

For the second alternative, where $n_1 = 3$, $n_2 = 1$, and $n_3 = 1$, we have

$$\begin{aligned}E[T_1] &= \frac{C_1(2, 1, 1)}{C_1(3, 1, 1)} = 0.07758, \\ E[T_2] &= \frac{C_1(3, 0, 1)}{C_1(3, 1, 1)} = 0.01667, \\ E[T_3] &= \frac{C_1(3, 1, 0)}{C_1(3, 1, 1)} = 0.0055569.\end{aligned}$$

We notice that with the second choice, the average throughput of class 1 jobs has gone up considerably at the expense of class 2 jobs.

#

Problems

- Consider a special case of an $M/G/1$ queue with PS scheduling discipline and an arrival rate λ . The Laplace-Stieltjes transform of the service time B is given by

$$L_B(s) = b \left(\frac{\mu_1}{s + \mu_1} \right) + (1 - b) \left(\frac{\mu_1}{s + \mu_1} \right) \left(\frac{\mu_2}{s + \mu_2} \right)$$

with $0 \leq b \leq 1$. For instance, if $b = 0$, we have a two-stage hypoexponential distribution (Erlang, if $\mu_1 = \mu_2$); if $b = 1$, we have an exponential distribution; and if $b = p_1 + p_2\mu_2/\mu_1$ (where $p_1 + p_2 = 1$), we have a two-stage hyperexponential service time distribution. Draw the state diagram of the system, write down the steady-state balance equations, and proceed to show that the steady-state probability of n jobs in the system has the product-form solution, identical to the $M/M/1$ FCFS solution.

2. Consider a mixed interactive–batch system with 30 terminals, each with an average think time of 15 s. The mean disk service time is 20 ms, and the disk scheduling algorithm is FCFS. An interactive job makes an average of 30 disk requests and a batch job an average of 10 disk requests. The average CPU requirement ($E[B_{0, \text{batch}}]$) of a batch job is 5 s while that of a terminal user is 0.4 s. A round robin (RR) CPU scheduling algorithm with a small enough time quantum is employed so that the use of PS approximation is considered adequate. Determine the effect of the degree of multiprogramming of batch jobs on the average response time and batch throughput varying batch multiprogramming level from 0 to 10 (assuming that such a variation is permissible). The actual size of main memory will support a batch multiprogramming level of 1. In order to improve batch throughput, three alternatives are being considered:
 - (a) Adding main memory so that batch multiprogramming level can be increased up to 5.
 - (b) Purchasing a new CPU with a speed improvement factor of 1.4.
 - (c) Adding another disk of the same type.

For each alternative, calculate the resulting improvement in batch throughput and the positive or adverse effect on the terminal response time.

9.5 NON-PRODUCT-FORM NETWORKS

Few useful queuing network models possess the properties required for a product-form solution. One such example is the network of Figure 9.7, where a job may hold one of the active resources (CPU or I/O device) simultaneously with a partition of main memory. One possible method of solution is to draw the state transition diagram of the CTMC, write down the balance equations, and proceed to solve them [BOLC 1998, SAHN 1996]. As an illustration, suppose that in Example 9.13 we require that CPU scheduling is FCFS rather than PS. Then it can be shown that the resulting network does not have a product-form solution, but the network can be represented by a CTMC and its steady-state solution obtained either directly or via the use of SPNs as discussed in Chapter 8. Nevertheless, this procedure is usually a formidable task, owing to the size of the state space.

Approximate solution techniques are applicable in many cases. Detailed expositions on approximation techniques may be found in the literature [7, 22, 48, 50]. We now illustrate approximation techniques based on a hierarchical (or multi-level) model by several examples.

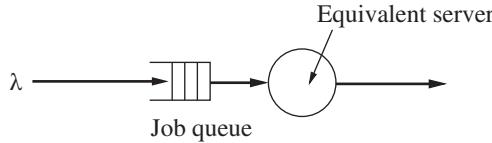


Figure 9.22. Equivalent server queuing system

Example 9.16

Returning to the queuing system shown in Figure 9.7, suppose that we represent the CPU-I/O subsystem by one equivalent server as seen by the job scheduler (see Figure 9.22). The service rate of this equivalent server is obtained from a closed version of the central server model (Figure 9.8); specifically, the average throughput of the central server model determines the service rate of the equivalent server. Since the average throughput depends on the degree of multiprogramming, the equivalent server has load-dependent service rates given by

$$\begin{aligned}\gamma_i &= E[T(i)], \quad 1 \leq i < n, \\ \gamma_i &= E[T(n)], \quad i \geq n,\end{aligned}\tag{9.55}$$

where n is the upper bound on the degree of multiprogramming. Once the average throughput vector $(E[T(1)], E[T(2)], \dots, E[T(n)])$ of the inner model is obtained, the outer model is recognized as a birth-death process with a constant birth rate λ and the death rates specified by equation (9.55).

Recall from our discussion of birth-death processes in Chapter 8 that if π_i is the steady-state probability that i jobs reside in the queuing system (of Figure 9.22), we have (assume that $\lambda E[T(n)] < 1$ for stability)

$$\begin{aligned}\pi_i &= \frac{\lambda}{\gamma_i} \pi_{i-1}, \quad i \geq 1, \\ &= \frac{\lambda^i}{\prod_{j=1}^i \gamma_j} \pi_0\end{aligned}$$

or

$$\pi_i = \begin{cases} \frac{\lambda^i}{\prod_{j=1}^i E[T(j)]} \pi_0, & 1 \leq i < n, \\ \frac{\lambda^i}{(E[T(n)])^{i-n} \prod_{j=1}^n E[T(j)]} \pi_0, & i \geq n. \end{cases}$$

From this, π_0 and $E[N] = \sum_{i=0}^{\infty} i\pi_i$ are determined as

$$\begin{aligned}\frac{1}{\pi_0} &= 1 + \sum_{i=1}^n \frac{\lambda^i}{\prod_{j=1}^i E[T(j)]} + \sum_{i=n+1}^{\infty} \frac{\lambda^i}{(E[T(n)])^{i-n} \prod_{j=1}^n E[T(j)]} \\ &= 1 + \sum_{i=1}^n \frac{\lambda^i}{\prod_{j=1}^i E[T(j)]} + \frac{1}{\prod_{j=1}^n E[T(j)]} \frac{\lambda^{n+1}}{E[T(n)] - \lambda},\end{aligned}$$

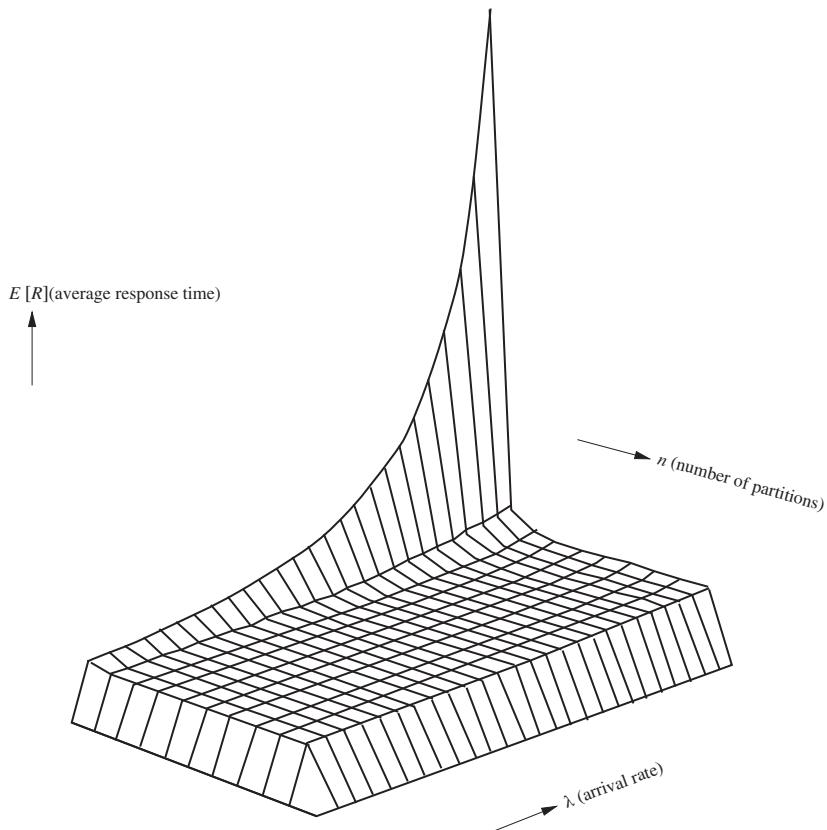


Figure 9.23. The behavior of the average response time as a function of the number of partitions and the arrival rate

$$E[N] = \pi_0 \left[\left\{ \sum_{i=1}^n \frac{i \lambda^i}{\prod_{j=1}^i E[T(j)]} \right\} + \frac{1}{\prod_{i=1}^n E[T(i)]} \frac{\lambda^{n+1}}{E[T(n)] - \lambda} \cdot \left\{ n + 1 + \frac{\lambda}{E[T(n)] - \lambda} \right\} \right].$$

Once we have computed $E[N]$, the average number of jobs in the system, we can determine the average response time $E[R]$ by using Little's formula:

$$E[R] = \frac{E[N]}{\lambda}.$$

In Figure 9.23 we have plotted the average response time, $E[R]$, as a function of the number of allowable partitions, n , and of the arrival rate λ . Various parameters of the system are specified in Table 9.9.

‡

TABLE 9.9. Parameters for Example 9.16

<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>
Number of I/O channels	m	2
Degree of multiprogramming	n	1–10
Branching probabilities	p_0	0.1
	p_1	0.7
	p_2	0.2
CPU service rate	μ_0	1
Drum service rate	μ_1	0.5
Disk service rate	μ_2	0.3
Arrival rate	λ	0.01–0.03

The approximation technique illustrated in Example 9.16 was developed by Chandy *et al.* [CHAN 1975] and is generally referred to in the literature as the *flow-equivalent server method*. The primary justification for this method is that it can be shown to yield exact results when applied to product-form networks [SAUE 1981]. Errors incurred in applying this technique to non-product-form networks are discussed by Tripathi [TRIP 1979], who also proposed methods of adjusting flow-equivalent server to account for the errors. The following example illustrates the nature of some of these errors.

Example 9.17

Consider the queuing network of Example 9.16 with the restriction that the number of partitions $n = 1$. The approximation technique of Example 9.16 will then imply that the equivalent server has load-independent service time, hence the reduced network is an $M/M/1$ queue. However, the actual “overall” service time of a job is composed of many exponential stages and will have a general distribution. To perform an exact analysis of this network, we can consider the system as an $M/G/1$ queue. In order to apply the P-K (Pollaczek–Khinchin) mean-value formula (7.60), we must compute the second moment of the service time distribution.

Each individual job can be seen to execute the following program.

Program 9.3 (Convolution Algorithm)

```

begin
  COMP;
  while B do
    begin
      case i of
        1: I/O1;
        2: I/O2;
      :
    end
  end

```

```

m: I/Om
end;
COMP
end
end.

```

However, this is precisely the program we analyzed in Chapter 5 (Example 5.24), where we derived an expression for the second moment $E[S^2]$ [equation (5.67)]. Then, using the $M/G/1$ formula (7.60), we obtain the average queue length $E[N]$, whence, using Little's result, we get the average response time $E[R]$. Note that with this analysis, service time distribution at each individual server is allowed to be general. We consider the system whose parameters are specified in Table 9.9 and compute the average response time with $n = 1$ using the $M/G/1$ formula. The $M/G/1$ results and the $M/M/1$ results (as in Example 9.16) are compared in Table 9.10. Both sets of results correspond fairly well because $\sigma_S = (E[S^2] - (E[S])^2)^{1/2} = 32.26$ while $E[S] = 30.667$, implying that the coefficient of variation $C_S \simeq 1$, and thus the overall service time distribution is close to exponential in this case.

‡

Example 9.18

Next, consider a model of a terminal-oriented system shown in Figure 9.24. In this model, the number of active jobs concurrently sharing main memory is limited by n , the number of partitions. Whenever more than n terminal requests are pending, the remaining jobs will have to queue up for memory. The resulting queuing network model does not belong to the product-form class.

One method to solve the problem is to construct the underlying homogeneous CTMC. To simplify the task of constructing the state space, we use the SRN, which was introduced in Section 8.7. Figure 9.25 shows the SRN model for this problem. The enabling of t_{enter} is controlled by its guard function $[g] = (\#P_{\text{CPU}} + \sum_{i=1}^m \#P_{\text{I/O}_i} < n)$. To obtain system throughput, we define the reward rate:

$$r_{\text{systhrou}} = p_0 \cdot \text{rate}(T_{\text{CPU}})$$

The expected steady-state reward rate will then yield the system throughput. Using Little's formula, we can obtain the mean response time:

$$E[R] = \frac{M - E[\#P_{\text{req}}]}{\text{systhrou}}$$

TABLE 9.10. $E[R]$

	λ			
	0.01	0.02	0.025	0.03
$M/G/1$	44.9556	81.9097	136.8145	402.2165
$M/M/1$	44.2315	79.3126	131.4347	383.3854

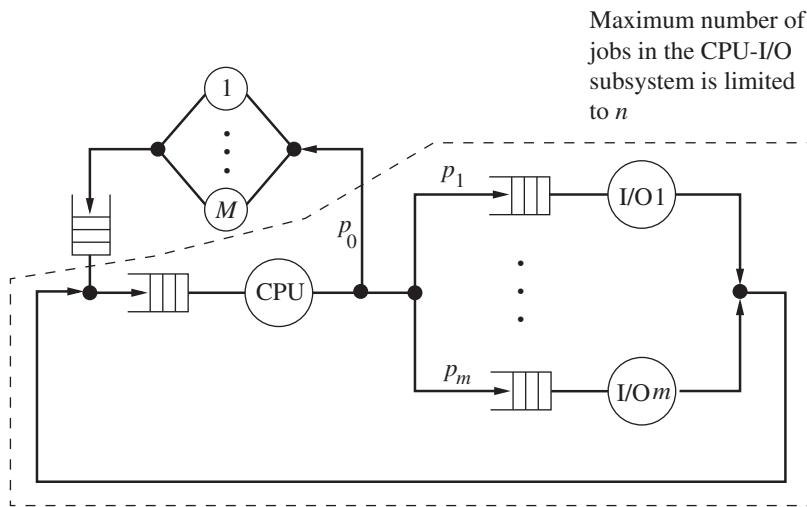


Figure 9.24. A terminal-oriented system with a limited number of memory partitions

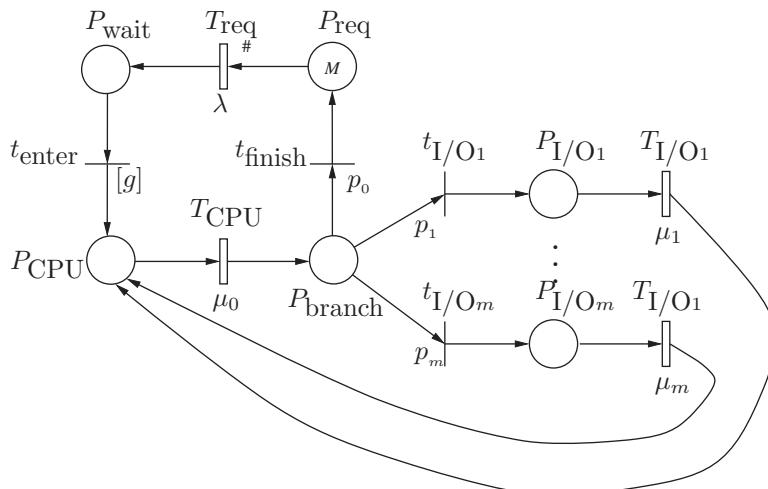


Figure 9.25. SRN model of terminal-oriented system with limited memory

TABLE 9.11. Parameters of Figure 9.24

	CPU	I/O1	I/O2	I/O3
μ_i	89.3	44.6	26.8	13.4
p_i	0.05	0.5	0.3	0.15

TABLE 9.12. The numbers of states and the numbers of nonzero entries in the Q matrix when $n = 4$ and $m = 3$

	M					
	10	20	30	40	50	60
# states	616	1526	2436	3346	4256	5166
# entries	3440	8800	14160	19520	24880	30240

Then, with the help of SPNP, we can obtain numerical results with the parameters given in Table 9.11. In Table 9.12, we provide the number of states and number of non-zero entries in the Q matrix of the underlying CTMC.

Next, we resort to an approximation technique to solve this problem (as in Example 9.16). First we replace the terminal subsystem with a short circuit and compute the average throughput of the resulting central server network as a function of the number of jobs in the network. We denote this average throughput vector by $(E[T(1)], E[T(2)], \dots, E[T(n)])$. Now we replace the central server subnetwork by an equivalent server, as shown in Figure 9.26.

The stochastic process corresponding to the model of Figure 9.26 is a birth–death process with the following birth rates

$$\lambda_i = (M - i)\lambda, \quad i = 0, 1, \dots, M$$

and death rates:

$$\gamma_i = \begin{cases} E[T(i)], & i = 1, 2, \dots, n, \\ E[T(n)], & i > n. \end{cases}$$

Here $1/\lambda$ is the average think time of the terminal users. It follows that the probability that the equivalent server is idle (denoted by π_0) can be obtained by equation (8.32) as

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^M \frac{\lambda^k M!}{\prod_{j=1}^k \gamma_j (M - k)!}},$$

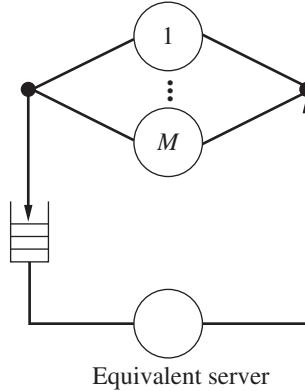


Figure 9.26. Upper level model

TABLE 9.13. Comparing the Exact and Approximate Values of Mean Response Time

	M					
	10	20	30	40	50	60
$E[R]$	1.023	1.22	1.54	2.23	3.84	6.78
$E[\tilde{R}]$	1.023	1.23	1.64	2.62	5.10	7.03
$E[\hat{R}]$	1.023	1.21	1.46	1.82	2.35	3.11

also

$$\pi_i = \frac{\lambda^i}{(M-i)!} \frac{M!}{\prod_{j=1}^i \gamma_j} \pi_0, \quad i = 1, 2, \dots, M.$$

The expected throughput, $E[T]$, of the equivalent server is obtained from

$$E[T] = \sum_{i=1}^M \pi_i \gamma_i = \sum_{i=1}^n \pi_i E[T(i)] + E[T(n)] \sum_{i=n+1}^M \pi_i.$$

Finally, the average response time $E[\tilde{R}]$ is computed from

$$E[\tilde{R}] = \frac{M}{E[T]} - \frac{1}{\lambda}.$$

As a numerical example, let the average think time $1/\lambda = 15$ s, and let other system parameters be as shown in Table 9.11, where $m = 3$. Assuming that n , the maximum number of programs allowed in the active set, is 4, we obtain the response time $E[\tilde{R}]$ as a function of the number of terminals M (see Table 9.13). $E[\hat{R}]$ denotes the value

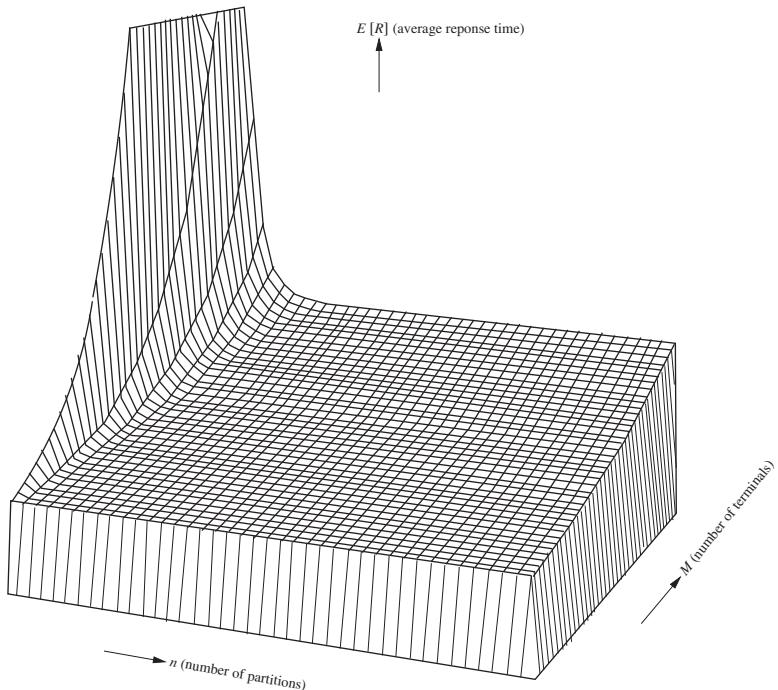


Figure 9.27. Plot of average response time for Example 9.18

of average response time, assuming that the main memory is large enough that no waiting in the job queue is required; that is, $n \geq M$. Sometimes, $E[\hat{R}]$ is used as an approximation to $E[R]$, but this example indicates that this approximation can be quite poor. In the table we also provide the exact result using the SRN method discussed earlier.

Figure 9.27 is a three-dimensional plot of the average response time as a function of the number of terminals M and the number of partitions n . We see that increasing n beyond 6 or 7 does not significantly reduce the response time. For instance, with $M = 40$ and $n = 8$, we have $E[\hat{R}] = 1.86$ while $E[\hat{R}] = 1.82$.

#

Example 9.19

Let us return to the two-node system with multiple processors discussed in Example 9.9. Assume that the failure rate of the I/O device is practically zero while the failure rate of each CPU is $\lambda = 10^{-4}$ failures per hour. Initially the system begins operation with five processors, and it continues to operate in a degraded mode in the face of processor failures until all processors have failed. The random time to failure X of the system is composed of five phases:

$$X = X_1 + X_2 + \cdots + X_5,$$

where the end of each phase marks the failure of a processor. From our discussion of reliability models we know that

$$X_k \sim EXP [(6 - k)\lambda].$$

Let W_k denote the number of jobs completed in phase k . Then the total number of jobs completed before system failure is

$$W = \sum_{i=1}^5 W_k.$$

Noting that the frequency of processor-I/O interactions is several orders of magnitude higher than the frequency of failure events, we may assume that the system reaches steady state long before the next failure. In this case

$$W_k \simeq E[T(6 - k)] \cdot X_k.$$

From this we compute the mean number of jobs completed before system failure as

$$\begin{aligned} E[W] &= \sum_{k=1}^5 E[W_k] \simeq E[T(6 - k)] \cdot E[X_k] \\ &= \sum_{k=1}^5 \frac{E[T(6 - k)]}{(6 - k)\lambda} \\ &= \frac{0.4817419}{\lambda} \\ &= 17,342,708 \text{ jobs}, \end{aligned}$$

using the results of Example 9.9 (Table 9.7).

Since the method discussed so far is approximate, we now carry out an exact analysis to assess the approximation error. The exact method to solve the problem is to construct the underlying homogeneous CTMC. Again, SRN can be used to construct the state space. Figure 9.28 shows the SRN model for this problem.

After generating the reachability graph of the SRN model and eliminating vanishing markings, we can get the CTMC model shown in Figure 9.29. In the CTMC model, state index is defined as (*jobs in cpu, available processors*) and state 0 is the absorbing state.

Using SHARPE, we can obtain expected accumulated reward until absorption:

$$E[Y(\infty)] = 17,343,008 \text{ jobs.}$$

It is interesting to investigate an alternative mode of system operation where only one CPU is active at one time while the remaining nonfaulty CPUs are in standby status. Now the average throughput is $0.099991 = E[T(1)]$ until the time to system failure. Since system MTTF is $5 \cdot 10^4$ h, the expected number of jobs completed before system crash is actually somewhat larger, equal to 17,999,838 jobs! However, the first system organization will provide better response time until it is degraded to one CPU. The technique illustrated here has been formalized under the topic of perfomability modeling [HAVE 2001].

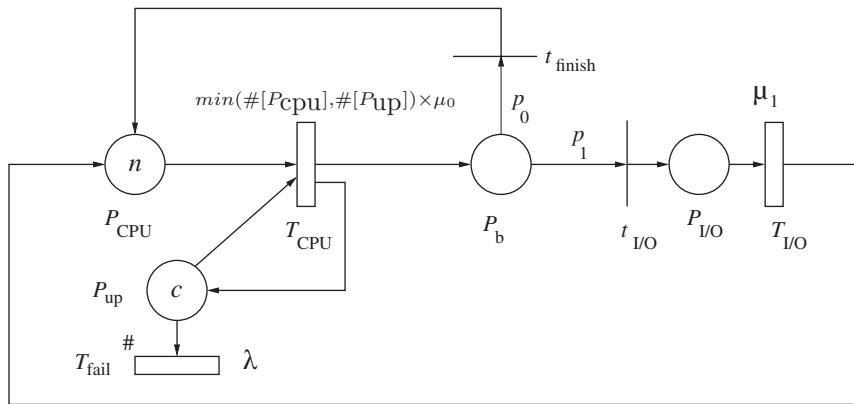


Figure 9.28. SRN model of the two-node system with multiple processors

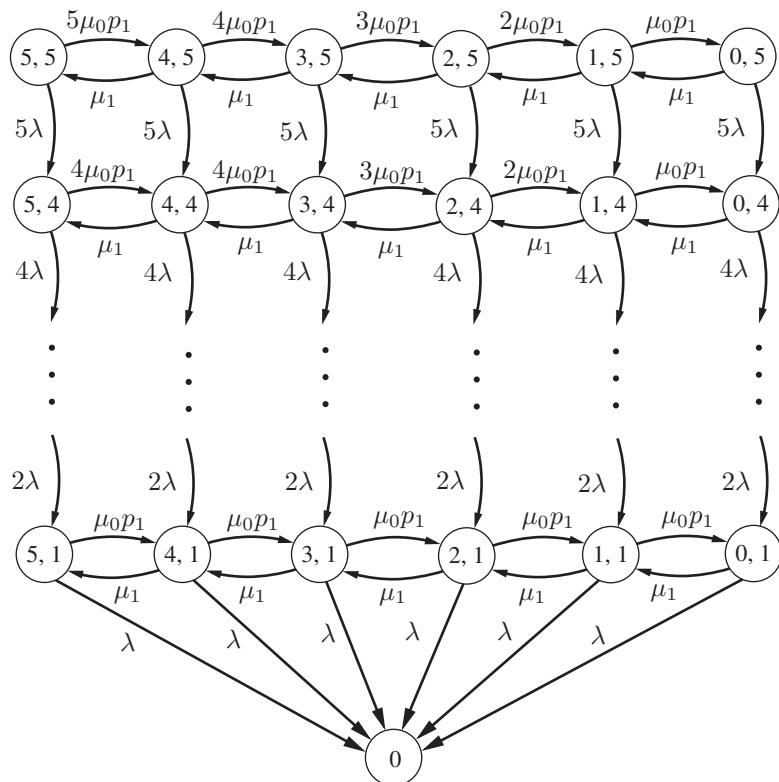


Figure 9.29. CTMC model for the two-node system with multiple processors

Problems

1. * Reconsider Example 9.16. We wish to study the behavior of the average response time $E[R]$ as a function of the degree of multiprogramming n . Decompose $E[R]$ into three parts:

$$E[R] = E[R_{eq}] + E[R_{iq}] + E[R_p],$$

where $E[R_{eq}]$ is the average time in the job queue (external queuing time), $E[R_{iq}]$ is the average queuing time in the queues internal to the CPU-I/O subsystem, and $E[R_p]$ is the average processing time on the servers in the subsystem. Derive expressions for $E[R_{eq}]$, $E[R_{iq}]$, and $E[R_p]$. Note that $E[R_{iq}] + E[R_p]$ is the average time spent by a job in the subsystem. For the parameters specified in Table 9.9, plot $E[R_{eq}]$, $E[R_{iq}]$, $E[R_p]$, and $E[R]$ as functions of n on the same graph paper. Give intuitive explanations for the shapes of these curves.

2. Consider the interactive system shown in Figure 9.P.2 (note that the topology is slightly different from that in Figure 9.24).

CPU MIPS rate = 1000

Number of instructions between two successive I/O requests = 2,000,000

Total number of instructions per program = 40,000,000

$1/\mu_2 = 5 \text{ ms}$, $1/\mu_1 = 0.8 \text{ ms}$.

Average think time = 5 s

Maximal degree of multiprogramming = 4.

Compute the average response time of the system first using the flow-equivalent server approximation and then exactly by first constructing a GSPN (or SRN) model of the system and then solving exactly using the SPNP software package.

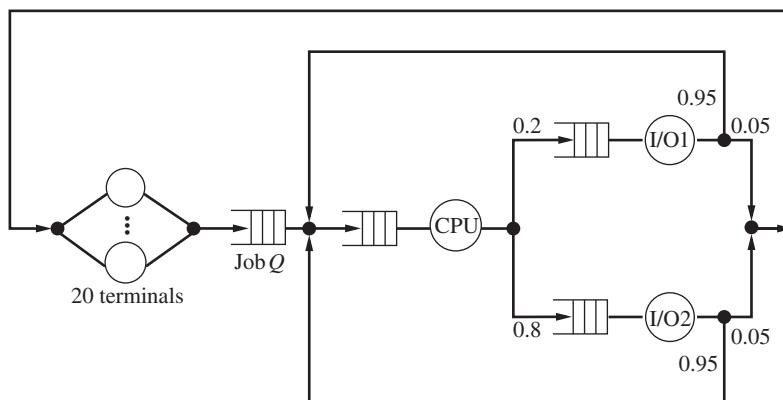


Figure 9.P.2. Another terminal-oriented system

3. Refer to Dowdy *et al.* [DOWD 1979]. Consider a closed central server model with two I/O channels with respective rates 5 s^{-1} and 3 s^{-1} . The CPU service rate is 7 s^{-1} , and the branching probabilities are $p_0 = 0.05$, $p_1 = 0.65$, and $p_2 = 0.3$. If the degree of multiprogramming is fixed, then the given network has a product-form solution as specified in Section 9.3. In practice, however, the degree of multiprogramming varies. Let the degree of multiprogramming N be a random variable. Compute and plot the average system throughput as a function of the average degree of multiprogramming $E[N]$ (varying from 1 to 10), assuming:
- N is a constant random variable with value $E[N]$.
 - N takes two different values: $N = E[N] - 1$ with probability 0.5 and $N = E[N] + 1$ with probability 0.5.
 - N is Poisson distributed with parameter $E[N]$.
 - N is binomially distributed with parameters $2E[N]$ and $\frac{1}{2}$.
 - N has a discrete uniform distribution over $\{0, 1, 2, \dots, 2E[N]\}$.
 - N takes two values: $N = 0$ with probability 0.5 and $N = 2E[N]$ with probability 0.5.
4. It can be shown that the procedure using the flow-equivalent server method gives exact results when applied to a queuing network belonging to the product-form class. Note that the interactive system model of Example 9.18 will be a product-form network if the number of partitions $n \geq M$. In this case, apply the technique [formula (9.29)] developed in Section 9.3 and compute the average response time $E[R]$ as a function of M . Now compute $E[R]$ using the procedure applied in Example 9.18 and compare the results.
5. Modify the problem of Example 9.18 so that the I/O device labeled 1 is a paging device. It is convenient to reparameterize the problem assuming $v_{\text{terminal-node}} = 1$. Then $\rho_0 = E[B_0] = 0.223964$ s, and $V_2 = 6$, $V_3 = 3$, and $V_1(n)$ are computed assuming the following page-fault characteristics of programs:

<i>Number of page faults, V_1</i>	13	13	162	235	240	300	900
<i>Page allotment</i>	25	12	6	5	4	3	2

Compute the average response time $E[R]$ as a function of the number of terminals M and the number of partitions n , assuming the total pageable memory is 50 page frames. Suppose now that we purchase a paging device with an average service time of 10 ms. Recompute $E[R]$ and compare with earlier results.

6. Compare three methods of solving the network in Example 9.18: (a) the approximation technique discussed in the example, (b) an exact solution obtained using stochastic reward nets (and SPNP software), and (c) using discrete-event simulation. Compare the three methods for their accuracy, execution time, and storage space needed.
7. Repeat the comparison of three methods for the network of Example 9.16.
8. Consider another approximate solution technique for the system of Example 9.18. Specifically, the equivalent server in Figure 9.26 is assumed to have a

load-independent service rate given by $E[T(n)]$, the average throughput of the subsystem at the maximal degree of multiprogramming. The resulting system was studied in Example 8.13. Let the average response time thus obtained be denoted by $E[\hat{R}]$. Compare the results obtained with those in Table 9.13. (Notice that the approximation $E[\hat{R}]$ will be good under heavy-load conditions, i.e., that is, $M \gg n$.)

9. Resolve Example 9.6 using flow-equivalent server method by first short-circuiting the CPU and determining the characteristics of the composite I/O server. Next solve a cyclic queuing network with a CPU and the composite I/O server (whose service rate is queue-size-dependent). Compare your answers with those obtained in the text.

9.6 COMPUTING RESPONSE TIME DISTRIBUTION

Closed-form solutions have been derived for the (Laplace–Stieltjes transform of) response time distributions through a particular path in open product-form queuing networks [SCHA 1987]. However, numerical inversion of the LST is a difficult numerical problem. Many results exist for response time distributions for networks with specific topological structures such as tandem, central server, and single queue with feedback. The communications literature also shows a focus on end-to-end packet delay in tandem-type queuing networks, with characteristics specific to communication systems. However, it is very difficult to derive exact closed-form solutions for networks with even slightly nonrestrictive topology and service and arrival characteristics. For closed Markovian networks, we advocate using a numerical solution technique via the use of SPNs [MUPP 1994]. For open networks we discuss an approximate numerical solution technique.

There are methods of approximating the distribution that can give good results with much less effort. One such method uses the original network and the knowledge of the response time distribution of its components to derive a homogeneous CTMC for which the distribution of the time to reach the absorbing state can be solved to find the approximate response time distribution of the queuing network model [WOOL 1993].

9.6.1 Response Time Distribution in Open Networks

In Section 8.2 it was shown that the response time for an $M/M/1$ FCFS queue is exponentially distributed with parameter $\mu - \lambda$. The model in Figure 9.4 is composed of two such queues. We will use this information to build the homogeneous CTMC shown in Figure 9.30. In Figure 9.30, state 0 is the starting state, and from there state 1 will be entered at a rate of $(\mu_0 - \lambda_0)p_1$ [λ_0 is given in equation (9.4)] and state C will be entered at a rate of $(\mu_0 - \lambda_0)p_0$. Observe that this rate is based on the response time distribution of the CPU node (i.e., $\mu_0 - \lambda_0$), and the probability of either entering the I/O queue or exiting the system, respectively. The rate at which state 0 is entered

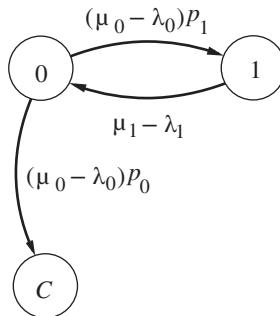


Figure 9.30. CTMC for response time distribution approximation of model in Figure 9.4a

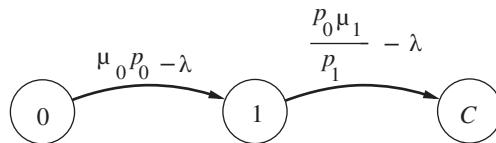


Figure 9.31. CTMC for response time distribution approximation of model in Figure 9.4b

from state 1 is $\mu_1 - \lambda_1$ [λ_1 is given in equation (9.5)], which is representative of the response time distribution of the I/O queue.

Figure 9.31 is the CTMC that can be used to find the response time distribution of the queuing model shown in Figure 9.4b. Recall that even though the two networks have identical mean response times, the distributions of the response time will be different.

To solve this model, we assume that $\lambda = 1$ job per second and that $p_0 = 0.2$. We assume that the CPU can process jobs at a rate of $\mu_0 = 10$ jobs per second and the I/O can process jobs at a rate of $\mu_1 = 5$ jobs per second. This CTMC was solved using the SHARPE [SAHN 1996] program to obtain the distribution of time to reach state C . For comparison, the original queuing network model was also simulated and the response time distribution found using the simulation tool SES/workbench¹.

Both the CTMC model and the simulation model found the mean response time to be 5 s for both network models. This is in agreement with the results expected as found in Example 9.2. The distributions of the response time for both network models as determined by both the CTMC approximation and the simulation model are shown in Figure 9.32. Note that for the model of

¹Registered trademark of Scientific and Engineering Software, Inc.

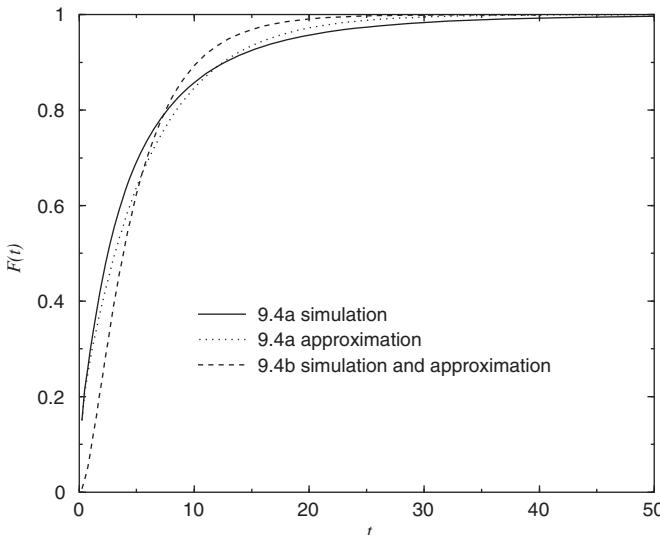


Figure 9.32. Response time distribution for networks of Figure 9.4, versions a and b

Figure 9.4a the CTMC approximation shows the distribution to be slightly lower than the resultant distribution of the simulation for values of $t < 12$, and slightly higher for the remaining values of t . For the model of Figure 9.4b, the results from CTMC approximation and the simulation model are in agreement for all t . This can be explained by the fact that the CTMC approach gives exact results when the queuing network is an open Jacksonian feedforward network with overtaking-free conditions [MAIN 1994b]. However, comparing the results from the “equivalent” network to the original, we see quite a bit of difference in the response time distribution.

The same approach that was used to approximate the response time distribution for Example 9.2 can be used for Example 9.3. The first thing that must be done to use the approximation is to limit the number of I/O servers, so that a solvable model can be created. We will assume 4 I/O servers, with p_1 through p_4 all having a value of 0.2. The service rates of the I/O processors are assumed to be $\mu_1 = 5$, $\mu_2 = 4$, $\mu_3 = 3$, and $\mu_4 = 2.5$. The CPU is again assumed to have a rate of $\mu_0 = 10$. Using the knowledge of the response time distribution for an $M/M/1$ FCFS queue and the structure of the network queuing model, we develop CTMC for the response time distribution approximation as shown in Figure 9.33 (see also Figure 9.34).

As in the preceding example, we start in state 0. We will then proceed to state C , 1, 2, 3, or 4 at a rate of $\mu_0 - \lambda_0$ factored by the appropriate probability. We return from states 1, 2, 3, or 4 (time spent at the respective I/O server) at a rate signified by the response time at that server, $\mu_i - \lambda_i$ for i ($i = 1, 2, 3, 4$). State C is indicative of service being complete.

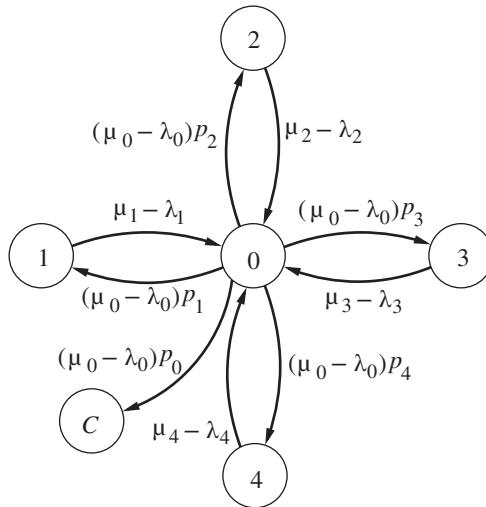


Figure 9.33. CTMC for response time distribution approximation of model in Example 9.3

9.6.1.1 Response Time Blocks. To derive blocks to approximate the response time distribution of queuing model nodes, we will assume that all nodes use first come, first served (FCFS) scheduling. The arrival rate to the nodes will be assumed to be λ and the service rate of each server of the nodes will be μ . To maintain a stable model, we will also assume that $\lambda < c\mu$ where c is the number of servers in the node.

M/M/1 In the previous examples, we have made use of the $M/M/1$ response time block. This block is based on the knowledge of an $M/M/1$ FCFS server with arrival rate λ and service rate μ having the following response time distribution, assuming that $\lambda < \mu$:

$$F(t) = 1 - e^{-(\mu-\lambda)t}.$$

The corresponding response time block is shown in Figure 9.35. State “In” indicates the starting state of the piece of the Markov chain model representing the $M/M/1$ queue in the corresponding network model. The “Out” state will be either an “In” state for another network model node or the absorbing state representing the job leaving the network model. Note that the “Out” state may actually be multiple states if the job exiting the corresponding node in the network model can proceed on multiple paths. Under these conditions, the rate entering the “Out” state would be weighted by appropriate probabilities and branched into multiple states.

M/M/ ∞ The $M/M/\infty$ server is the simplest node for which we can find the response time distribution. Since there are always enough servers

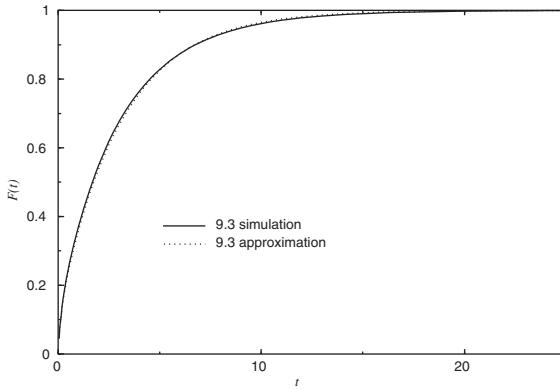


Figure 9.34. Response time distribution for Example 9.3.

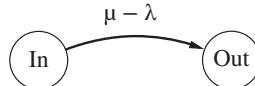


Figure 9.35. The $M/M/1$ response time block

for any customer, the response time distribution is the same as the service time distribution,

$$F(t) = 1 - e^{-\mu t}.$$

Figure 9.36 shows the response time block for this distribution. It is very similar to Figure 9.35, except that for the $M/M/\infty$ case the rate of leaving the “In” node is simply μ .

M/M/c For the $M/M/c$ FCFS queue, we assume that $\lambda < c\mu$. The $M/M/c$ FCFS response time distribution is given by

$$F(t) = \frac{\lambda - c\mu + \mu W_c}{\lambda - (c-1)\mu} [1 - e^{-\mu t}] + \frac{(1 - W_c)\mu}{\lambda - (c-1)\mu} [1 - e^{-(c\mu - \lambda)t}], \quad (9.56)$$

where

$$W_c = 1 - \frac{c (\lambda/\mu)^c}{c!(c-\lambda/\mu)} \pi_0$$

and

$$\pi_0 = \left[\sum_{j=0}^{c-1} \frac{1}{j!} \left(\frac{\lambda}{\mu} \right)^j + \frac{1}{c!} \left(\frac{\lambda}{\mu} \right)^c \left(\frac{c\mu}{c\mu - \lambda} \right) \right]^{-1}.$$

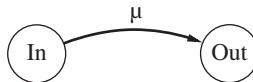


Figure 9.36. The $M/M/\infty$ response time block

Equation (9.56) represents a two-phase hyperexponential distribution in which one phase has parameter μ and the other phase has parameter $c\mu - \lambda$. An alternate form of this distribution can be shown to be the following phase-type distribution

$$F(t) = W_c(1 - e^{-\mu t}) + (1 - W_c) \left[\frac{c\mu - \lambda}{(c-1)\mu - \lambda} [1 - e^{-\mu t}] - \frac{\mu}{(c-1)\mu - \lambda} [1 - e^{-(c\mu - \lambda)t}] \right]. \quad (9.57)$$

This is a mixture of W_c fraction having exponential distribution with parameter μ and $(1 - W_c)$ fraction having hypoexponential distribution with parameters μ and $c\mu - \lambda$. Equation (9.57) can be described as a building block as shown in Figure 9.37. The upper path represents the exponentially distributed portion and the lower path is the hypoexponentially distributed portion. State T is a transient state that is required to obtain the hypoexponential distribution. If the output of the queuing node had multiple path possibilities, then the “Out” state shown would in fact be multiple states and the incoming arcs must be weighted with the appropriate probabilities.

With the building blocks for three types of nodes described, Woollet presents the procedure of mapping the response time distribution of an open queuing network to the time to absorption distribution of a CTMC [WOOL 1993]. We will illustrate the procedure via an example.

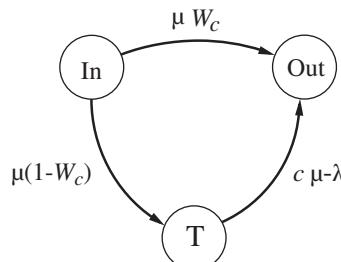


Figure 9.37. The $M/M/c$ response time block

Example 9.20

Consider a distributed system (Figure 9.38) in which users send requests at the rate λ from terminal with a delay time of $1/\mu_T$. A job first obtains service from a front-end processor (F), and may exit the system with probability x_{FO} after completion of service. With probability x_{FA} , it proceeds to the communications processor (A). After completion of service it may go back to the front-end processor with probability x_{AF} or to a database processor (D) with probability x_{AD} or to a general-purpose processor (P) with probability x_{AP} .

The terminals (T) are assumed to be $M/M/\infty$ servers having a service rate μ_T . F is an $M/M/c_F$ server with each of the c_F servers having service rate μ_F . A and D are assumed to be single-server $M/M/1$ queues having service rates μ_A and μ_D , respectively. P is assumed to be a $M/M/c_P$ queue. The service rate of each server is assumed to be μ_P .

On solving equation (9.9), we obtain the following values for effective arrival rates λ_F , λ_A , λ_D , and λ_P to each of the queues F, A, D, and P, respectively [WOOL 1993]:

$$\begin{aligned}\lambda_F &= \frac{\lambda}{x_{FO}}, \\ \lambda_A &= (1 - x_{FO}) \frac{\lambda}{x_{FO} x_{AF}}, \\ \lambda_D &= (1 - x_{FO}) \frac{x_{AD} \lambda}{x_{FO} x_{CF}}, \\ \lambda_P &= (1 - x_{FO}) \frac{x_{AP} \lambda}{x_{FO} x_{AF}}.\end{aligned}$$

Figure 9.39 shows the CTMC corresponding to the approximate response time distribution of this queuing network. States $F1, F2$ and $P1, P2$ are the two states of the building block corresponding to $M/M/c_F$ and $M/M/c_P$ respectively. The other states can be similarly identified.

For $\mu_F = \frac{1.5}{4}, \mu_A = 1, \mu_D = 0.2, \mu_T = \frac{1}{5}, \mu_P = \frac{0.2}{4}, c_F = 4, c_P = 2, x_{FO} = 0.5, x_{FA} = 0.5, x_{AF} = 0.46, x_{AD} = 0.33$, and $x_{AP} = 0.21$, the distribution of response time in this network is shown in Figure 9.41. For the values of $\lambda = \frac{1}{15}, \frac{1}{30}$, and $\frac{1}{100}$, the response time distribution is shown.

The CTMC of Figure 9.39 can be automatically generated from a GSPN model shown in Figure 9.40.

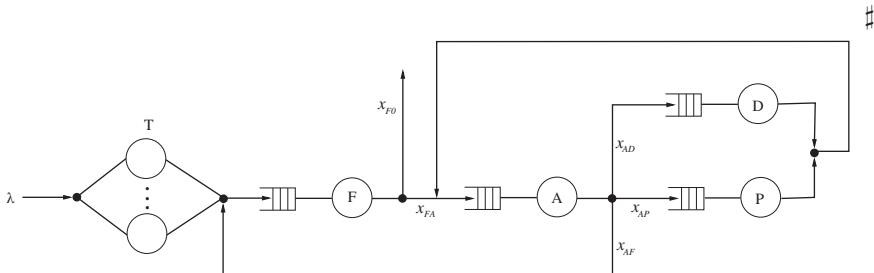


Figure 9.38. Distributed system queuing network

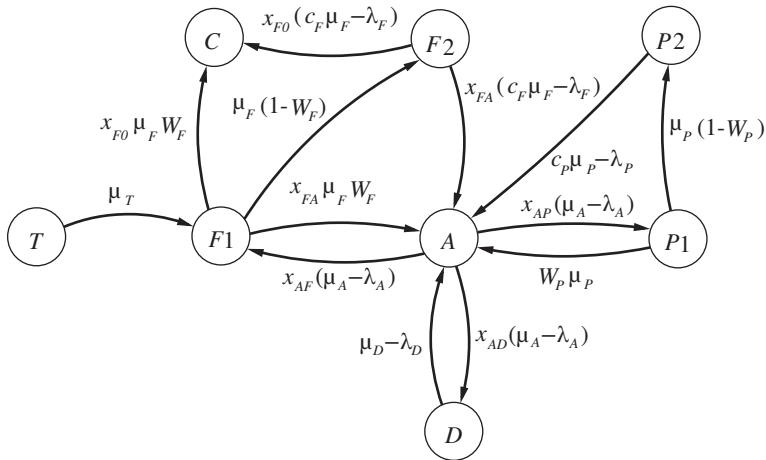


Figure 9.39. CTMC corresponding to response time distribution

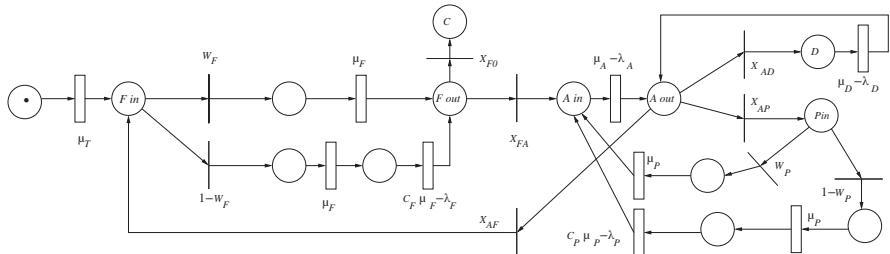


Figure 9.40. GSPN model for calculating the response time distribution

Now, we shall consider the numerical computation of response time distribution for closed queuing networks.

9.6.2 Response Time Distribution in Closed Networks

As we mentioned earlier, it is very difficult to obtain the closed-form expression for the response time distribution for queuing networks with a general structure. Numerical computation of the response time distribution is then the only alternative short of very expensive discrete-event simulation. One such method is based on the **tagged customer** approach. In this method, an arbitrary customer is picked as the tagged customer and its passage through the network is traced. By this method, the problem of computing the conditional response time distribution of the tagged customer is transformed into the time to absorption distribution of a finite state CTMC.

The tagged customer approach allows us to compute the response time distribution conditioned on the state of the queuing network at the time of

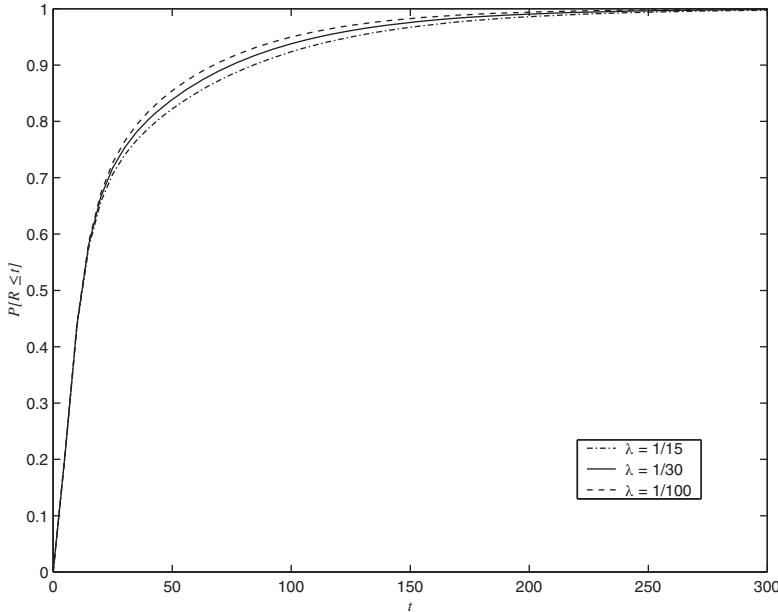


Figure 9.41. Response time distribution of distributed system

arrival of the tagged customer. To compute the unconditional response time distribution, we need to determine all the states in which the tagged customer may find the queuing network on arrival, that is, how the remaining customers are distributed among the queues in the network. For closed product-form queuing networks with n customers, the Sevcik–Mitrani [SEVC 1981] (see also Lavenberg and Reiser [LAVE 1980]) arrival theorem states that an arriving customer would see the network in equilibrium with one less customer. Thus, computing the response time distribution using the tagged customer approach is a two-step process, giving rise to a hierarchical model. The first step involves computing the steady-state probability vector for the queuing network with one less customer, $\pi(n - 1)$. The second step uses the probability vector to compute the unconditional response time distribution, $P[R \leq t]$.

Melamed and Yadin [MELA 1984] present a numerical method based on the tagged customer approach for evaluating the response time distribution in a discrete-state Markovian queuing network. The problem in using the tagged customer approach is the difficulty of constructing and solving rather large Markov chains. Muppala et al. [MUPP 1994] use the variation of stochastic Petri nets called **stochastic reward nets** (SRN) for the compact specification, and automated generation and solution of these large Markov chains [CIAR 1989]. This allows them to solve large and complex models. The following simple example will illustrate the computation of response time distribution using CTMC.

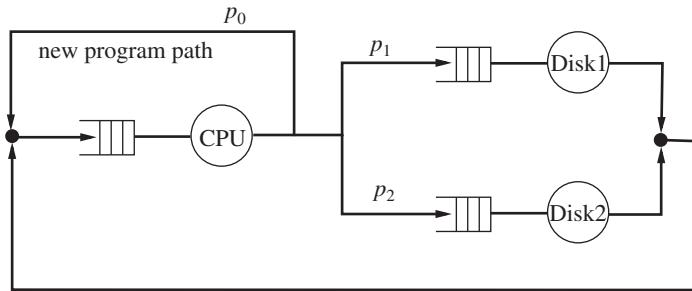


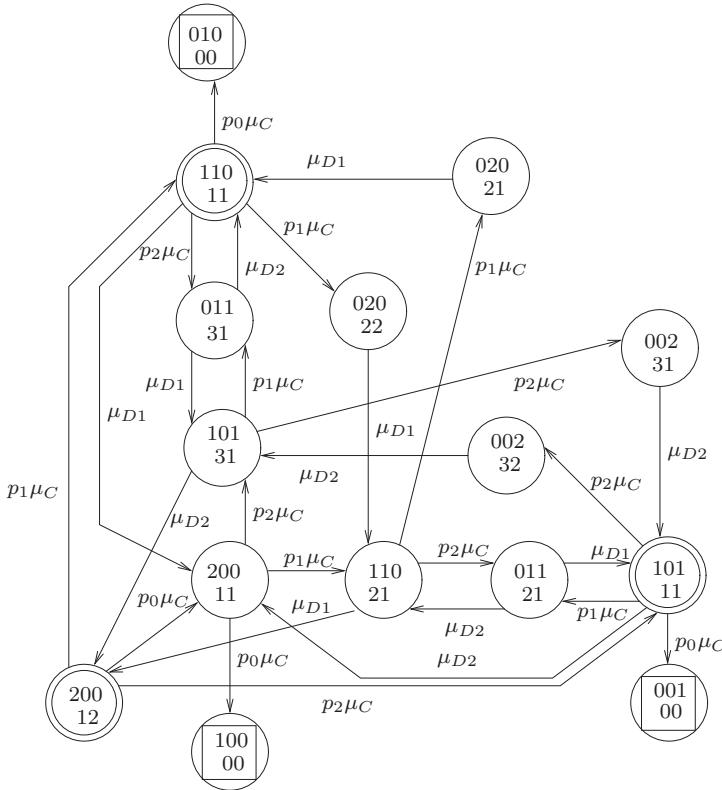
Figure 9.42. Central server model of a computer system

Example 9.21

Consider a central server model (CSM) of a computing system as shown in Figure 9.42. We assume that the service discipline at all the queues is FCFS and the service time distributions are exponential. The service rates of the CPU, Disk1 and Disk2 are μ_C , μ_{D_1} , and μ_{D_2} , respectively. When a customer finishes receiving a burst of service at the CPU, it will request access to Disk1 or Disk2 with probability p_1 and p_2 , respectively. After completing the disk access, the customer rejoins the CPU queue for another burst of service. The customer will complete execution and exit the system with probability $p_0 = 1 - (p_1 + p_2)$. At the same time a statistically identical customer enters the system as indicated by the *new program path* in the figure. We assume that there are n customers in the system. For this model we define the response time as the amount of time elapsed from the instant at which the customer enters the CPU queue for its first service until the instant at which it emerges on the new program path.

As mentioned earlier, the response time distribution can be formulated in terms of the absorption time distribution of a CTMC. Suppose that we need to solve for the response time distribution of the CSM model with two customers. The corresponding CTMC whose absorption time distribution needs to be computed is shown in Figure 9.43. It is interesting to note that even a system with two customers results in a complex CTMC. In this figure, the first three components of the state label correspond to the number of customers in the CPU, Disk1, and Disk2, respectively. The next two components give the position of the tagged customer; the first one is the index of the queue in which the tagged customer is residing and the second corresponds to the position of the tagged customer in the queue. The queue is numbered as follows: 1 (CPU), 2 (Disk1), and 3 (Disk2). Here, 00 indicates that the tagged customer has departed from the system. There are three such absorbing states in the Markov chain, namely, (10000), (01000), and (00100). These states are explicitly identified in the figure by the squares enclosed within the circles.

The tagged customer may arrive into the queuing system in states (20012), (11011), and (10111), which correspond to the other job being at the CPU, Disk1, and Disk2, respectively. These three states are explicitly identified in the figure by double circles. Starting with any of these states as the initial state, we can compute the absorption time distribution of the CTMC. This gives the conditional response time distribution of the tagged customer conditioned on the position of the other customer at the instant of arrival of the tagged customer into



State label : $(i \ j \ k \ l \ m)$

- i : No. of jobs in CPU
- j : No. of jobs in Disk1
- k : No. of jobs in Disk2
- l : Queue in which tagged customer is present
(1=CPU, 2=Disk1, 3=Disk2)
- m : Position of tagged customer in queue
- $l = 0, m = 0$: Job has exited from the system

Figure 9.43. CTMC model of the CSM for computing the response time distribution

the queuing system. To compute the unconditional response time distribution, we need the probabilities of the queuing network being in these states at the instant of arrival of the tagged customer. This is obtained by solving the CTMC shown in Figure 9.44, which has three states corresponding to the nontagged customer being present at the CPU, Disk1, and Disk2, respectively. The three components in the labels of the state correspond to the number of customers at the CPU, Disk1, and Disk2, respectively. In general, suppose that I represents the set of all states in the CTMC whose solution yields the response time distribution. Let A ($\subseteq I$) be the set of absorbing states in the CTMC. Let S ($\subseteq I$) be the set of states in which

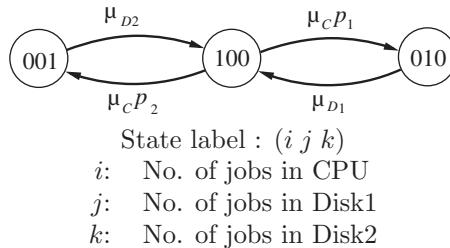


Figure 9.44. CTMC model for computing the steady-state probabilities of the non-tagged customer

the tagged customer will find the network at the instant of arrival. Let R_i be the random variable representing the response time for an arbitrary customer arriving when the queuing network is in state i , where $i \in S$. Then

$$P[R_i \leq t] = \sum_{j \in A} p_{ij}(t),$$

where $p_{ij}(t)$ is the transient probability of state j at time t given that i is the initial state of the CTMC. The unconditional response time distribution can be obtained once we compute $\pi_i(n - 1)$, the probability that the tagged customer will see the network in state i , ($i \in S$) at the instant of arrival. For a closed queuing network, S is the set of all possible states of the network with one less customer. Let R be the random variable representing the unconditional response time distribution. Then

$$\begin{aligned} P[R \leq t] &= \sum_{i \in S} \pi_i(n - 1) P[R_i \leq t] = \sum_{i \in S} \pi_i(n - 1) \sum_{j \in A} p_{ij}(t) \\ &= \sum_{j \in A} \sum_{i \in S} \pi_i(n - 1) p_{ij}(t) = \sum_{j \in A} \pi_j(t). \end{aligned}$$

Here $\pi_j(\tau)$ represents the unconditional transient probability of state j , which is obtained by setting the initial probability of the state i ($\forall i \in S$) of the CTMC to $\pi_i(n - 1)$ and the initial probabilities of all the other states ($i \in I - S$) to zero and solving the CTMC for its transient probability vector at time t . It must be noted that the unconditional response time distribution is directly computed by assigning the initial probabilities for the Markov chain and carrying out the transient analysis only once.

The response time distributions of the central server model for different numbers of customers (5, 10 and 15) are shown in Figure 9.45. In this example, we assume that $\mu_C = 50.0$, $\mu_{D_1} = 30.0$, $\mu_{D_2} = 20.0$, $p_1 = 0.45$, and $p_2 = 0.3$. As expected, a customer has a higher probability of completing by a given time t when the number of customer is smaller since there are fewer customers competing for resources. \sharp

In this section, we have shown how to compute the response time distribution for Markovian queuing networks. The practical question of generation and solution of large CTMCs has been addressed via the use of SPNs [MUPP 1994]. From the Baskett–Chandy–Muntz–Palacios (BCMP) theorem [BASK

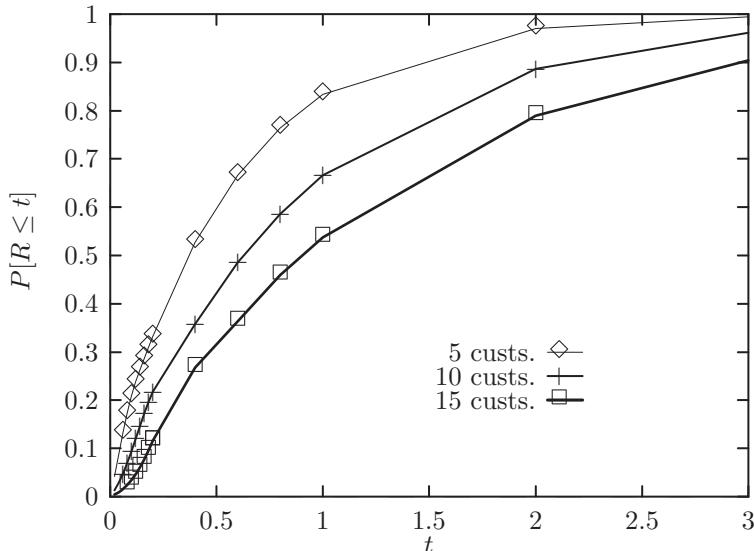


Figure 9.45. Response time distribution for different number of customers

1975], we know that the mean response time for a customer in a queuing network is independent of the service disciplines at queuing centers as long as the service rates remain the same and the service discipline is either FCFS, PS, or LCFS preemptive resume. However, the response time distribution is sensitive to the service discipline. Coffman et al. [COFF 1986] dealt with an open tandem queuing network with two nodes, and Muppala et al. [MUPP 1994] dealt with closed queuing networks, with different service disciplines. For the non-Markovian open networks, we can apply the basic paradigm of decomposition in computing approximations to the response time distribution. For doing so we have made use of existing results on response time distribution at a single queue. Using these, a queuing network is translated into a semi-Markov chain, whose absorption time distribution approximates the response time distribution of the queuing network. We can compute approximations for response time distributions for queuing networks with Poisson or phase-type arrival processes and general service time distributions [MAIN 1994a].

Problems

1. Consider the open central server queuing model of problem 1 in Section 9.2 and compute the response time distribution.
2. Compute the response time distribution for the terminal-oriented distributed computing system of Example 9.10. First construct the Markov chains by hand and solve using SHARPE. Then use stochastic reward nets to solve the problem via SPNP or SHARPE.

9.7 SUMMARY

In this chapter we have introduced networks of queues, which provide an important paradigm for computer communication system performance analysis and prediction.

Open queuing networks are useful in studying the behavior of computer communication networks [KLEI 1976, TAKA 1993]. Closed networks are more useful in computer system performance evaluation [BASK 1975, CHAN 1977, BOLC 1998]. Readers interested in case studies may consult a number of works in the literature [BALB 1988, BRUE 1980, BUZE 1978, GREI 1998, KEIN 1979, KULK 1996, LAVE 1983, MENA 1998, PERR 1994, SAUE 1981, TRIV 1978, WHIT 1983–WOOL 1993].

When queuing networks (open, closed, or mixed) satisfy certain properties (a sufficient condition is known as *local balance* [CHAN 1977]), we can often obtain a convenient product-form solution for the network. However, certain interesting queuing networks do not possess these properties and hence do not have convenient product-form solutions. In these cases we must make use of a number of approximation techniques to obtain a convenient solution. One such technique, the flow-equivalent server method, was illustrated in this chapter by means of two examples. For a deeper study of the approximation techniques, the reader may consult the literature [BOLC 1998, COUR 1977, SAHN 1996, SAUE 1981, TRIP 1979]. Sevcik [SEVC 1977] discusses several approximation techniques for incorporating priority scheduling in multiclass closed queuing networks.

This chapter has only touched on the wide range of results and applications for networks of queues. Extensive treatments of queuing theory have been published [GROS 1998, KELL 1979, KLEI 1975, VAND 1993], as have applications [BOLC 1998, COUR 1977, HAVE 1998, JAIN 1991, KLEI 1976, LAZO 1984, PERR 1994, TAKA 1993, WANG 1996]. A deeper treatment of response time distribution is also available in the literature [BOXM 1990, MAIN 1994a, MUPP 1994, SCHA 1987].

Review Problems

1. *Two-queue blocking system* [HOGA 1975]. Consider the closed cyclic queuing network of Figure 9.9 with the service rates of two servers equal to μ_0 and μ_1 , respectively. Let the branching probabilities $p_0 = 0.1$ and $p_1 = 0.9$. Assume that there is a finite waiting room at node 1 so that the total number of jobs at the node (at its queue plus any at the server) is limited to three jobs. There is no such restriction at node 0. For the degree of multiprogramming $n = 4$, draw the state diagram for the system and proceed to derive an expression for the steady-state probabilities. Repeat for $n = 3, 5, 6, 10$. Compute the average system throughput as a function of the degree of multiprogramming n using $\mu_0 = 1$ and $\mu_1 = 0.5$. Now remove the restriction on the queue size at node 1 and compute the average system throughput as a function of n (this then becomes a case of Example 9.4) and compare the results.

2. * Draw the state diagram and write down the steady-state balance equations for the queuing system $M/E_2/1$ with FCFS scheduling. Solve for the steady-state probabilities. Recall that E_2 implies that the service time distribution is two-stage Erlang.
3. * Recall that in problem 18 at the end of Section 9.4 we assumed a PS discipline for CPU scheduling. Now assume that a preemptive priority is given to interactive jobs over batch jobs. Then the resulting queuing network does not belong to the product-form class. Sevcik [SEVC 1977] has suggested the following approximation technique to solve such problems. We provide separate CPUs to both classes of jobs. The CPU for the batch jobs is known as the “shadow CPU” and it is slowed down by a factor equal to $(1 - U_{0a})$, where U_{0a} is the utilization of the CPU for interactive jobs. Since U_{0a} is not known initially, it may be estimated by solving a single-class queuing network with batch jobs deleted; then an iteration is performed using the abovementioned approximation until convergence is reached. Resolve problem 18 of Section 9.4 using this technique.
4. * Returning to the concurrent program of problem 1 in Section 8.4.2, assume that two independent processes executing the same program are concurrently sharing the resources of a system with a single CPU and a single I/O processor. TCP_{ij} tasks can be executed only on the CPU and TIO_j tasks can be executed only on the I/O processor. Describe the state space of the continuous-parameter Markov chain for this system and draw its state diagram. Solve for the steady-state probabilities and hence obtain expressions for the utilizations of the two devices. Next, assuming that all tasks in a process are to be executed sequentially, solve for the utilizations of the two processors (the problem then becomes a special case of the cyclic queuing model). Using the parameters specified earlier, determine the percentage improvement in resource utilizations due to CPU/IO overlap. Resolve the problem using stochastic reward nets via SPNP or SHARPE.

REFERENCES

- [AKYI 1987] I. Akyildiz, “Exact product form solution for queueing networks with blocking,” *IEEE Trans. Comput.* **36**(1), 122–25 1987.
- [AKYI 1983] I. Akyildiz and G. Bolch, “Erweiterung der mittelwertanalyse zur berechnung der zustandswahrscheinlichkeiten fur geschlossene und gemischte netze,” in *Informatik-Fachberichte*, Springer, Berlin, Vol. **61**, 1983, pp. 267–276.
- [BALB 1988] G. Balbo, S. C. Bruell, and S. Ghanta, “Combining queueing networks and generalized stochastic Petri nets for the solution of complex models of system behavior”, *IEEE Trans. Comput.* **37**(10), 1251–1267 (1988).
- [BASK 1975] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, “Open, closed and mixed networks of queues with different classes of customers,” *J. ACM*, **22**(2), 248–260 (1975).
- [BARD 1979] Y. Bard, “Some extensions to multiclass queueing network analysis,” in M. Arato, A. Butrimenko, and E. Gelenbe (eds.), *Performance of Computer Systems*, North-Holland, Amsterdam, 1979, pp. 51–62.

- [BEUT 1978] F. J. Beutler and B. Melamed, “Decomposition and customer streams of feedback networks of queues in equilibrium,” *Oper. Res.*, **26**(6), 1059–1072 (1978).
- [BOLC 1998] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains—Modeling and Performance Evaluation with Computer Science Applications*, Wiley, New York, 1998.
- [BOXM 1990] O. J. Boxma and H. Daduna, “Sojourn times in queueing networks,” *Stochastic Analysis of Computer and Communication Systems*, in H. Takagi (ed.), Elsevier/North-Holland, Amsterdam, 1990, pp. 401–450.
- [BRUE 1980] S. C. Bruell and G. Balbo, *Computational Algorithms for Closed Queueing Networks*, Elsevier/North-Holland, New York, 1980.
- [BURK 1956] P. J. Burke, “Output of a queueing system,” *Oper. Res.*, **4**, 699–704, (1956).
- [BURK 1976] P. J. Burke, “Proof of a conjecture on the interarrival time distribution in a $M/M/1$ queue with feedback,” *IEEE Trans. Commun.*, **24**, 175–178, (1976).
- [BUZE 1973] J. P. Buzen, “Computational algorithms for closed queueing networks with exponential servers,” *Commun. ACM*, **16**(9), 527–531 (1973).
- [BUZE 1978] J. P. Buzen, “A queueing network model of MVS,” *ACM Comput. Surv.*, **10**(3), 319–331, (1978).
- [CHAN 1975] K. M. Chandy, U. Herzog and L. Woo, “Parametric analysis of queueing networks,” *IBM J. Res. and Devel.*, **19**(1), 36–42 (1975).
- [CHAN 1977] K. M. Chandy, J. H. Howard and D. F. Towsley, “Product form and local balance in queueing networks,” *J. ACM*, **24**(2), 250–263 (1977).
- [CHAN 1983] K. M. Chandy and A. Martin, “A characterization of product-form queueing networks,” *J. ACM*, **30**(2), 286–299 (1983).
- [CHAN 1978] K. M. Chandy and C. H. Sauer, “Approximate methods for analyzing queueing network models of computer systems,” *ACM Comput. Surv.*, **10**, 281–317 (1978).
- [CHAN 1980] K. M. Chandy and C. H. Sauer, “Computational algorithm for product form queueing networks,” *Commun. ACM*, **23**(10), 573–583 (1980).
- [CIAR 1989] G. Ciardo, J. Muppala, and K. S. Trivedi, “SPNP: Stochastic Petri net package,” *Proc. Int. Workshop on Petri Nets and Performance Models*, Los Alamitos, CA, IEEE Computer Society Press, 1989, pp. 142–150.
- [COFF 1986] E. G. Coffman, Jr., G. Fayolle, and I. Mitrani, “Sojourn times in a tandem queue with overtaking: Reduction to a boundary value problem,” *Stochastic Models*, **2**, 43–65 (1986).
- [CONW 1989] A. E. Conway and N. D. Georganas, *Queueing Networks—Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*, The MIT Press, Cambridge, MA, 1989.
- [COUR 1977] P. J. Courtois, *Decomposability: Queueing and Computer System Applications*, Academic Press, New York, 1977.

- [DENN 1978] P. J. Denning and J. P. Buzen, "The operational analysis of queuing network models," *ACM Comput. Surv.*, **10**, 225–261 (1978).
- [DOWD 1979] L. Dowdy *et al.*, On the Multiprogramming Level in Closed Queuing Networks, Technical Report, Department of Computer Science, Univ. Maryland, College Park, MD, 1979.
- [GORD 1967] W. Gordon and G. Newell, "Closed queuing systems with exponential servers," *Oper. Res.*, **15**, 254–265 (1967).
- [GREI 1998] S. Greiner, G. Bolch, and K. Begain, "A generalized analysis technique for queuing networks with mixed priority strategy and class switching," *J. Comput. Commun.*, **21**, 819–832 (1998).
- [GROS 1998] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 3rd ed. Wiley, New York, 1998.
- [HAVE 1998] B. R. Haverkort, *Performance of Computer Communication System: A Model-based Approach*, Wiley, New York, 1998.
- [HAVE 2001] B. R. Haverkort, R. Marie, G. Rubino and K. S. Trivedi, *Performability Modelling: Techniques and Tools*, Wiley, New York, 2001.
- [HILLSTON] J. HILLSTON, A Compositional Approach to Performance
- [HOGA 1975] J. Hogarth, *Optimization and Analysis of Queueing Networks*, Ph.D. dissertation, Department of Computer Science, Univ. Texas, 1975.
- [JACK 1957] J. R. Jackson, "Networks of waiting lines," *Oper. Res.*, **5**, 518–521, (1957).
- [JAIN 1991] R. Jain, *The Art of Computer Systems Performance Analysis*, Wiley, New York, 1991.
- [KEIN 1979] M. G. Keinze and K. C. Sevcik, "A systematic approach to the performance modeling of computer systems," *Proc. 4th Int. Symp. Modeling and Performance Evaluation of Computer Systems*, Vienna, Feb. 1979.
- [KELL 1979] F. P. Kelly, *Reversibility and Stochastic Networks*, Wiley, New York, 1979.
- [KLEI 1975] L. Kleinrock, *Queueing Systems*, Vol. I, *Theory*, Wiley, New York, 1975.
- [KLEI 1976] L. Kleinrock, *Queueing Systems*, Vol. II, *Computer Applications*, Wiley, New York, 1976.
- [KULK 1996] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, Chapman & Hall, London, 1996.
- [LAM 1977] S. Lam, "Queueing networks with population size constraints," *IBM J. Res. and Devel.*, **21**(4), 370–378 (1977).
- [LAVE 1983] S. S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press, New York, 1983.
- [LAVE 1980] S. S. Lavenberg and M. Reiser, "Stationary state probabilities of arrival instants for closed queuing networks with multiple types of customers," *J. Appl. Probability*, **1**, 1048–1061 (1980).

- [LAZO 1984] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance, Computer System Analysis using Queueing Network Models*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
- [MAIN 1994a] V. Mainkar, *Solutions of Large and Non-Markovian Performance Models*, Ph.D. dissertation, Department of Computer Science, Duke Univ., Durham, NC, 1994.
- [MAIN 1994b] V. Mainkar, S. Woonlet, and K. S. Trivedi, "Fast approximate computation of response time distribution in open Markovian network of queues," *Proc. 17th Int. Conf. Modeling Techniques and Tools for Computer Performance Evaluation*, 1994, pp. 67–70.
- [MELA 1984] B. Melamed and M. Yadin, "Numerical computation of sojourn-time distributions in queuing networks," *J. ACM*, **31**(9), 839–854 (1984).
- [MENA 1998] D. A. Menasce and V. A. F. Almeida, *Capacity Planning for Web Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [MUPP 1994] J. Muppala, K. S. Trivedi, V. Mainkar, and V. G. Kulkarni, "Numerical computation of response time distributions using stochastic reward nets," *Ann. Oper. Res.*, **48**, 155–184 (1994).
- [PERR 1994] H. Perros, *Queueing Networks with Blocking*, Oxford Univ. Press, Oxford, 1994.
- [REIS 1980] M. Reiser and S. S. Lavenberg, "Mean-value analysis of closed multichain queuing networks," *J. ACM*, **27**(2), 313–322 (1980).
- [SAHN 1996] R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer Systems—An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.
- [SAUE 1981] C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [SCHA 1987] R. Schassberger and H. Daduna, "Sojourn times in queuing networks with multiserver modes," *J. Appl. Probability*, **24**, 511–521 (1987).
- [SEVC 1977] K. C. Sevcik, "Priority scheduling disciplines in queueing network models of computer systems," *Proc. IFIP Congress*, IFIP Press, Toronto, Canada, 1977, pp. 565–570.
- [SEVC 1980] K. C. Sevcik, G. S. Graham, and J. Zahorjan, "Configuration and capacity planning in a distributed processing system," *Proc. Computer Performance Evaluation Users Group Meeting*, Orlando, FL, 1980.
- [SEVC 1981] K. C. Sevcik and I. Mitrani, "The distribution of queueing network states at input and output instants," *J. ACM*, **28**, 358–371 (1981).
- [SIMO 1979] B. Simon and R. D. Foley, "Some results on sojourn times in acyclic Jackson networks," *Manage. Sci.*, **25**(10), 1027–1034 (1979).
- [TAKA 1993] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation*, Vols. 1–3, North-Holland, Amsterdam, 1991–1993.

- [TRIP 1979] S. K. Tripathi, *On Approximate Solution Techniques for Queueing Network Models of Computer Systems*, Computer Systems Research Group, Technical Report, Univ. Toronto, Canada, 1979.
- [TRIV 1978] K. S. Trivedi and R. L. Leech, “The design and analysis of a functionally distributed computer system,” *1978 Int. Conf. on Parallel Processing*, Michigan, Aug. 1978.
- [VAND 1993] N. M. Van Dijk, *Queueing Networks and Product form: A Systems Approach*, Wiley, New York, 1993.
- [WANG 1996] C. Wang, D. Logothetis, K. S. Trivedi, and I. Viniotis, “Transient behavior of ATM networks under overloads,” *Proc. IEEE INFOCOM '96*, San Francisco, CA, 1996, pp. 978–985.
- [WANG 2000] H. Wang and K. C. Sevcik, “Experiments with improved approximate mean value analysis algorithms,” *Perform. Evaluation*, **39**(2), 189–206 (2000).
- [WHIT 1983] W. Whitt, “The queuing network analyzer,” *Bell Syst. Tech. J.*, **62**(9), 2779–2815 (1983).
- [WILL 1976] A. C. Williams and R. A. Bhandiwad, “A generating function approach to queueing network analysis of multiprogrammed computers,” *Networks*, **6**, 1–22 (1976).
- [WOOL 1993] S. Woole, *Performance Analysis of Computer Networks*, Ph.D. dissertation, Department of Electrical Engineering, Duke Univ., Durham, NC, 1993.
- [ZAHO 1981] J. Zahorjan and E. Wong, “The solution of separable queueing network models using mean value analysis,” *ACM Sigmetrics Perform. Evaluation Rev.*, **10**(3), 80–85 (1981).

Chapter 10

Statistical Inference

10.1 INTRODUCTION

The probability distributions discussed in the preceding chapters will yield probabilities of the events of interest, provided that the family (or the type) of the distribution and the values of its parameters are known in advance. In practice, the family of the distribution and its associated parameters have to be estimated from data collected during the actual operation of the system under investigation.

In this chapter we investigate problems in which, from the knowledge of some characteristics of a suitably selected subset of a collection of elements, we draw inferences about the characteristics of the entire set. The collection of elements under investigation is known as the **population**, and its selected subset is called a **sample**. Methods of **statistical inference** help us in estimating the characteristics of the entire population based on the data collected from (or the evidence produced by) a sample. Statistical techniques are useful in both planning of the measurement activities and interpretation of the collected data.

Two aspects of the sampling process seem quite intuitive. First, as the sample size increases, the estimate generally gets closer to the “true” value, with complete correspondence being reached when the sample embraces the entire population. Second, whatever the sample size, the sample should be representative of the population. These two desirable aspects (not always satisfied) of the sampling process will lead us to definitions of the **consistency** and **unbiasedness** of an estimate.

When we say that the population has the distribution $F(x)$, we mean that we are interested in studying a characteristic X of the elements of this

population and that this characteristic X is a random variable whose distribution function is $F(x)$.

The following issues will occupy us in this chapter:

1. Different samples from the same population will result, in general, in distinct estimates, and these estimates themselves will follow some form of statistical distribution, called a **sampling distribution**.
2. Assuming that the distributional form (e.g., normal or exponential), of the parent population is known, unknown parameters of the population may be estimated. One also needs to define the confidence in such estimates. This will lead us to **interval estimates**, or **confidence intervals**.
3. In cases where the distributional form of the parent population is not known, we can perform a **goodness-of-fit** test against some specified family of distributions and thus determine whether the parent population can reasonably be declared to belong to this family.
4. Instead of estimating properties of the population distribution, we may be interested in **testing a hypothesis** regarding a relationship involving properties of the distribution function. Based on the collected data, we will perform statistical tests and either reject or fail to reject the hypothesis. We will also study the errors involved in such judgments.

Statistical techniques are extremely useful in algorithm evaluation, system performance evaluation, and reliability estimation. Suppose that we want to experimentally evaluate the performance of some algorithm A with the input space S . Since the input space S is rather large, we execute the algorithm and observe its behavior for some randomly chosen subset of the input space. On the basis of this experiment we wish to estimate the properties of the random variable, T , denoting the execution time of algorithm A . It usually suffices to estimate some parameter of T such as its mean $E[T]$ or its variance $\text{Var}[T]$.

Assume that the interarrival times of jobs coming to a server are known to be exponentially distributed with parameter λ but the value of λ is unknown. After observing the arrival process for some finite time, we could obtain an estimate $\hat{\lambda}$ of λ . Sometimes we may not be interested in estimating the value of λ but wish to test the hypothesis $\lambda < \lambda_0$ against an alternative hypothesis $\lambda > \lambda_0$. Thus, if λ_0 represents some threshold of job arrival rate beyond which the server becomes overloaded or unstable, then the acceptance of the preceding hypothesis will imply that no new server needs to be purchased. At other times we may wish to test the hypothesis that job interarrival times are exponentially distributed.

As another example of a statistical problem arising in system performance evaluation, assume we are interested in making a purchase decision between two servers on the basis of their response times (respectively denoted by R_1 and R_2) to trivial requests (such as a simple editing command). Let the corresponding mean response times to trivial requests be denoted by θ_1 and θ_2 . On the basis of measurement results, we would like to test the hypothesis $\theta_1 = \theta_2$

versus the alternative $\theta_1 < \theta_2$ (or the alternative $\theta_1 > \theta_2$). We may also wish to test a hypothesis on variances: $\text{Var}[R_1] = \text{Var}[R_2]$ versus the alternative $\text{Var}[R_1] < \text{Var}[R_2]$ (or the alternative $\text{Var}[R_1] > \text{Var}[R_2]$).

Now suppose that we are interested in tuning a cluster by varying the parameters associated with resource schedulers. In this case we may want to investigate the functional dependence of the average response time on these parameters. We may want to further investigate the functional dependence of the average response time on various characteristics of the transaction such as its CPU time requirement, number of disk I/O requests, number of users logged on, and so on. In this case we will collect a set of measurements and perform a **regression analysis** to estimate and characterize the functional relationships.

As a last example, consider a common method of reliability estimation known as *life testing*. A random sample of n components is taken and the times to failure of these components are observed. On the basis of these observed values, the mean life of a component can be estimated or a hypothesis concerning the mean life may be tested.

First we discuss problems of estimating parameters of the distribution of a single random variable X . Next we discuss hypothesis testing. In the next chapter we discuss problems involving more than one random variable and the associated topic of regression analysis.

10.2 PARAMETER ESTIMATION

Suppose that the parent population is distributed in a form that is completely determinate except for the value of some parameter θ . The parameter θ being estimated could be the population (or true) mean $\mu = E[X]$ or the population (or true) variance $\sigma^2 = \text{Var}[X]$. The estimation will be based on a collection of n experimental outcomes x_1, x_2, \dots, x_n . Each experimental outcome x_i is a value of a random variable X_i . The set of random variables X_1, X_2, \dots, X_n is called a **sample** of size n from the population.

Definition (Random Sample). The set of random variables X_1, X_2, \dots, X_n is said to constitute a **random sample** of size n from the population with the distribution function $F(x)$, provided that they are mutually independent and identically distributed with distribution function $F_{X_i}(x) = F(x)$ for all i and for all x .

Note that this definition does not hold for sampling without replacement from a finite population (of size N), since the act of drawing an object changes the characteristics of the population. In this case the requirement of independence in this definition is replaced by the following requirement:

$$\begin{aligned} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) \\ = \frac{1}{N} \cdot \frac{1}{N-1} \cdot \dots \cdot \frac{1}{N-n+1} = \frac{(N-n)!}{N!}. \end{aligned}$$

Unless otherwise specified, we will assume that the population is very large or conceptually infinite, so that this definition of random sampling is applicable.

In general, we will want to obtain some desired piece of information about the population from a random sample. If we are lucky, the information may be obtained by direct examination or by pictorial methods. However, it is usually necessary to reduce the set of observations to a few meaningful quantities.

Definition (Statistic). Any function $W(X_1, X_2, \dots, X_n)$ of the observations X_1, X_2, \dots, X_n , is called a **statistic**.

Thus, a statistic is a function of n random variables, and assuming that it is also a random variable, its distribution function, called the **sampling distribution** of W , can be derived from the population distribution. The types of functions we will be interested in include the sample mean

$$\bar{X} = \sum_{i=1}^n \frac{X_i}{n},$$

and the sample variance S^2 (to be defined later).

Definition (Estimator). Any statistic $\hat{\Theta} = \hat{\Theta}(X_1, X_2, \dots, X_n)$ used to estimate the value of a parameter θ of the population is called an **estimator** of θ . An observed value of the statistic $\hat{\theta} = \hat{\theta}(x_1, x_2, \dots, x_n)$ is known as an **estimate** of θ .

A statistic $\hat{\Theta}$ cannot be guaranteed to give a close estimate of θ for every sample. We must design statistics that will give good results “on the average” or “in the long run.”

Definition (Unbiasedness). A statistic $\hat{\Theta} = \hat{\Theta}(X_1, X_2, \dots, X_n)$ is said to constitute an **unbiased** estimator of parameter θ provided

$$E[\hat{\Theta}(X_1, X_2, \dots, X_n)] = \theta.$$

In other words, on the average, the estimator is on target.

Example 10.1

The sample mean \bar{X} is an unbiased estimator of the population mean μ whenever the latter exists:

$$\begin{aligned} E[\bar{X}] &= E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] \\ &= \frac{1}{n} \sum_{i=1}^n E[X_i] \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{n} \sum_{i=1}^n E[X] \\
&= \frac{1}{n} nE[X] \\
&= E[X] \\
&= \mu.
\end{aligned}$$

We can also compute the variance of the sample mean (assuming that the population variance is finite) by noting the independence of X_1, X_2, \dots, X_n as

$$\begin{aligned}
\text{Var}[\bar{X}] &= \sum_{i=1}^n \text{Var}\left[\frac{X_i}{n}\right] \\
&= \frac{n\text{Var}[X_i]}{n^2} \\
&= \frac{\text{Var}[X]}{n} \\
&= \frac{\sigma^2}{n}.
\end{aligned}$$

This implies that the accuracy of the sample mean as an estimator of the population mean increases with the sample size n when the population variance is finite.

‡

If the population distribution is Cauchy, so that

$$f_X(x) = \frac{1}{\pi[1 + (x - \theta)^2]}, \quad -\infty < x < \infty,$$

then, by problem 1 at the end of this section, it follows that the sample mean \bar{X} is also Cauchy distributed. If \bar{X} is used to estimate the parameter θ , then it does not increase in accuracy as n increases. Note that in this case neither the population mean nor the population variance exists.

If we take a sample of size n without replacement from a finite population of size N , then the sample mean \bar{X} is still an unbiased estimator of the population mean but the $\text{Var}[\bar{X}]$ is no longer given by σ^2/n . In this case, it can be shown [KEND 1961] that

$$\text{Var}[\bar{X}] = \frac{\sigma^2}{n} \left(1 - \frac{n}{N}\right) \left(\frac{N}{N-1}\right).$$

Note that as the population size N approaches infinity, we get the formula σ^2/n for the variance of the sample mean.

Next consider the function

$$\hat{\Theta}(X_1, X_2, \dots, X_n) = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}$$

as an estimator of the population variance. It can be seen that this function provides a biased estimator of the population variance. On the other hand, the function

$$\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

is an unbiased estimator of the population variance. These two functions differ little when the sample size is relatively large.

Example 10.2

The sample variance S^2 , defined by

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2,$$

is an unbiased estimator of the population variance σ^2 whenever the latter exists. This can be shown as follows:

$$\begin{aligned} S^2 &= \frac{1}{n-1} \sum_{i=1}^n (X_i^2 - 2X_i\bar{X} + \bar{X}^2) \\ &= \frac{1}{n-1} \left(\sum_{i=1}^n X_i^2 \right) - \frac{2n}{n-1} \left(\frac{1}{n} \sum_{i=1}^n X_i \right) \bar{X} + \frac{n\bar{X}^2}{n-1} \\ &= \frac{1}{n-1} \sum_{i=1}^n X_i^2 - \frac{n}{n-1} \bar{X}^2. \end{aligned}$$

Therefore

$$E[S^2] = \frac{1}{n-1} \sum_{i=1}^n E[X_i^2] - \frac{n}{n-1} E[\bar{X}^2].$$

But

$$E[X_i^2] = \text{Var}[X_i] + (E[X_i])^2 = \sigma^2 + \mu^2$$

and

$$E[\bar{X}^2] = \text{Var}[\bar{X}] + (E[\bar{X}])^2 = \frac{\sigma^2}{n} + \mu^2.$$

Thus

$$\begin{aligned} E[S^2] &= \frac{1}{n-1} n(\sigma^2 + \mu^2) - \frac{n}{n-1} \left(\frac{\sigma^2}{n} + \mu^2 \right) \\ &= \sigma^2. \end{aligned}$$

Thus the sample variance S^2 is an unbiased estimator of the population variance σ^2 . ‡

The preceding formula applies to the case of an infinite population. The unbiased estimator of the variance of a finite population of size N (assuming sampling without replacement) is given by

$$S^2 = \frac{1 - \frac{1}{N}}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2.$$

Unbiasedness is one of the most desirable properties of an estimator, although not essential. This criterion by itself does not provide a unique estimator for a given estimation problem as shown by the following example.

Example 10.3

The formula

$$\hat{\Theta} = \sum_{i=1}^n a_i X_i$$

is an unbiased estimator of the population mean μ (if it exists) for any set of real weights a_i such that $\sum a_i = 1$. ‡

Thus we need another criterion to enable us to choose the best among all unbiased estimators. For an estimator $\hat{\Theta}$ of parameter θ to be a good estimator, we would like the probability of the dispersion $P(|\hat{\Theta} - \theta| \geq \epsilon)$ to be small. We note that for an unbiased estimator, $E[\hat{\Theta}] = \theta$, Chebyshev's inequality gives us

$$P(|\hat{\Theta} - \theta| \geq \epsilon) \leq \frac{\text{Var}[\hat{\Theta}]}{\epsilon^2}, \quad \text{for } \epsilon > 0.$$

Thus, one way of comparing two unbiased estimators is to compare their variances.

Definition (Efficiency). An estimator $\hat{\Theta}_1$ is said to be a more **efficient** estimator of the parameter θ than the estimator $\hat{\Theta}_2$, provided that

1. $\hat{\Theta}_1$ and $\hat{\Theta}_2$ are both unbiased estimators of θ , and
2. $\text{Var}[\hat{\Theta}_1] \leq \text{Var}[\hat{\Theta}_2]$, for all θ .
3. $\text{Var}[\hat{\Theta}_1] < \text{Var}[\hat{\Theta}_2]$, for some θ .

Example 10.4

The sample mean

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

is the most efficient (minimum-variance) linear estimator of the population mean, whenever the latter exists. To show this, we first note that

$$\begin{aligned}\text{Var}[\hat{\Theta}] &= \text{Var} \left[\sum a_i X_i \right] = \sum_{i=1}^n a_i^2 \text{Var}[X_i] \\ &= \sum a_i^2 \text{Var}[X] \\ &= \sum a_i^2 \sigma^2,\end{aligned}$$

since X_1, X_2, \dots, X_n are mutually independent and identically distributed. Thus we solve the following optimization problem:

$$\min : \text{Var}[\hat{\Theta}] = \sigma^2 \sum_{i=1}^n a_i^2 \quad \text{s.t.: } \sum_{i=1}^n a_i = 1, \quad i = 1, 2, \dots, n.$$

We can solve this problem using the method of Lagrange multipliers to obtain the following result:

$$a_i = \frac{1}{n}, \quad i = 1, 2, \dots, n.$$

In other words, the estimator minimizing the variance is

$$\sum a_i X_i = \bar{X}, \quad \text{the sample mean.}$$

#

It can also be shown that under some mild conditions the sample variance, S^2 , is a minimum-variance (quadratic) unbiased estimator of the population variance σ^2 , whenever the latter exists. Thus, in most practical situations, the sample mean and the sample variance are acceptable estimators of μ and σ^2 , respectively.

As in the case of the sample mean, the variance of the sampling distribution of an estimator generally decreases with increasing n . This leads us to another desirable property of an estimator.

Definition (Consistency). An estimator $\hat{\Theta}$ of parameter θ is said to be consistent if $\hat{\Theta}$ converges in probability to θ :

$$\lim_{n \rightarrow \infty} P(|\hat{\Theta} - \theta| \geq \epsilon) = 0.$$

As the sample size increases, a consistent estimator gets close to the true value. If we consider a population with finite mean and variance, then $\text{Var}[\bar{X}] = \sigma^2/n$, and using the Chebyshev inequality, we conclude that the sample mean is a consistent estimator of the population mean. In fact, any unbiased estimator $\hat{\Theta}$ of θ , with the property

$$\lim_{n \rightarrow \infty} \text{Var}[\hat{\Theta}] = 0$$

is a consistent estimator of θ owing to the Chebyshev inequality.

The data collected in a sample may be summarized by the arithmetic methods (sample mean, sample variance, etc.) discussed so far. Alternative methods are pictorial in nature. For example, a bar chart or a histogram is often used. Yet another useful way of summarizing data is to construct an **empirical distribution function**, $\hat{F}(x)$; let k_x be the number of observed values x_i (out of a total of n values) that are less than or equal to x ; then $\hat{F}(x) = k_x/n$.

The empirical distribution function is a consistent estimator of the true distribution function $F(x)$. To show this, let us perform n independent trials of the event, “Sample value observed is less than or equal to x .” Each observation y_i is then a value of a Bernoulli random variable Y_i with the probability of success $p = F(x)$, so that

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i = \hat{F}(x)$$

and $E[\bar{Y}] = p$. Now the law of large numbers tells us that

$$\lim_{n \rightarrow \infty} P(|\bar{Y} - E[\bar{Y}]| \geq \epsilon) = 0.$$

But here $\bar{Y} = \hat{F}(x)$ and $E[\bar{Y}] = E[Y] = p = F(x)$, so

$$\lim_{n \rightarrow \infty} P[|\hat{F}(x) - F(x)| \geq \epsilon] = 0,$$

as desired.

Next we discuss two general methods of parameter estimation.

Problems

- Given that the population X has the Cauchy distribution, show that the sample mean \bar{X} has the same distribution.

2. Solve the optimization problem posed in Example 10.4 using the method of Lagrange multipliers. To obtain an algebraic proof, show that

$$\sum_{i=1}^n a_i^2 \geq \sum_{i=1}^n \left(\frac{1}{n}\right)^2$$

10.2.1 The Method of Moments

Suppose that one or more parameters of the distribution of X are to be estimated on the basis of a random sample of size n . Define the k th sample moment of the random variable X as

$$M'_k = \sum_{i=1}^n \frac{X_i^k}{n}, \quad k = 1, 2, \dots$$

Of course, the k th population moment is

$$\mu'_k = E[X^k], \quad k = 1, 2, \dots$$

and this moment will be a function of the unknown parameters.

The **method of moments** consists of equating the first few population moments with the corresponding sample moments to obtain as many equations as there are unknown parameters and then solving these equations simultaneously to obtain the required estimates. This method usually yields fairly simple estimators that are consistent. However, it can give estimators that are biased (see problem 1 at the end of this section) and inefficient.

Example 10.5

Let X denote the main-memory requirement of a job as a fraction of the total user-allocatable main memory of a compute server. We suspect that the density function of X has the form

$$f(x) = \begin{cases} (k+1)x^k, & 0 < x < 1, k > 0 \\ 0, & \text{otherwise.} \end{cases}$$

A large value of k implies a preponderance of large jobs. If $k = 0$, the distribution of memory requirement is uniform. We have a sample of size n from which we wish to estimate the value of k . Since one parameter is to be estimated, only the first moments need to be considered:

$$\mu'_1 = \int_0^1 (k+1)x^k x dx = \frac{k+1}{k+2}$$

$$M'_1 = \sum_{i=1}^n \frac{X_i}{n} = \text{sample mean, } \bar{X}.$$

Then the required estimate k is obtained by letting

$$M'_1 = \mu'_1$$

and hence

$$\hat{k} = \frac{2M'_1 - 1}{1 - M'_1} = \frac{2\bar{X} - 1}{1 - \bar{X}}.$$

As a numerical example, we are given the following sample of size 8:

0.25 0.45 0.55 0.75 0.85 0.85 0.95 0.90

The sample mean $M'_1 = 5.55/8 = 0.69375$. Thus $k = 1.265306$.

#

Example 10.6

Assume that the repair time of a server has a gamma distribution with parameters λ and α . This could be suggested, for instance, by the sequential (stage-type) nature of the repair process. After taking a random sample of n actual repair times, we compute the first two sample moments M'_1 and M'_2 . Now the corresponding population moments for a gamma distribution are given by

$$\mu'_1 = \frac{\alpha}{\lambda} \quad \text{and} \quad \mu'_2 = \frac{\alpha}{\lambda^2} + \mu'^{'}_1.$$

Then the estimates $\hat{\lambda}$ and $\hat{\alpha}$ can be obtained by solving

$$\frac{\hat{\alpha}}{\hat{\lambda}} = M'_1 \quad \text{and} \quad \frac{\hat{\alpha}}{\hat{\lambda}^2} + \frac{\hat{\alpha}^2}{\hat{\lambda}^2} = M'_2.$$

Hence

$$\hat{\alpha} = \frac{M'^2_1}{M'_2 - M'^2_1} \quad \text{and} \quad \hat{\lambda} = \frac{M'_1}{M'_2 - M'^2_1}.$$

#

Problems

1. Show that the method-of-moments estimators of the population mean and of the population variance are given by the sample mean, \bar{X} and $(n - 1)S^2/n$, respectively. Show that the method-of-moments estimator of the population variance is **biased**.
2. Consider the problem of deriving method-of-moments estimates for the three parameters α , λ_1 , and λ_2 of a two-stage hyperexponential distribution. Clearly, three sample moments will be needed for this purpose. But if we have only the sample mean \bar{x} and the sample variance s^2 available, we can solve the problem

by imposing a restriction on the parameters. Assume that

$$\frac{1}{\lambda_1 + \lambda_2} = \frac{\frac{\alpha}{\lambda_1} + \frac{1-\alpha}{\lambda_2}}{2}$$

is the chosen restriction. Show that the method-of-moments estimates of the parameters are given by (assuming that $s^2 \geq \bar{x}^2$)

$$\hat{\lambda}_1, \hat{\lambda}_2 = \frac{1}{\bar{x}} \pm \frac{1}{\bar{x}} \sqrt{\frac{s^2 - \bar{x}^2}{s^2 + \bar{x}^2}} \quad (10.1)$$

and

$$\hat{\alpha} = \frac{\hat{\lambda}_1(\hat{\lambda}_2\bar{x} - 1)}{\hat{\lambda}_2 - \hat{\lambda}_1}. \quad (10.2)$$

3. The memory residence times of 13,171 jobs were measured, and the sample mean was found to be 0.05 s and the sample variance, 0.006724. Estimate the parameters α and λ using the method of moments, assuming that the memory residence time is gamma-distributed. Using the result of problem 2, obtain the method-of-moments estimates for parameters α, λ_1 , and λ_2 , assuming that the memory residence time possesses a two-stage hyperexponential distribution.
4. Show that method-of-moments estimates for the parameters λ_1 and λ_2 of a two-stage hypoexponential distribution are given by

$$\lambda_1, \lambda_2 = \frac{\frac{2}{\bar{x}}}{1 \pm \sqrt{1 + 2\left\{\left(\frac{s}{\bar{x}}\right)^2 - 1\right\}}} \quad (10.3)$$

10.2.2 Maximum-Likelihood Estimation

The method of maximum-likelihood produces estimators that are usually consistent and under certain regularity conditions can be shown to be most efficient in an asymptotic sense (i.e., as the sample size n approaches infinity). However, the estimators may be biased for small sample sizes. The principle of this method is to select as an estimate of θ the value for which the observed sample is most “likely” to occur.

We introduce this method through an example. Suppose that we want to estimate the probability, p , of a successful transmission of a message over a communication channel. We observe the transmission of n messages and observe that k have been transmitted without errors. From these data, we wish to obtain a maximum-likelihood estimate of the parameter p .

The transmission of a single message is modeled by a Bernoulli random variable X with the pmf.

$$p_X(x) = p^x(1-p)^{1-x}, \quad x = 0, 1; \quad 0 \leq p \leq 1.$$

A random sample X_1, X_2, \dots, X_n is taken, and the problem is to find an estimator $W(X_1, X_2, \dots, X_n)$ such that $w(x_1, x_2, \dots, x_n)$ is a good estimate of p , where x_1, x_2, \dots, x_n are the observed values of the random sample. The joint probability that X_1, X_2, \dots, X_n take these values is given by the compound pmf:

$$\begin{aligned} P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) &= \prod_{i=1}^n p^{x_i} (1-p)^{1-x_i} \\ &= p^{\sum x_i} (1-p)^{n-\sum x_i}. \end{aligned}$$

If we fix n and the observed values x_1, x_2, \dots, x_n in this pmf, then it can be considered a function of p , called a **likelihood function**:

$$L(p) = p^{\sum x_i} (1-p)^{n-\sum x_i}, \quad 0 \leq p \leq 1.$$

The value of p , say, \hat{p} , maximizing $L(p)$, is the maximum-likelihood estimate of p . Thus this method selects the value of the unknown parameter for which the probability of obtaining the measured data is maximum, and \hat{p} is the “most likely” value of p .

Maximizing $L(p)$ is equivalent to maximizing the natural logarithm of $L(p)$:

$$\ln L(p) = \left(\sum_{i=1}^n x_i \right) \ln p + \left(n - \sum_{i=1}^n x_i \right) \ln(1-p).$$

To find the maximum of this function, when $0 < p < 1$, we take the first derivative and set it equal to zero:

$$\frac{d \ln L(p)}{dp} = \left(\sum_{i=1}^n x_i \right) \left(\frac{1}{p} \right) + \left(n - \sum_{i=1}^n x_i \right) \left(\frac{-1}{1-p} \right) = 0$$

and get

$$p = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}.$$

Verifying that the second derivative of $\ln L(p)$ is negative, we conclude that $p = \bar{x}$ actually maximizes $\ln L(p)$. Therefore, the statistic $\sum X_i/n = \bar{X}$ is known as the **maximum-likelihood estimator of p** :

$$\hat{P} = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}.$$

Since X_i is 0 for a garbled message and 1 for a successful transmission, the sum $\sum X_i$ is the number of successfully transmitted messages. Therefore, this

estimator of p is simply the one we would get by using a relative-frequency argument.

More generally for a discrete population, we define the likelihood function as the joint probability of the event, $[X_1 = x_1, X_2 = x_2, \dots, X_n = x_n]$:

$$L(\boldsymbol{\theta}) = P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n | \boldsymbol{\theta}) = \prod_{i=1}^n p_{X_i}(x_i | \boldsymbol{\theta})$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ is the vector of parameters to be estimated. Analogously, in the case of a continuous population, the likelihood function is defined as the product of the marginal densities:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n f_{X_i}(x_i | \boldsymbol{\theta})$$

Thus the likelihood function is the joint pmf or pdf of the random variables X_1, X_2, \dots, X_n . Under certain regularity conditions, the maximum-likelihood estimate of $\boldsymbol{\theta}$ is the solution of the simultaneous equations

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_i} = 0, \quad i = 1, 2, \dots, k.$$

This method usually works quite well, but sometimes difficulties arise. There may be no closed-form solution to the preceding system of equations. For example, if we wish to estimate the parameters α and λ of a gamma distribution, the method of maximum likelihood will produce two simultaneous equations that are impossible to solve in closed form. Alternatively, there may be no unique solution to the system of equations presented above. In this case it is necessary to verify which solution, if any, maximizes the likelihood function. Another possibility is that the solution to the preceding system may not be in the parameter space, in which case a constrained maximization of the likelihood function becomes necessary.

Example 10.7

It is desired to estimate the arrival rate of new calls to a cell in a mobile communication system, based on a random sample $X_1 = x_1, \dots, X_n = x_n$, where X_i denotes the number of calls per hour in the i th observation period. Let the number of calls per hour, X , be Poisson distributed with parameter λ :

$$p_X(x|\lambda) = e^{-\lambda} \frac{\lambda^x}{x!}.$$

The likelihood function is then

$$L(\lambda) = \prod_{i=1}^n e^{-\lambda} \frac{\lambda^{x_i}}{x_i!} = \frac{1}{x_1! x_2! \cdots x_n!} e^{-n\lambda} \lambda^{\sum_{i=1}^n x_i}$$

Taking logs, we have

$$\ln L(\lambda) = -\ln(x_1!x_2!\cdots x_n!) - n\lambda + \left(\sum_{i=1}^n x_i\right) \ln(\lambda).$$

Taking the derivative with respect to λ and setting it equal to zero, we get

$$-n + \frac{1}{\lambda} \sum_{i=1}^n x_i = 0.$$

Thus the maximum-likelihood estimator of the arrival rate is the sample mean:

$$\hat{\Lambda} = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}. \quad (10.4)$$

‡

A common method of estimating parameters related to component (system) reliability is that of life testing. This consists of selecting a random sample of n components, testing them under specified environmental conditions, and observing the time to failure of each component.

Example 10.8

Assume that the time to failure, X , of a telephone switching system is exponentially distributed with a failure rate λ . We wish to estimate the failure rate λ from a random sample of n times to failure. Then

$$L(\lambda) = \prod_{i=1}^n \lambda e^{-\lambda x_i} = \lambda^n e^{-\lambda \sum_{i=1}^n x_i}$$

$$\frac{dL}{d\lambda} = n\lambda^{n-1} e^{-\lambda \sum_{i=1}^n x_i} - \left(\sum_{i=1}^n x_i \right) \lambda^n e^{-\lambda \sum_{i=1}^n x_i} = 0,$$

from which we get the maximum-likelihood estimator of the failure rate to be the reciprocal of the sample mean:

$$\hat{\Lambda} = \frac{n}{\sum_{i=1}^n X_i} = \frac{1}{\bar{X}}. \quad (10.5)$$

The corresponding maximum-likelihood estimator of the mean life (MTTF) is equal to the sample mean \bar{X} .

‡

Usually, the MTTF (mean time to failure) is so large as to forbid such exhaustive life tests; hence **truncated (censored) life tests** are

common. Such a life test is terminated after the first r failures have occurred (sample-truncated, or type II) or after a specified time has elapsed (time-truncated, or type I). If a failed component is repaired or is replaced by a new one, then the life test is called a **replacement test**; otherwise it is a **nonreplacement test**.

Example 10.9

Consider a sample truncated test of n components without replacement. Let T_1, T_2, \dots, T_r be the observed times to failure so that $T_1 \leq T_2 \leq \dots \leq T_r$. Specific values of these random variables are denoted by t_1, t_2, \dots, t_r . Let θ be the MTTF to be estimated and assume that components follow an exponential failure law.

Since $(n - r)$ components have not failed when the test is completed, the likelihood function is defined in the following way. Assume T_{r+1}, \dots, T_n are the times to failure of the remaining components, whose failures will not actually be observed. Then

$$L(\theta) \prod_{i=1}^r h_i = P(t_i \leq T_i < t_i + h_i, i = 1, 2, \dots, r; T_i > t_r, i = r+1, \dots, n)$$

and, dividing by the product of h_i 's and taking the limit as $h_i \rightarrow 0$, we get

$$\begin{aligned} L(\theta) &= \prod_{i=1}^r f(t_i|\theta) \prod_{j=r+1}^n [1 - F(t_r|\theta)] \\ &= \prod_{i=1}^r \frac{1}{\theta} e^{-t_i/\theta} \prod_{j=r+1}^n e^{-t_r/\theta} \\ &= \frac{1}{\theta^r} \exp \left[-\frac{(\sum_{i=1}^r t_i) + (n-r)t_r}{\theta} \right]. \end{aligned}$$

Let

$$s_{n;r} = \left(\sum_{i=1}^r t_i \right) + (n-r)t_r$$

be the accumulated life on test. Differentiating the likelihood function with respect to θ and setting it equal to zero, we get

$$-\frac{r}{\theta^{r+1}} e^{-s_{n;r}/\theta} + \frac{1}{\theta^r} (-s_{n;r}) \left(-\frac{1}{\theta^2} \right) e^{-s_{n;r}/\theta} = 0.$$

Then the maximum-likelihood estimator (MLE) of the mean life is given by

$$\hat{\Theta} = \frac{S_{n;r}}{r} = \frac{(\sum_{i=1}^r T_i) + (n-r)T_r}{r}.$$

Thus the estimator of the mean life is given by the accumulated life on test, $S_{n;r}$, divided by the number of observed failures.

#

However, when performing this estimation in practice, a common mistake is to ignore that for $i = r + 1, \dots, n$, one obtains $T_i > T_r$, which is the truncating time. They only use the data that record the lifetimes of the failed components. One type of mistake occurs when we ignore the observations $\{T_i > T_r, i = r + 1, \dots, n\}$ altogether. Then an incorrect MLE is obtained which is the arithmetic mean of $\{T_i, i = 1, \dots, r\}$:

$$\hat{\Theta}_{m1} = \frac{\sum_{i=1}^r T_i}{r}$$

Others use t_r as the observation for $\{T_i, i = r + 1, \dots, n\}$, which will lead to another incorrect result:

$$\hat{\Theta}_{m2} = \frac{(\sum_{i=1}^r T_i) + (n - r)T_r}{n}$$

It is easy to see that $\hat{\Theta} > \hat{\Theta}_{m2} > \hat{\Theta}_{m1}$, which shows that the incorrect estimates will be smaller than the correct estimate of MTTF.

Example 10.10 (Sampling from the Weibull Distribution)

Now we consider another sample truncated test of n components without replacement. The settings are the same as in the previous example, except that the lifetimes of these components follow a Weibull distribution.

As before, let $\{t_i, i = 1, \dots, n\}$ denote the observations. Actually, there are r observed failures. To simplify the expression, let $x_i = \min\{t_i, t_r\}$ for $i = 1, \dots, n$. Recall that Weibull density and distribution functions, respectively, are

$$f(t) = \lambda \alpha t^{\alpha-1} e^{-\lambda t^\alpha}$$

and

$$F(t) = 1 - e^{-\lambda t^\alpha}$$

The parameters λ and α are to be estimated. The likelihood function is then

$$\begin{aligned} L(\lambda, \alpha) &= \prod_{i=1}^r f(t_i | \lambda, \alpha) \prod_{i=r+1}^n R(t_r | \lambda, \alpha) \\ &= \prod_{i=1}^r \lambda \alpha t_i^{\alpha-1} e^{-\lambda t_i^\alpha} \prod_{i=r+1}^n e^{-\lambda t_r^\alpha} \\ &= \lambda^r \alpha^r (\prod_{i=1}^r x_i)^{\alpha-1} e^{-\lambda \sum_{i=1}^n x_i^\alpha}. \end{aligned}$$

Taking logarithms, we have

$$\ln L(\lambda, \alpha) = r \ln \lambda + r \ln \alpha + (\alpha - 1) \sum_{i=1}^r \ln x_i - \lambda \sum_{i=1}^n x_i^\alpha.$$

To maximize the log-likelihood, we take derivatives with respect to λ and α , respectively, and setting them equal to zero. We get

$$\frac{r}{\lambda} - \sum_{i=1}^n x_i^\alpha = 0$$

and

$$\frac{r}{\alpha} + \sum_{i=1}^r \ln x_i - \lambda \sum_{i=1}^n x_i^\alpha \ln x_i = 0.$$

There are no closed-form solutions for $\hat{\lambda}$ and $\hat{\alpha}$. However, the first equation can be solved for λ in terms of α as follows:

$$\lambda = \frac{r}{\sum_{i=1}^n x_i^\alpha}. \quad (10.6)$$

Plugging this into the second equation and applying some algebra, this equation reduces to

$$\frac{r}{\alpha} + \sum_{i=1}^r \ln x_i - \frac{r \sum_{i=1}^n x_i^\alpha \ln x_i}{\sum_{i=1}^n x_i^\alpha} = 0, \quad (10.7)$$

which must be solved iteratively. When the MLE $\hat{\alpha}$ is obtained, it is used to calculate $\hat{\lambda} = r / \sum_{i=1}^n x_i^{\hat{\alpha}}$. For more details, refer to Leemis [LEEM 1995].

#

Example 10.11 (Parameter Estimation from Interval Data)

When the values of lifetimes or survival times of components in a test or in service are available, such data are called *point data*. Test data and data from in-service experience of a component sometimes is available only in a coarse form because of the limitations on data collection such as cost. One such form of coarse data is called *interval data*, where the component lifetimes and survival times are known only to be within an interval or range. This type of data might arise naturally if the test items are inspected periodically and the failures recorded when found.

Strictly speaking, even point data are available in only interval form since the data points are recorded only to a few decimal places or recorded in digital format with a limited precision. For practical purposes the interval is generally small enough to consider such data to be point data.

Now consider developing an MLE estimate for a parameter from interval data. Assume a CDF with one parameter, $F(x|\lambda)$, for the lifetime of the component. Suppose that x_i ($i = 1, \dots, n$) are n independent samples for lifetimes from this distribution. We do not know the actual values for the x_i . Each x_i could fall in any of J_i intervals, whose endpoints $X_{i,0} = 0, X_{i,1}, \dots, X_{i,J_i-1}, X_{i,J_i} = \infty$, are independent of x_i and assumed to be known. The component is failed in interval j if $X_{i,j-1} < x_i \leq X_{i,j}$ and the component is considered to have never failed if $x_i > X_{i,J_i-1}$.

Suppose that x_i falls in the interval j_i . Then the likelihood function for these data is written as

$$L(\lambda) = \prod_{i=1}^n [F(X_{i,j_i}|\lambda) - F(X_{i,j_i-1}|\lambda)].$$

The log-likelihood function will be

$$\ln L(\lambda) = \sum_{i=1}^n \ln[F(X_{i,j_i}|\lambda) - F(X_{i,j_i-1}|\lambda)].$$

Taking the derivative with respect to λ and setting it to zero will yield a set of nonlinear equations for λ which can be solved for the MLE.

As an example suppose $F(x|\lambda) = 1 - e^{-\lambda x}$. Let $J_i = N + 1$ and $X_{i,j} = jh$ for all $i = 1, \dots, n$ and $j = 1, \dots, N$, where h is a constant inspection interval. Suppose that out of the n components tested, k of them fail in the first period $(0, h)$ and m never fail, that is, their lifetimes fall in (Nh, ∞) . Then the likelihood function is given by

$$L(\lambda) = (1 - e^{-\lambda h})^k \prod_{i=1}^{n-k-m} \{e^{-(\eta_{i-1})\lambda h} - e^{-\eta_i \lambda h}\} (e^{-\lambda Nh})^m$$

where η_i 's are the intervals during which the other $(n - k - m)$ components fail. Taking the logarithms, we get

$$\ln L(\lambda) = k \ln(1 - e^{-\lambda h}) + \sum_{i=1}^{n-k-m} \ln\{e^{-(\eta_{i-1})\lambda h} - e^{-\eta_i \lambda h}\} - \lambda m Nh.$$

Then, taking the derivative with respect to λ and setting it to zero will yield the following nonlinear equation for λ :

$$ke^{-\lambda h} + (n - k - m) = (1 - e^{-\lambda h})(mN + \sum_{i=1}^{n-k-m} \eta_i).$$

This equation can be solved so long as some components fail after the first inspection point. If all components fail before the first inspection ($k = n$) or if all survive past the last inspection ($m = n$), then the MLE does not exist. For the special case of $k = 0$ and $m = 0$, that is, where all components fail after the first inspection, then we have the following MLE for λ :

$$\lambda = -\frac{1}{h} \ln \left(1 - \frac{n}{\sum_{i=1}^n \eta_i} \right).$$

#

Example 10.12 (Parameter Estimation in Goel–Okumoto Model)

The finite-failure NHPP model is a class of software reliability models [OHBA 1984] that assume that software failures display the behavior of a nonhomogeneous Poisson process (NHPP). The parameter, $\lambda(t)$, of the stochastic process, is time-dependent. The function $\lambda(t)$ denotes the instantaneous failure intensity of the software at

time t . Let $N(t)$ denote the cumulative number of faults detected by time t , and let $m(t)$ denote its expectation $E[N(t)]$. Then $m(t)$ and the failure intensity $\lambda(t)$ are related as follows:

$$m(t) = \int_0^t \lambda(x) dx \quad (10.8)$$

$$\frac{dm(t)}{dt} = \lambda(t). \quad (10.9)$$

$N(t)$ is known to have a Poisson pmf with parameter $m(t)$:

$$P\{N(t) = n\} = \frac{[m(t)]^n e^{-m(t)}}{n!}, \quad n = 0, 1, 2, \dots, \infty.$$

Various NHPP models differ in their approach to determining $\lambda(t)$, and thus $m(t)$. Here we suppose $\lambda(t)$ and $m(t)$ are known. Let S_i denote the time of the occurrence of the i th failure, s_i denotes the observation of S_i . The probability density function of S_i at s_i , given the previous observations, is of the form

$$f(s_i | s_1, s_2, \dots, s_n, m(t), \lambda(t)) = \lambda(s_i) e^{-(m(s_i) - m(s_{i-1}))}.$$

Then the joint density or the likelihood function of S_1, S_2, \dots, S_n can be written as

$$f(s_1, s_2, \dots, s_n | m(t), \lambda(t)) = e^{-m(s_n)} \prod_{i=1}^n \lambda(s_i). \quad (10.10)$$

The NHPP models can be classified into finite-failure and infinite-failure categories. Finite-failure NHPP models assume that the expected number of faults detected given infinite amount of testing time will be finite, whereas the infinite failures models assume that an infinite number of faults would be detected in infinite testing time.

The Goel–Okumoto model [GOEL 1979] (see Example 8.14) is one of the most influential finite-failure NHPP models. It has a mean-value function $m(t)$ described by

$$m(t) = a(1 - e^{-bt}),$$

and a failure intensity function $\lambda(t)$, which is the derivative of $m(t)$, given as

$$\lambda(t) = abe^{-bt},$$

where a is the expected number of faults that would be detected given infinite testing time, and b is the failure occurrence rate per fault.

Now we have the observations $\{s_1, s_2, \dots, s_n\}$. In order to obtain the MLE of the parameters a and b in the Goel–Okumoto model, we consider the log-likelihood function, which is the natural logarithm of (10.10):

$$L(a, b) = \ln f(s_1, s_2, \dots, s_n | a, b) = n \ln a + n \ln b - a(1 - e^{-bs_n}) - b \sum_{i=1}^n s_i. \quad (10.11)$$

Maximizing equation (10.11) with respect to a and b , we have

$$\frac{n}{a} = 1 - e^{-bs_n} \quad (10.12)$$

and

$$\frac{n}{b} = as_n e^{-bs_n} + \sum_{i=1}^n s_i, \quad (10.13)$$

which are solved numerically to give the MLE for a and b . ‡

Problems

1. Show that the maximum-likelihood estimator of the mean life θ with a replacement test until r failures is

$$\hat{\Theta} = \frac{nT_r}{r},$$

where the random variable T_r denotes the time for the r th failure from the beginning of the experiment.

2. Suppose that the CPU service time X of a job is gamma-distributed with parameters λ and α . On the basis of a random sample of n observed service times, x_1, x_2, \dots, x_n , we wish to estimate parameters λ and α . Show that MLEs of λ and α do not yield a closed-form solution. Recall that method-of-moments estimators of λ and α are simple closed-form expressions.
3. Derive the MLE estimates of the parameters a , λ , and κ of the log-logistic software reliability growth model considered in Example 8.14 (in Chapter 8).
4. Show that the MLE estimates of the parameters of the Pareto distribution described in Chapter 3 (Section 3.4.8) satisfy the relation

$$\alpha = \left[\frac{1}{n} \sum_{i=1}^n \ln(x_i) - \ln(k) \right]^{-1}.$$

10.2.3 Confidence Intervals

The methods of parameter estimation discussed so far produce a **point estimate** of the desired parameter. Of course, the point estimate $\hat{\theta}$ rarely coincides with the actual value of the parameter θ being estimated. It is, therefore, desirable to find an **interval estimate**. We construct an interval, called a **confidence interval**, in such a way that we are reasonably confident that it contains the true value of the unknown parameter. If we can ascertain that the estimator $\hat{\Theta}$ satisfies the condition

$$P(\hat{\Theta} - \epsilon_1 < \theta < \hat{\Theta} + \epsilon_2) = \gamma,$$

then we say that the random interval $A(\theta) = (\hat{\Theta} - \epsilon_1, \hat{\Theta} + \epsilon_2)$ is a $100\gamma\%$ confidence interval for parameter θ . γ is called the **confidence coefficient**.

Often we choose a symmetric confidence interval so that $\epsilon_1 = \epsilon_2 = \epsilon$, as it gives the smallest interval width.

The meaning of this probability statement needs some clarification. For any specific set of observations, x_1, x_2, \dots, x_n , the estimate $\hat{\theta}$ is a fixed value. The confidence interval $A(\theta)$ either will or will not contain the true value of θ , in which case the probabilities are one and zero, respectively. However, when $\hat{\Theta}$ is considered as a function of random variables X_1, X_2, \dots, X_n , the endpoints of the interval $A(\theta)$ are then random variables, and it is possible to say that the probability is γ that the (random) interval $A(\theta)$ will contain the (fixed) true value of θ . The relative-frequency interpretation of the preceding probability implies that if this process of sampling is repeated many times, the fraction of the time in which the true value of θ is contained in the confidence interval $A(\theta)$ will be γ .

One simple way to obtain a confidence interval (involving an unbiased estimator) is to apply Chebyshev's inequality:

$$P(\hat{\Theta} - \epsilon < \theta < \hat{\Theta} + \epsilon) \geq 1 - \frac{\text{Var}[\hat{\Theta}]}{\epsilon^2},$$

provided $\text{Var}[\hat{\Theta}]$ is known (or can be estimated).

Example 10.13

Let $\theta = \mu$ and $\hat{\Theta} = \bar{X}$, the sample mean. Assume that the population variance σ^2 is known. Then $\text{Var}[\bar{X}] = \sigma^2/n$ and the Chebyshev inequality yields

$$P(\bar{X} - \epsilon < \mu < \bar{X} + \epsilon) \geq 1 - \frac{\sigma^2}{n\epsilon^2}.$$

Thus, for a given $\epsilon > 0$, we may make the confidence coefficient arbitrarily close to 1 by choosing a sufficiently large value of n .

#

Confidence intervals obtained by Chebyshev's inequality can usually be improved on if the distribution of X is known. In general, the steps involved in obtaining a confidence interval for the parameter θ from a random sample, X_1, X_2, \dots, X_n , are as follows:

1. Find a random variable that is a function of X_1, X_2, \dots, X_n :

$$W = W(X_1, X_2, \dots, X_n; \theta),$$

such that the distribution of W is known.

2. Find numbers a and b such that:

$$P(a < W < b) = \gamma.$$

3. After sampling the values x_i of X_i , determine the range of values that θ can take on while maintaining the condition $a < w(\theta) < b$, where $w(\theta) = W(x_1, x_2, \dots, x_n; \theta)$. This range of values is a $100\gamma\%$ confidence interval of θ .

It should be clear at the outset that step 1 depends on the distribution of X . Therefore, our subsequent discussion is divided according to some common distributions of X .

10.2.3.1 Sampling from the Normal Distribution. Suppose that a random sample of size n is taken from a normal population with unknown mean μ and a known variance σ^2 ; that is, $X \sim N(\mu, \sigma^2)$. Then it is easy to show that the sample mean is $N(\mu, \sigma^2/n)$, so that $Z = (\bar{X} - \mu)/(\sigma/\sqrt{n})$ is standard normal; that is, $Z \sim N(0, 1)$. Now, if we want a $100\gamma\%$ confidence interval for the population mean μ , we find numbers a and b [from $N(0, 1)$ tables] such that

$$P(a < Z < b) = \gamma.$$

Once the numbers a and b are determined, we obtain the required confidence interval as follows:

$$a < \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} < b$$

or

$$\bar{x} - \frac{b\sigma}{\sqrt{n}} < \mu < \bar{x} - \frac{a\sigma}{\sqrt{n}}.$$

Therefore, $(\bar{x} - b\sigma/\sqrt{n}, \bar{x} - a\sigma/\sqrt{n})$ is a $100\gamma\%$ confidence interval for μ .

Example 10.14

It is common to choose a symmetric confidence interval for μ so that we have $a = -b$. (If the estimator has a symmetric pdf, as it does in this case, then the choice $a = -b$ is known to produce the confidence interval of minimum width.) Then

$$P(-b < Z < b) = \gamma.$$

We let $\gamma = 1 - \alpha$ for convenience. Now, from the symmetry of the pdf of Z , we obtain

$$P(Z < -b) = \frac{\alpha}{2} \quad \text{and} \quad P(Z > b) = \frac{\alpha}{2}.$$

This value of b is usually denoted by $z_{\alpha/2}$ (see Figure 10.1), and these values can be read from a table. Now

$$\left(\bar{x} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}}, \bar{x} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right)$$

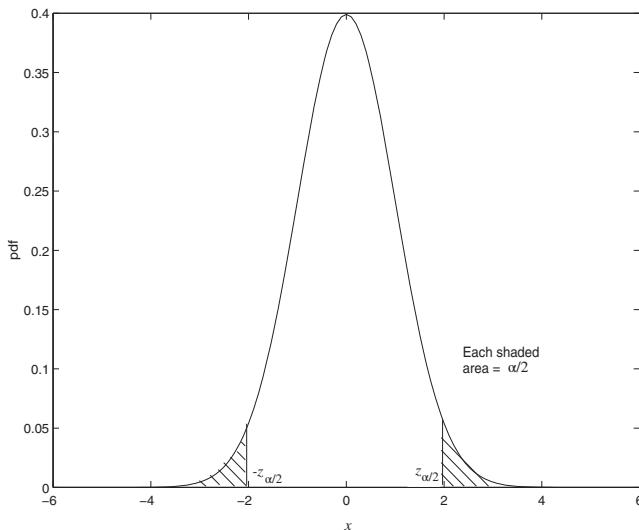


Figure 10.1. $P(Z > z_{\alpha/2}) = \alpha/2 = P(Z < -z_{\alpha/2})$

TABLE 10.1. Critical Values of $N(0,1)$

$1 - \alpha$	0.90	0.95	0.99
$z_{\alpha/2}$	1.645	1.96	2.576

is a $100(1 - \alpha)\%$ confidence interval for the population mean μ . The usual values of $(1 - \alpha)$ and corresponding $z_{\alpha/2}$ are shown in Table 10.1.

We have $100(1 - \alpha)\%$ confidence that the sample mean \bar{X} deviates from the population mean μ by less than $E = z_{\alpha/2}\sigma/\sqrt{n}$. Then, the sample size required in order to produce a symmetric $100(1 - \alpha)\%$ confidence interval of width $2E$ for the population mean is given by

$$n = \left\lceil \left(\frac{z_{\alpha/2}\sigma}{E} \right)^2 \right\rceil.$$

‡

Example 10.15

The average working-set size X of a program is normally distributed with unknown mean μ and a known variance $\sigma^2 = 81$. The program was executed 36 times and the average working-set size for each run recorded. The sample mean was computed to be 100 page frames. Assuming that successive runs of the program are independent, the 95% confidence interval for the mean average working-set size is given by

$$(100 - 1.96 \cdot 9/6, 100 + 1.96 \cdot 9/6) = (97.06, 102.94).$$

‡

Example 10.16

Suppose that we wish to estimate the average CPU service time of a job and we wish to assert with a 99% confidence that the estimated value is within less than 0.5 s of the true value. Suppose that past experience suggests that CPU service time is normally distributed with $\sigma^2 = 2.25$ s². Then the required number of random samples is given by

$$n = \left\lceil \left(\frac{2.576 \cdot 1.5}{0.5} \right)^2 \right\rceil = \lceil 59.722 \rceil = 60.$$

‡

Two difficulties with the interval estimation procedure discussed so far should be noted. First, the assumption that the population is normally distributed does not always hold. In the next few sections we will discuss interval estimation when the population is not normally distributed. Also note that, in practice, the assumption of normality does not pose a problem when the sample size is large, owing to the central-limit theorem, which states that the statistic $(\bar{X} - \mu)/(\sigma/\sqrt{n})$ is asymptotically normal (under appropriate conditions).

The second difficulty with the formula for confidence interval given above is that it requires the knowledge of population variance σ^2 . If σ^2 is unknown, we may replace it by its estimate s^2 to get an approximate confidence interval for μ :

$$\bar{x} - \frac{z_{\alpha/2}s}{\sqrt{n}} < \mu < \bar{x} + \frac{z_{\alpha/2}s}{\sqrt{n}},$$

which will be a good approximation for large values of n ($n > 30$). When the sample size is relatively small, this approximation is poor. But in this case we can make use of Student t distribution.

If \bar{X} is the sample mean of a random sample of size n from a normal population having the mean μ and variance σ^2 , then by Example 3.39 we have that the random variable

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}}$$

has Student t distribution with $n - 1$ degrees of freedom.

In Figure 10.2, the pdf of t distribution with 3 degrees of freedom is plotted, together with the standard normal density for comparison.

Thus we obtain the $100(1 - \alpha)\%$ confidence interval of μ as

$$\bar{x} - t_{n-1;\alpha/2} \frac{s}{\sqrt{n}} < \mu < \bar{x} + t_{n-1;\alpha/2} \frac{s}{\sqrt{n}}, \quad (10.14)$$

where $t_{n-1;\alpha/2}$ is defined such that the area under the t pdf to its right is equal to $\alpha/2$ or $P(T > t_{n-1;\alpha/2}) = \alpha/2$ (Figure 10.3). This value can be read from a table (see Appendix C).

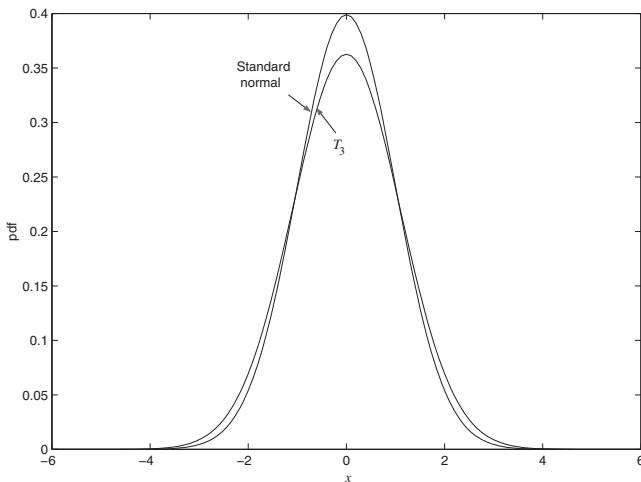


Figure 10.2. Comparing the standard normal pdf with the pdf of the t distribution with 3 degrees of freedom

Example 10.17

We wish to estimate the average execution time of a program. The program was run six times with randomly chosen data sets, and the sample mean of the execution times was evaluated as $\bar{x} = 230$ ms and the sample standard deviation as $s = 14$ ms. To obtain a 98% confidence interval of the true mean execution time μ , we read $t_{5;0.01}$ from the table of t distribution with $n - 1 = 5$ degrees of freedom to be 3.365. Then the required confidence interval is

$$230 - \frac{3.365 \cdot 14}{\sqrt{6}} < \mu < 230 + \frac{3.365 \cdot 14}{\sqrt{6}}$$

or

$$210.767 < \mu < 249.233 \quad (\text{with 98\% confidence}).$$

‡

So far, we have considered confidence intervals for the population mean. Next we discuss confidence intervals for the population variance. If X is normally distributed, then we have shown in Example 3.37 that the random variable

$$X_{n-1}^2 = \frac{(n-1)S^2}{\sigma^2}$$

possesses a chi-square distribution with $n - 1$ degrees of freedom.

To determine a $100(1 - \alpha)\%$ confidence interval of σ^2 , we find two numbers a and b such that

$$P \left[a < \frac{(n-1)S^2}{\sigma^2} < b \right] = 1 - \alpha.$$

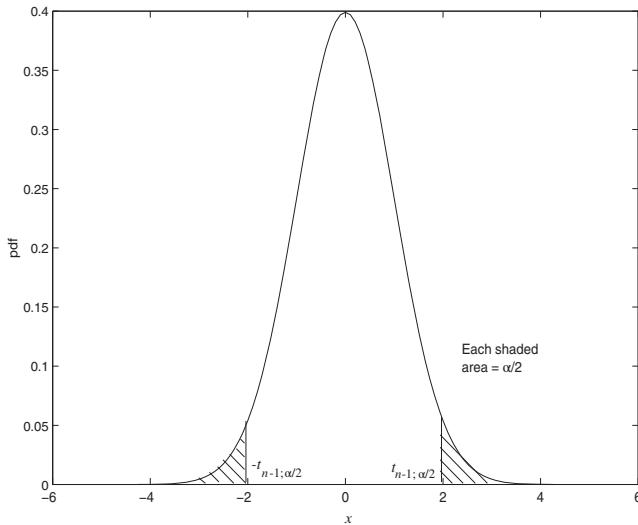


Figure 10.3. $P(T_{n-1} > t_{n-1;\alpha/2}) = P(T_{n-1} < -t_{n-1;\alpha/2})$

Since chi-square is a nonnegative random variable, its density is obviously not symmetric about zero. Therefore, in choosing a and b , the requirement of “equal tails” is usually imposed (see Figure 10.4), so that

$$P(X_{n-1}^2 > b) = \frac{\alpha}{2} \quad \text{and} \quad P(X_{n-1}^2 < a) = \frac{\alpha}{2}.$$

In this case, b and a are denoted by $\chi_{n-1;\alpha/2}^2$ and $\chi_{n-1;1-\alpha/2}^2$, respectively. The $100(1 - \alpha)\%$ confidence interval of the population variance is then given by

$$\frac{(n-1)s^2}{\chi_{n-1;\alpha/2}^2} < \sigma^2 < \frac{(n-1)s^2}{\chi_{n-1;1-\alpha/2}^2}.$$

Note that, like the confidence interval for the population mean μ found using the t distribution, this interval does not require knowledge of any parameters of the population distribution function.

Example 10.18

A usual complaint of file server users is the large variance of the response time. While contemplating the purchase of a new file server, we measure 30 random samples of response times, and compute the sample variance to be 25 ms^2 . Assuming the response times are approximately normally distributed, a 95% confidence interval for the population variance σ^2 is given by

$$\frac{(n-1)s^2}{\chi_{n-1;\alpha/2}^2} < \sigma^2 < \frac{(n-1)s^2}{\chi_{n-1;1-\alpha/2}^2}, \quad \text{where } \alpha = 0.05.$$

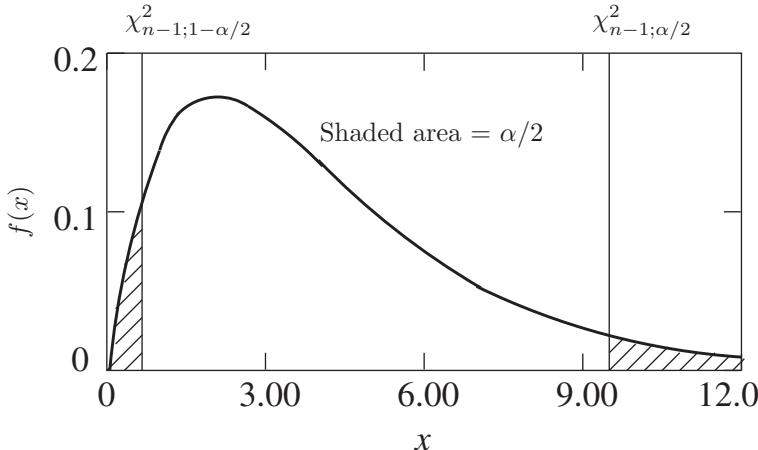


Figure 10.4. $P(X_{n-1}^2 > \chi_{n-1;\alpha/2}^2) = P(X_{n-1}^2 < \chi_{n-1;1-\alpha/2}^2)$

From a table of chi-square distribution (see Appendix C) with $n - 1 = 29$ degrees of freedom, $\chi_{29;0.025}^2 = 45.722$ and $\chi_{29;0.975}^2 = 16.047$, so that the required confidence interval is obtained from the relation

$$\frac{29.25}{45.722} < \sigma^2 < \frac{29.25}{16.047}$$

as (15.86, 45.18). ‡

The construction of confidence intervals discussed so far was based on the assumption that X is normally distributed. Empirical evidence suggests that the confidence interval for μ based on the normality of the statistic $(\bar{X} - \mu)\sqrt{n}/\sigma$ is highly reliable (in the sense of providing adequate coverage) even when the distribution of X is considerably different from normal. However, the confidence interval of σ^2 derived above can be quite poor when X has a distribution significantly different from normal.

Problems

1. In an exhaustive, nonreplacement life test of 10 components, the observed times to failure (in hours) are: 1200, 1500, 1625, 1725, 1750, 1785, 1800, 1865, 1900, and 1950. Assuming that component lifetimes are normally distributed, compute an estimate of the mean life μ and the variance σ^2 . Also compute a 90% confidence interval for the mean life.
2. Given the following 20 measurements of the mean length of a CPU queue, compute the best estimates of the mean queue length, variance of the queue length, and a 95% confidence interval of the mean queue length; assume that

queue-length distribution is approximately normal.

3.00	2.87	3.58	3.28	3.87
4.14	5.23	3.86	2.88	4.37
4.75	4.33	3.17	2.85	4.16
4.03	3.57	3.68	3.95	3.58

3. A program was tested using a random collection of 30 input data sets, and execution time was measured for each run. The sample mean and the sample variance of the execution time were found to be $\bar{x} = 65$ ms and $s^2 = 36$ ms², respectively. Derive a 95% confidence interval for the average execution time of the program assuming that the population is normal.
4. Execution times (in seconds) of 40 jobs processed by a server were measured and found to be

10	19	90	40	15	11	32	17	4	152
23	13	36	101	2	14	2	23	34	15
27	1	57	17	3	30	50	4	62	48
9	11	20	13	38	54	46	12	5	26

Calculate the sample mean and the sample variance. Find the 90% confidence intervals for the mean and the variance of execution time of a job. Assume that the execution time is approximately normally distributed.

10.2.3.2 Sampling from the Exponential Distribution. Now we consider the special case when X is exponentially distributed. This case is of importance in queuing theory as well as in reliability theory (life testing).

We have noted that under exhaustive testing of n components, the maximum-likelihood estimator of mean life is given by the sample mean:

$$\hat{\Theta} = \bar{X}.$$

Now if X is exponentially distributed with parameter λ , then $\sum_{i=1}^n X_i$ is an n -stage Erlang random variable with parameter λ , and $\bar{X} = (\sum_{i=1}^n X_i)/n$ is n -stage Erlang with parameter $n\lambda$. This implies that

$$2n\lambda\bar{X} = \frac{2n\bar{X}}{\theta} = \frac{2n\hat{\Theta}}{\theta}$$

has an n -stage Erlang distribution with parameter $\frac{1}{2}$, which is the chi-square distribution with $2n$ degrees of freedom. Then a $100(1 - \alpha)\%$ confidence interval of mean life θ is obtained from

$$\chi^2_{2n;1-\alpha/2} < \frac{2n\hat{\theta}}{\theta} < \chi^2_{2n;\alpha/2}$$

as

$$\hat{\theta} \frac{2n}{\chi^2_{2n;\alpha/2}} < \theta < \hat{\theta} \frac{2n}{\chi^2_{2n;1-\alpha/2}}.$$

Next we consider a test (without replacement) terminated after $r \leq n$ failures have occurred. Consider the accumulated life on test, $S_{n;r}$:

$$S_{n;r} = \left(\sum_{i=1}^r T_i \right) + (n - r)T_r.$$

Let Y_i denote the time from the $(i - 1)$ st failure to the i th failure so that

$$T_i = \sum_{j=1}^i Y_j, \quad i = 1, 2, \dots, r.$$

Then

$$\begin{aligned} S_{n;r} &= \sum_{i=1}^r T_i + (n - r)T_r \\ &= \sum_{i=1}^r \sum_{j=1}^i Y_j + (n - r) \sum_{j=1}^r Y_j \\ &= (Y_1) + (Y_1 + Y_2) + (Y_1 + Y_2 + Y_3) + \cdots + (Y_1 + Y_2 + \cdots + Y_r) \\ &\quad + (n - r)(Y_1 + Y_2 + \cdots + Y_r) \\ &= \sum_{i=1}^r (r - i + 1)Y_i + \sum_{i=1}^r (n - r)Y_i \\ &= \sum_{i=1}^r (n - i + 1)Y_i. \end{aligned}$$

Now, from our discussion in Example 3.26, Y_i is exponentially distributed with parameter $(n - i + 1)\lambda$, and therefore $(n - i + 1)Y_i$ is exponentially distributed with parameter λ . Therefore $S_{n;r}$ is r -stage Erlang with parameter λ , and hence $2\lambda S_{n;r} = 2S_{n;r}/\theta$ is r -stage Erlang with parameter $\frac{1}{2}$,—that is, the X_{2r}^2 distribution. Thus a $100(1 - \alpha)\%$ confidence interval for θ is given by

$$\frac{2s_{n;r}}{\chi_{2r;\alpha/2}^2} < \theta < \frac{2s_{n;r}}{\chi_{2r;1-\alpha/2}^2}.$$

Example 10.19

Assume that $n = 50$ chips are placed on a life test without replacement and the test is to be truncated after $r = 10$ failures have been observed. Observed failure times are $t_1 = 80, t_2 = 95, t_3 = 370, t_4 = 415, t_5 = 505, t_6 = 590, t_7 = 635, t_8 = 835, t_9 = 895$, and $t_{10} = 960$ h. Then

$$\begin{aligned} s_{n;r} &= (80 + 95 + 370 + 415 + 505 + 590 + 635 + 835 + 895 + 960) \\ &\quad + (50 - 10)960 = 43,780 \text{ h.} \end{aligned}$$

The estimated mean life is $\hat{\theta} = 43780/10 = 4378$ h, and the estimated failure rate is $\hat{\lambda} = 0.0002284$ failures per hour. Finally, a 90% confidence interval for mean life is

$$\frac{2(43,780)}{31.410} < \theta < \frac{2(43,780)}{10.851}$$

or

$$2787 < \theta < 8069 \text{ h},$$

where $\chi^2_{20;0.05} = 31.410$ and $\chi^2_{20;0.95} = 10.851$ values are obtained from a table of chi-square distribution (see Appendix C) with 20 degrees of freedom.

#

Recalling that the interevent times of a Poisson process are exponentially distributed, we can obtain a confidence interval for the average arrival rate. Assume that a Poisson process of rate λ is observed until a fixed number n of events have been counted. Let X_i denote the time between the $(i-1)$ st and i th event. Then X_i is exponentially distributed with parameter λ , and the statistic

$$S_n = \sum_{i=1}^n X_i$$

is n -stage Erlang with parameter λ . It follows that $2\lambda S_n$ is chi-square distributed with $2n$ degrees of freedom. Consequently

$$\left(\frac{\chi^2_{2n;1-\alpha/2}}{2s_n}, \frac{\chi^2_{2n;\alpha/2}}{2s_n} \right)$$

is a confidence interval for λ , with confidence coefficient $(1 - \alpha)$.

Example 10.20

Arrival of jobs to a file server was monitored, and it was found that 50 jobs arrived within 100 min. Assuming a Poisson model, the maximum-likelihood estimate for the average job arrival rate is $\hat{\lambda} = 50/100 = 0.5$ jobs per minute. Noting that $\chi^2_{100;0.05} = 124.34$ and $\chi^2_{100;0.95} = 77.93$, we find the 90% confidence interval for λ to be $(0.39, 0.62)$.

#

Sometimes a one-sided confidence interval is sought in place of the two-sided interval given above. For example, an **upper one-sided** confidence interval of the mean life θ is denoted by (Θ_L, ∞) where Θ_L is known as the **lower confidence limit**. Since $2S_{n;r}/\theta$ is chi-square distributed with $2r$ degrees of freedom, we have

$$P\left(\frac{2S_{n;r}}{\theta} < \chi^2_{2r;\alpha}\right) = 1 - \alpha.$$

It follows that

$$\frac{2s_{n;r}}{\theta_L} = \chi^2_{2r;\alpha} \quad \text{or} \quad \theta_L = \frac{2s_{n;r}}{\chi^2_{2r;\alpha}}.$$

Similarly, a **lower one-sided** confidence interval of the mean life is denoted by $(0, \Theta_U)$, where a value of the **upper confidence limit** Θ_U is given by

$$\theta_U = \frac{2s_{n;r}}{\chi^2_{2r;1-\alpha}}.$$

Example 10.21 (Continued from Example 10.19)

Returning to Example 10.19, we note that for a chi-square distribution with $2r = 20$ degrees of freedom, $\chi^2_{2r;\alpha} = \chi^2_{20;0.1} = 28.41$ and $\chi^2_{2r;1-\alpha} = \chi^2_{20;0.9} = 12.443$. It follows that the 90% lower confidence limit of the mean life is given by

$$\theta_L = \frac{2s_{n;r}}{\chi^2_{20;0.1}} = \frac{87560}{28.41} = 3082 \text{ h.}$$

Therefore, with 90% confidence we can assert that the true mean life is greater than 3082 h. The 90% upper confidence limit is

$$\theta_U = \frac{2s_{n;r}}{\chi^2_{20;0.9}} = \frac{87560}{12.443} = 7036 \text{ h.}$$

Therefore, with 90% confidence we can assert that the true mean life is less than 7036 h. ‡

For ultra-high-reliability systems, the mean life may be much larger than the duration of a normal “mission.” In this case we are more interested in obtaining a confidence interval for system reliability given a mission time t . We proceed to derive such a confidence interval starting from the $100(1 - \alpha)\%$ upper one-sided confidence interval of the mean life θ . Thus

$$\begin{aligned} 1 - \alpha &= P(\theta \geq \Theta_L) \\ &= P(e^{-t/\theta} \geq e^{-t/\Theta_L}) \end{aligned}$$

(since the exponential is a monotonic function)

$$= P\left[R(t) \geq e^{-t/\Theta_L}\right].$$

In other words

$$R_L = e^{-t/\theta_L} = e^{\frac{-(t\chi^2_{2r;\alpha})}{2s_{n;r}}}$$

is the lower $100(1 - \alpha)\%$ confidence limit for the reliability, given a mission time t . Note that the chi-square distribution here has $2r$ degrees of freedom, since we are discussing a test, without replacement, truncated after r failures.

In the running example of this section, we have 90% confidence that the chip reliability exceeds the threshold $R_L = e^{-t/3082}$. Thus, if we observe a large number of chips for 30.82 h, we are 90% confident that at least $100 \cdot e^{-0.01} = 99\%$ of the chips will still be functioning properly. Similarly, if we observe a large number of chips for 3082 h, we are 90% confident that at least $100 \cdot e^{-1.0} = 36.79\%$ of chips to be functioning properly.

Problems

1. Assume that 15 RAM chips are put into operation, and a truncated non-replacement life test is conducted until three chips have failed. Corresponding failure times are noted as $t_1 = 850$ h, $t_2 = 900$ h, and $t_3 = 1000$ h. Assume that the devices follow an exponential failure law.
 - (a) Obtain a point estimate of the mean life.
 - (b) Obtain a 90% confidence interval for the mean life of a chip.
2. Show that the $100(1 - \alpha)\%$ confidence interval for the mission time t_γ such that the reliability for this mission time satisfies

$$R(t_\gamma) = P(X > t_\gamma) = \gamma,$$

is given by

$$\left[\frac{2s_{n;r}}{\chi^2_{2r;\alpha/2}} \ln\left(\frac{1}{\gamma}\right), \frac{2s_{n;r}}{\chi^2_{2r;1-\alpha/2}} \ln\left(\frac{1}{\gamma}\right) \right].$$

It is assumed that the lifetime X is exponentially distributed with the parameter λ (which is to be estimated from the data), and the remaining assumptions are the same as those made throughout the section.

3. Assume that 20 items are placed on a life test. The first and the second failures occur at 3001 and 7030 h, respectively, after which time the test is terminated. Assuming that the lifetimes of items are exponentially distributed, compute
 - (a) The maximum-likelihood estimate of the MTTF
 - (b) The 95% confidence interval for the MTTF
 - (c) The 95% confidence interval for the length of the mission with reliability 0.9

10.2.3.3 Sampling from the Weibull Distribution. In the case that the exponential distribution is not good enough to fit the data, the Weibull distribution can yield better results. Typically, the Weibull distribution is more appropriate for modeling the lifetimes with increasing and decreasing failure rates. In fact, the exponential distribution which models the lifetimes having a constant failure rate is only a special case of the Weibull distribution with the shape parameter $\alpha = 1$.

We have already derived the likelihood function and the maximum likelihood estimator in Example 10.10. Now we are trying to find an interval estimator. Since there are two parameters in the model, the estimator should be a confidence region rather than a confidence interval. In Example 10.10, we do not have a closed-form solution for the maximum-likelihood estimator. Estimating the confidence region is even harder. Leemis [LEEM 1995] found the likelihood ratio statistic, $2[\ln L(\hat{\lambda}, \hat{\alpha}) - \ln L(\lambda, \alpha)]$, to be asymptotically chi-square distributed with 2 degrees of freedom. Therefore, an asymptotical 95% confidence region for the parameters is all λ and α satisfying

$$2[\ln L(\hat{\lambda}, \hat{\alpha}) - \ln L(\lambda, \alpha)] < \chi^2_{2;0.05}.$$

Example 10.22

Consider the following data that could represent failure times of identical copies of a component:

134.21	304.27	416.82	450.61	525.96
553.94	627.11	817.53	870.2	1034.26
1057.86	1065.73	1070.47	1170.9	1254.87
1283.32	1697.44	1891.23	1939.45	2077.96

We would like to analyze these data to see if the component's failure rate increases with the age of the component. If we assume that a Weibull CDF will fit these data, we can obtain the MLE for the two parameters α and λ , using the equations (10.6) derived earlier in Example 10.10. Solving the MLE equations we obtain $\hat{\lambda} = 0.00000131$ and $\hat{\alpha} = 1.924$. The confidence region for a particular value of confidence, say, 90%, is quite complex, and we can obtain the bounding box of this region from the approximate chi-square distribution of the log-likelihood ratio. For these data we obtain the bounding box of the 90% confidence region as the cross-product of the interval (1.394, 2.544) for α and (0.00000088, 0.00000198) for λ . Since the 90% confidence region does not include $\alpha \leq 1.0$, it is seen that the component has an increasing failure rate at the 90% confidence interval.

The data given above are actually synthetic and were generated from a Weibull CDF with $\lambda = 0.000001$ and $\alpha = 2.00$ using a random-number generator. The MLE values of $\hat{\lambda} = 0.00000131$ and $\hat{\alpha} = 1.924$ are close to the true values. The 90% confidence region bounding box encloses the true values in this case, although this need not be the case in general. The Figure 10.5 shows the comparison between the empirical CDF, the MLE estimate, and the CDF from which the data were generated.

#

10.2.3.4 Sampling from the Bernoulli Distribution. In many situations we are interested in the percentage of certain components that will perform satisfactorily for a given period. At other times we may be interested in the proportion of client requests whose response times do not exceed a

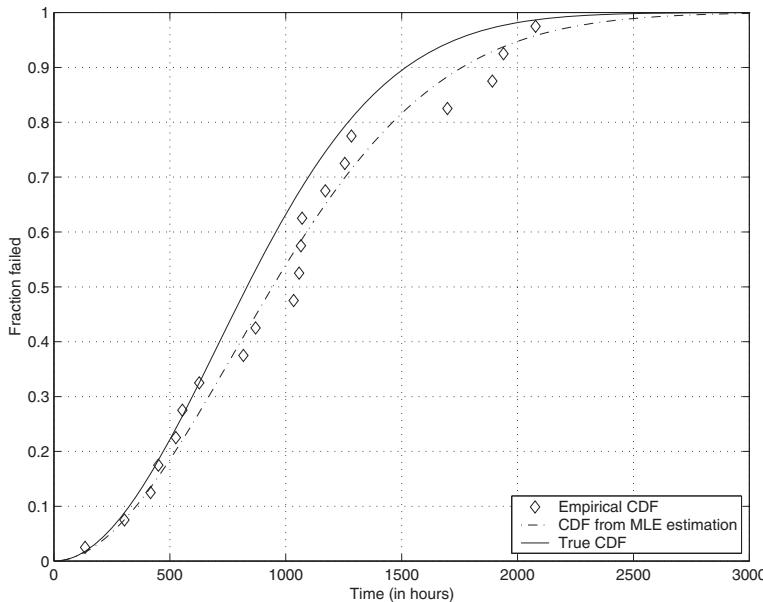


Figure 10.5. Weibull estimation

threshold. Each individual experimental observation, X_i , can then be treated as a Bernoulli random variable with an unknown parameter p . The statistic

$$S_n = \sum_{i=1}^n X_i$$

is then binomially distributed:

$$F_{S_n}(k) = B(k; n, p).$$

As we have seen earlier, the maximum-likelihood estimator of the proportion of successes is given by

$$\hat{P} = \frac{S_n}{n} = \bar{X}.$$

Since $E[S_n] = np$, we have $E[\hat{P}] = p$, and $\text{Var}[S_n] = np(1 - p)$ implies that $\text{Var}[\hat{P}] = p(1 - p)/n$. Thus, the sample proportion, \hat{P} , is a consistent and unbiased estimator of the “true” proportion, p .

For a statistic such as S_n that has a discrete distribution, it is possible to use a software package such as Mathematica, as we shall soon see. In absence of an access to such a tool, we derive a confidence interval for p with an approximate degree of confidence $(1 - \alpha)$.

Let k_0 be the largest integer such that

$$\sum_{k=0}^{k_0} b(k; n, p) = B(k_0; n, p) \leq \frac{\alpha}{2},$$

and let k_1 be the smallest integer such that

$$\sum_{k=k_1}^n b(k; n, p) = 1 - B(k_1 - 1; n, p) \leq \frac{\alpha}{2}.$$

Note that k_0 and k_1 are functions of p . Then, since $P[k_0(p) < S_n < k_1(p)] \simeq 1 - \alpha$, an approximate $100(1 - \alpha)\%$ confidence interval for p can be obtained by inverting

$$k_0(p) < s_n < k_1(p).$$

Unfortunately, there are no closed-form expressions for k_0 and k_1 as functions of p . Therefore, a tabular technique is usually employed to obtain the desired confidence interval.

Example 10.23

From a large population of RAM chips, a sample of 20 is taken and a test carried out on each to see whether they perform correctly. In the test 7 chips are found to perform correctly and the remaining 13 do not perform to specifications. Therefore a point estimate of the yield of these chips is $\hat{p} = \frac{7}{20} = 0.35$.

In order to determine a 95% confidence interval for p , we have to determine integers k_0 and k_1 such that k_0 is the largest integer satisfying

$$B(k_0; 20, p) \leq 0.025,$$

while k_1 is the smallest integer satisfying

$$1 - B(k_1 - 1; 20, p) \leq 0.025.$$

Using the binomial formula, and varying p from 0.1 to 0.9, we obtain the following table of values:

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
k_0	—	0	1	3	5	7	9	11	14
k_1	6	9	11	13	15	17	19	20	—

Now, since $s_n = k = 7$, the interval of p satisfying

$$k_0(p) < 7 < k_1(p)$$

is $(0.133, 0.600)$ from the table above (together with a little interpolation). This is the 95% confidence interval that was sought for p . ‡

Another approach is to use a software package such as Mathematica [WOLF 1999] to calculate the lower and upper bounds of p . The result will be the exact solution (subject to numerical computation errors). The following Mathematica code will solve the problem in Example 10.23:

```

n = 20
y = 7
alpha = 0.05
proot = FindRoot[ Sum[Binomial[n,k] * p^k * (1-p)^(n-k),
{k,y,n}] == alpha/2, {p, y/n}]
pl = p /. proot[[1]];
proot = FindRoot[ Sum[Binomial[n,k] * p^k * (1-p)^(n-k),
{k,0,y}] == alpha/2 ,{p, y/n}]
pu = p /. proot[[1]];
Print[pl, " ", pu]
(* <<Statistics`ContinuousDistributions` *)

```

The execution of this code yields the interval (0.154, 0.592).

If the sample size n is large or p is not suspected to be very close to either 0 or 1, we can use the normal approximation to the binomial distribution. As a rule of thumb, $np \geq 5$ and $nq \geq 5$ usually suffice. Thus S_n is approximately normal with $\mu = np$ and $\sigma^2 = np(1 - p)$. Note that σ^2 contains the unknown parameter p , but we may approximate it by $\hat{\sigma}^2 = n\hat{p}(1 - \hat{p})$. Then

$$\frac{S_n - np}{\hat{\sigma}}$$

is approximately standard normal. It follows that an approximate $100(1 - \alpha)\%$ confidence interval for p is obtained from

$$-z_{\alpha/2} < \frac{s_n - np}{\hat{\sigma}} < z_{\alpha/2}$$

or from

$$\frac{s_n}{n} - z_{\alpha/2} \sqrt{\frac{s_n(1 - \frac{s_n}{n})}{n^2}} < p < \frac{s_n}{n} + z_{\alpha/2} \sqrt{\frac{s_n(1 - \frac{s_n}{n})}{n^2}}.$$

For instance, using this approximation for the data of Example 10.23, and noting that $z_{0.025} = 1.96$, we get a 95% confidence interval of p to be

$$0.35 - 1.96 \sqrt{\frac{7(1 - 0.35)}{400}} < p < 0.35 + 1.96 \sqrt{\frac{7(1 - 0.35)}{400}}.$$

Thus the required interval is (0.141, 0.559), which, considering the assumptions, is in fair agreement with the results obtained in Example 10.23.

In many practical situations of interest, the Bernoulli parameter p is very close to either 0 or 1. For example, while estimating the fault coverage of a fault-tolerant computer system, we would expect the probability of successful recovery to be close to 1—say, > 0.9 . On the other hand, in a quality-control inspection plan \hat{p} will represent a fraction of the total inspected items found to be defective. In this case we will expect the population parameter to be close to 0—say, < 0.1 . In such cases the normal approximation to the binomial distribution will be poor. However, for $p < 0.1$ we may use the Poisson approximation to the binomial, provided that the sample size is large enough. In the complementary case of $p > 0.9$, we obtain a confidence interval for $q = 1 - p$ using the same approach.

In the case $p < 0.1$, we will be interested in a one-sided confidence interval with an approximate confidence coefficient γ so that if k is the observed number of successes, we write

$$\begin{aligned}\gamma &\leq P(S_n \leq k) \\ &= \sum_{i=0}^k \binom{n}{i} p^i (1-p)^{n-i} \\ &\simeq \sum_{i=0}^k e^{-np} \frac{(np)^i}{i!}.\end{aligned}$$

Now, by comparison with the distribution function of the $(k+1)$ -stage Erlang random variable with parameter np , denoted by Y , we see that the right-hand side is equal to $1 - F_Y(1)$. But then $2npY$ is chi-square distributed with $2(k+1)$ degrees of freedom; hence we have

$$P(X_{2(k+1)}^2 < 2np) < 1 - \gamma.$$

It follows that the required confidence interval for p is given by

$$p < \frac{1}{2n} \chi_{2(k+1);\gamma}^2. \quad (10.15)$$

For example, if a sample of 50 RAM chips selected at random from a large batch is found to contain nine defectives, an approximate 90% confidence interval for the fraction defective in the entire batch is given by

$$p < \frac{1}{2 \cdot 50} \chi_{20;0.1}^2 = 0.284.$$

The point estimate of p in this case is given by

$$\hat{p} = \frac{9}{50} = 0.18.$$

The errors in the confidence intervals due to the approximation are studied in Leemis and Trivedi [LEEM 1996]. The rules of thumb for normal approximation and Poisson approximation are compared when these errors are considered. Charts are given there to indicate which approximation is appropriate for certain sample sizes and point estimators. Some of the recommendations are as follows:

- The Poisson approximation should be used when $n \geq 20$ and $p \leq 0.05$ at $\alpha = 0.05$ if the analyst can tolerate an absolute error in either limit that may be as large as 0.04.
- For sample sizes larger than 150, the maximum absolute error of the upper and lower confidence limits is less than 0.01 if the appropriate approximation technique is used.

Example 10.24 (Deriving Confidence Intervals of the Error Detection Coverage Probability)

Physical and simulated fault injection experiments are performed on a prototype server to derive error detection coverage probability [CONS 1999]. Because of the randomness of the fault injection, statistical inference has to be employed for processing the experimental results. Sampling in partitioned and nonpartitioned spaces and stratified and two-stage sampling are the main techniques used by Powell et al. [POWE 1995]. Multistage, stratified, and combined sampling techniques have also been employed [CONS 1995] for estimating coverage probabilities.

In this analysis we are particularly interested in deriving independent coverage probabilities and their confidence intervals for each duration of the injected transient faults. Student t and chi-square probability distributions are used for this purpose. We use the following notations: n as number of injected faults, k as number of detected errors, and c as error-detection coverage probability. Let us represent the outcome of the i th fault injection experiment by a Bernoulli random variable, X_i . Its observed value is

$$x_i = \begin{cases} 1 & \text{detected error,} \\ 0 & \text{undetected error.} \end{cases}$$

The statistic $S_n = \sum_{i=1}^n X_i$ is binomially distributed, with the distribution function $F_{S_n}(k) = B(k; n, c)$. An unbiased estimate of the coverage probability is $\hat{c} = \sum_{i=1}^n x_i/n = \bar{x}$.

Confidence intervals of the error-detection coverage probability can be obtained by using the binomial formula or tables of the binomial distribution. The normal distribution provides a good approximation to the binomial distribution as long as the sample size is large enough and c is not close to either 0 or 1. For smaller sample sizes, Student t distribution gives a more accurate estimation of the confidence intervals. From equation (10.14), the $100\gamma\%$ confidence interval of the error detection coverage, obtained with the aid of Student t distribution with $n - 1$ degrees of freedom, is

$$\bar{x} - t_{n-1;\alpha/2} \frac{s}{\sqrt{n}} < c < \bar{x} + t_{n-1;\alpha/2} \frac{s}{\sqrt{n}}, \quad (10.16)$$

where

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}.$$

When c is expected to take low ($c < 0.1$) or high ($c > 0.9$) values, it is desirable to derive one-sided confidence intervals, with the aid of Poisson approximation and the χ^2 distribution with $2(k+1)$ degrees of freedom. From equation (10.15), we have the following for $c < 0.1$:

$$c < \frac{1}{2n} \chi^2_{2(k+1);\gamma}. \quad (10.17)$$

In the case of $c > 0.9$ the one-sided confidence interval is derived for $q = 1 - c$, using a similar approach.

Because of schedule requirements the sample size used in our experiments is limited to 30. Confidence intervals are derived with the aid of equation (10.16), when $0.1 \leq c \leq 0.9$ (Student t distribution). Equation (10.17) is used for $c < 0.1$ and $c > 0.9$ (χ^2 distribution).

Confidence intervals of the error detection coverage are given in Figure 10.6. A 90% confidence level is considered. Physical fault injection shows that error-detection coverage is close to zero for transient faults in the 25 ns–8 μ s range (curves labeled I,LCL and I,UCL plots). Higher coverage is observed as fault duration increases from 8 to 120 μ s.

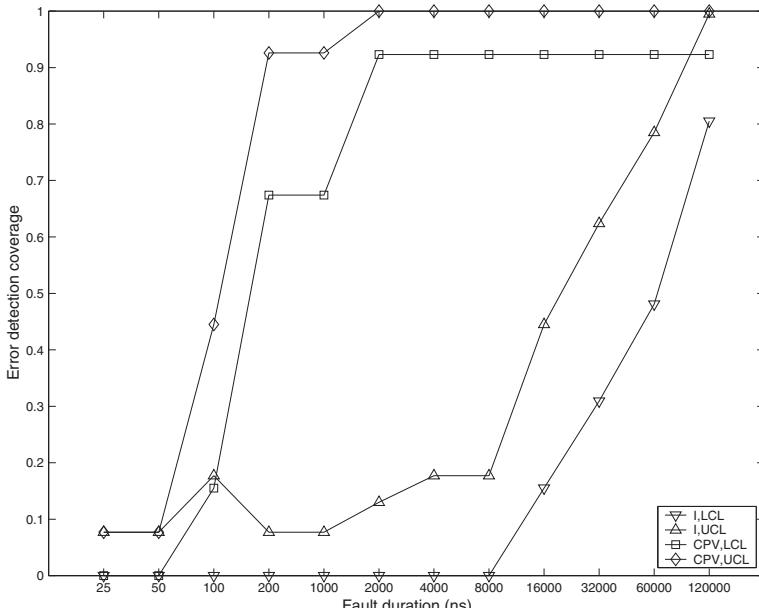


Figure 10.6. 90% confidence intervals of error-detection coverage [I—initial (physical fault injection); CPV—checking for protocol violations (simulated fault injection); LCL—lower confidence limit; UCL—upper confidence limit.]

Simulated fault injection is used to evaluate the effectiveness of a new detection mechanism, based on checking for protocol violations. Experiments prove that the protocol checker provides better error detection, especially for transients greater than 200 ns. CPV, LCL and CPV, UCL plots represent the confidence interval of the improved error-detection coverage.

‡

Problems

- Assume that CPU activity is being probed and let the i th observation $X_i = 0$ if the CPU is idle and 1 otherwise. Assume that the successive observations are sufficiently separated in time so as to be independent. Assume that X is a Bernoulli random variable with parameter p . Thus the expected utilization is p . We use the sample mean \bar{X} as an estimator \hat{P} of p . Determine an approximate $100(1 - \alpha)\%$ confidence interval for p . Next assume that we wish to determine the sample size that is required to attain a measurement error at most equal to E with confidence coefficient $1 - \alpha$. Then show that

$$n = \left\lceil p(1-p) \left(\frac{z_{\alpha/2}}{E} \right)^2 \right\rceil$$

Since this formula requires a prior knowledge of p (or its estimate), we would like an approximation here. Since for values of p for which $0.3 < p < 0.7$, $p(1-p)$ is close to 0.25, an approximate sample size for this range of p is obtained as:

$$n = \left\lceil \left(\frac{z_{\alpha/2}}{2E} \right)^2 \right\rceil.$$

(This expression is known to give conservative results.)

- Returning to the text example of inspection of a lot of RAM chips, obtain a 95% confidence interval for p using the following two methods and compare with the one obtained in the text using the Poisson approximation:
 - Either by consulting an extensive table for the CDF of the binomial distribution or by using a Mathematica program, obtain a one-sided confidence interval for p using exact binomial probabilities.
 - Compute the required one-sided confidence interval using the normal approximation to the binomial.
- Obtain a distribution-free confidence interval for population median $\pi_{0.5}$ of a continuous population by first ordering the random sample X_1, X_2, \dots, X_n , with the resulting order statistics denoted by Y_1, Y_2, \dots, Y_n . Now the observed values y_i and y_j for $i < j$ can be used to provide a confidence interval (y_i, y_j) of the median $\pi_{0.5}$. Show that the corresponding confidence coefficient is given by

$$\gamma = \sum_{k=i}^{j-1} \binom{n}{k} \left(\frac{1}{2} \right)^k \left(\frac{1}{2} \right)^{n-k}$$

Generalize to obtain a confidence interval for population percentile π_p , where π_p is defined by

$$P(X \leq \pi_p) = p.$$

4. In order to estimate the fault-detection coverage c of a fault-tolerant computer system, 200 random faults were inserted [GAY 1978]. The recovery mechanism successfully detected 178 of these faults. Determine 95% one-sided confidence intervals for the coverage c using exact binomial probabilities and using the normal and the Poisson approximations to the binomial.

10.2.4 Estimation related to Markov Chains

So far, we have restricted our attention to estimating parameters related to the probability distribution of a single random variable X . In this section we will study the estimation of parameters of a Markov chain.

10.2.4.1 Discrete-Time Markov Chains. First, consider a homogeneous discrete-time Markov chain with a finite number of states. Let the state space be $\{1, 2, \dots, m\}$. Assume that the chain is observed for a total of n transitions so that N_{ij} is the number of transitions from state i to state j ($i, j = 1, 2, \dots, m$). Let $N_i = \sum_{j=1}^m N_{ij}$ be all transitions out of state i , and note that $n = \sum_{i=1}^m N_i$ is a fixed constant while N_i and N_{ij} are random variables. Particular values of these random variables are denoted by n_i and n_{ij} , respectively. From these observations, we wish to estimate the m^2 elements of the transition probability matrix $P = [p_{ij}]$. It can be shown that the maximum-likelihood estimator, \hat{P}_{ij} , of p_{ij} , is given by [BHAT 1984]:

$$\hat{P}_{ij} = \frac{N_{ij}}{N_i}. \quad (10.18)$$

Example 10.25

Consider the CPU of a computer system modeled as a three-state homogeneous discrete-time Markov chain. The states are indexed 1, 2, and 3 and respectively denote the CPU in supervisor state, user state, and idle state. We record the states of the CPU at 21 successive time instants, and the recorded sequence is

1	2	3	3	2	1	1	2	2	3	2
3	1	3	2	3	1	2	3	1	2	

With 21 observations, the number of transitions, n , is 20. From the data, we derive the values of n_{ij} and n_i to be

n_{ij}	1	2	3	
1	1	4	1	6
2	1	1	5	7
3	3	3	1	7
				20

Thus, the maximum-likelihood estimate of the transition probability matrix P for the CPU (modeled as a DTMC) is given by

$$\hat{P} = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{1}{7} & \frac{1}{7} & \frac{5}{7} \\ \frac{3}{7} & \frac{3}{7} & \frac{1}{7} \end{bmatrix}$$

‡

A confidence interval for p_{ij} may be obtained by assuming that a fixed number n_i of transitions out of state i have been observed, out of which a random number N_{ij} of transitions are to state j . Owing to the assumptions of a DTMC, N_{ij} is then binomially distributed so that N_{ij} is $B(k; n_i, p_{ij})$. Now the methods of Section 10.2.3.4 can be used to derive the confidence interval for p_{ij} . Thus, if n_i is sufficiently large and if p_{ij} is not close to 0 or 1, an approximate $100(1 - \alpha)\%$ confidence interval for p_{ij} is given by

$$\left[\frac{N_{ij}}{n_i} - z_{\alpha/2} \frac{\sqrt{N_{ij}(1 - N_{ij}/n_i)}}{n_i}, \frac{N_{ij}}{n_i} + z_{\alpha/2} \frac{\sqrt{N_{ij}(1 - N_{ij}/n_i)}}{n_i} \right].$$

10.2.4.2 Estimating Parameters of an M/M/1 Queue. Next consider estimating parameters of a homogeneous continuous-time Markov chain, such as a simple birth-death process with constant birth rate λ and constant death rate μ . This corresponds to an $M/M/1$ queue with an arrival rate λ and a service rate μ . To estimate the arrival rate λ , we observe that the arrival process is Poissonian and therefore the method of Section 10.2.3.3 is applicable. If S_n is the time required to observe n arrivals, then the maximum-likelihood estimator of λ is

$$\hat{\Lambda} = \frac{n}{S_n}.$$

Also, since $2\lambda S_n$ has a chi-square distribution with $2n$ degrees of freedom, a $100(1 - \alpha)\%$ confidence interval for λ is given by

$$\left(\frac{\chi^2_{2n;1-\alpha/2}}{2s_n}, \frac{\chi^2_{2n;\alpha/2}}{2s_n} \right).$$

In order to estimate the service rate μ , we note that, owing to our assumptions, the service times are independent exponentially distributed random variables; hence the method of Section 10.2.3.2 is applicable. Thus, if service times X_1, X_2, \dots, X_m have been observed, and if we let

$$Y_m = \sum_{i=1}^m X_i,$$

the maximum-likelihood estimator of μ is given by

$$\hat{M} = \frac{m}{Y_m}.$$

Y_m may also be interpreted as the total busy time of the server during the observation period. Noting that $2\mu Y_m$ has a X_{2m}^2 distribution, we get a $100(1 - \alpha)\%$ confidence interval for μ as

$$\left(\frac{\chi_{2m;1-\alpha/2}^2}{2y_m}, \frac{\chi_{2m;\alpha/2}^2}{2y_m} \right).$$

Server utilization ρ is now estimated by

$$\hat{R} = \frac{\hat{M}}{\hat{M}} = \frac{n/S_n}{m/Y_m} = \frac{Y_m/m}{S_n/n}.$$

To obtain confidence intervals for ρ , we use the ratio

$$\frac{\hat{R}}{\rho} = \frac{(Y_m/m)/(S_n/n)}{\lambda/\mu} = \frac{2\mu Y_m/2m}{2\lambda S_n/2n}.$$

Now, since Y_m and S_n are independent, and $2\mu Y_m$ and $2\lambda S_n$ are both chi-square distributed, it follows that \hat{R}/ρ has an F distribution with $2m$ and $2n$ degrees of freedom (see Theorem 3.9). To obtain a $100(1 - \alpha)\%$ confidence interval for ρ , we write the following for some constants c and d :

$$1 - \alpha = P(c < F < d) = P\left(c < \frac{\hat{R}}{\rho} < d\right).$$

Select c and d so that $P(F \leq c) = \alpha/2$ and $P(F < d) = 1 - \alpha/2$. Then, by our usual notation, $c = f_{2m,2n;1-\alpha/2}$ and $d = f_{2m,2n;\alpha/2}$ so that the required confidence interval for ρ is given by (ρ_L, ρ_U) where

$$\rho_L = \frac{\hat{\rho}}{f_{2m,2n;\alpha/2}} \quad \text{and} \quad \rho_U = \frac{\hat{\rho}}{f_{2m,2n;1-\alpha/2}}.$$

From the confidence interval of ρ , we can obtain a confidence interval for any monotonically increasing function, $H(\rho)$, of ρ . Thus

$$(H(\rho_L), H(\rho_U))$$

is the $100(1 - \alpha)\%$ confidence interval for $H(\rho)$.

Example 10.26

Assume that a communication channel can be modeled as an $M/M/1$ queue. Suppose that we observe the time until 30 message arrivals to be 59.46 min and these 30

messages keep the channel busy for a total of 29 min. Thus, $m = n = 30$, $s_n = 59.46$ min, and $y_m = 29$ min. Point estimates of the arrival rate, the service rate, and the channel utilization are given by

$$\begin{aligned}\hat{\lambda} &= \frac{n}{s_n} = \frac{30}{59.46} = 0.505 \text{ messages per minute,} \\ \hat{\mu} &= \frac{m}{y_m} = \frac{30}{29} = 1.03 \text{ messages per minute,} \\ \hat{\rho} &= \frac{\hat{\lambda}}{\hat{\mu}} = 0.488.\end{aligned}$$

To obtain 95 percent confidence intervals for λ and μ , we use a chi-square distribution with $2m, 2n = 60$ degrees of freedom. Noting that

$$\chi^2_{60;0.025} = 83.3 \quad \text{and} \quad \chi^2_{60;0.975} = 40.48,$$

the required confidence interval for λ is $(0.34, 0.7)$ and that for μ is $(0.698, 1.436)$.

To obtain a confidence interval for ρ , we use an F distribution with $(60,60)$ degrees of freedom. Noting that

$$f_{60,60;\alpha/2} = f_{60,60;0.025} = 1.67$$

and

$$f_{60,60;1-\alpha/2} = f_{60,60;0.975} = 0.5988,$$

we obtain

$$\rho_L = \frac{\hat{\rho}}{f_{60,60;0.025}} = 0.292 \quad \text{and} \quad \rho_U = \frac{\hat{\rho}}{f_{60,60;0.975}} = 0.815.$$

Thus, the 95% confidence interval for ρ is $(0.292, 0.815)$.

From the confidence interval for ρ , we can obtain a confidence interval for the average number of messages queued or in service, $E[N] = \rho/(1 - \rho)$. Thus, a 95% confidence interval for $E[N]$ is given by

$$\left(\frac{0.292}{1 - 0.292}, \frac{0.815}{1 - 0.815} \right) \text{ or } (0.412, 4.405).$$

Much more data must be collected if we want to narrow down this interval.

#

10.2.4.3 Estimation of Availability. Similar to the estimation in $M/M/1$ queue, we consider estimating the steady-state availability from a two-state continuous-time Markov chain (Figure 10.7). In this model, the parameters to be estimated are failure rate λ and repair rate μ , or MTTF

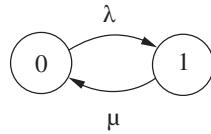


Figure 10.7. Two-state CTMC

and MTTR. The steady-state availability is computed as

$$A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} = \frac{\mu}{\lambda + \mu} = \frac{1}{1 + \lambda/\mu} = \frac{1}{1 + \rho}, \quad (10.19)$$

where ρ is the ratio λ/μ .

Since the failure time and the repair time are both exponentially distributed, this is very similar to the $M/M/1$ queue case where interarrival times and service times are both exponentially distributed. Assume we observed n failure events and repair events, the total failure time is S_n and the total repair time is Y_n . The maximum-likelihood estimator of λ is

$$\hat{\lambda} = \frac{n}{S_n},$$

and a $100(1 - \alpha)\%$ confidence interval for λ is given by

$$\left(\frac{\chi^2_{2n;1-\alpha/2}}{2s_n}, \frac{\chi^2_{2n;\alpha/2}}{2s_n} \right).$$

The maximum-likelihood estimator of μ is

$$\hat{M} = \frac{n}{Y_n}.$$

and a $100(1 - \alpha)\%$ confidence interval for μ is

$$\left(\frac{\chi^2_{2n;1-\alpha/2}}{2y_n}, \frac{\chi^2_{2n;\alpha/2}}{2y_n} \right).$$

The ratio λ/μ is estimated by

$$\hat{\rho} = \frac{\hat{\lambda}}{\hat{M}} = \frac{n/S_n}{n/Y_n} = \frac{Y_n/n}{S_n/n} = \frac{Y_n}{S_n},$$

and a $100(1 - \alpha)\%$ confidence interval for ρ is given by (ρ_L, ρ_U) , where

$$\rho_L = \frac{\hat{\rho}}{f_{2n,2n;\alpha/2}} \quad \text{and} \quad \rho_U = \frac{\hat{\rho}}{f_{2n,2n;1-\alpha/2}}.$$

The MLE estimate of A is $\hat{A} = 1/(1 + \hat{\rho})$. Since the availability A is a monotonically decreasing function of ρ , the $100(1 - \alpha)\%$ confidence interval for A is

$$\left(\frac{1}{1 + \rho_U}, \frac{1}{1 + \rho_L} \right).$$

Next, we consider the $100(1 - \alpha)\%$ upper one-sided confidence interval for A , $(A_L, 1)$. A_L is given by

$$A_L = \frac{1}{1 + \frac{\hat{\rho}}{f_{2n,2n;1-\alpha}}} = \frac{1}{1 + \frac{\frac{1}{\hat{A}} - 1}{f_{2n,2n;1-\alpha}}}. \quad (10.20)$$

Example 10.27

Assume for a certain system that we observed only one failure event and one repair event, so that $n = 1$, $S_1 = 999$ h, and $Y_1 = 1$ h. The point estimate of steady-state availability is

$$\hat{A} = \frac{1}{1 + \hat{\rho}} = 0.999.$$

To obtain 95% confidence intervals for A , we use an F distribution with $(2, 2)$ degrees of freedom. Noting that

$$f_{2,2;\alpha/2} = f_{2,2;0.025} = 39$$

and

$$f_{2,2;1-\alpha/2} = f_{2,2;0.975} = 0.02564,$$

we compute the 95% confidence interval for A as $(0.9624, 1)$.

If we observed 10 failure events and 10 repair events, so that $n = 10$, $S_{10} = 9990$ h, and $Y_{10} = 10$ h. The point estimate of availability is unchanged:

$$\hat{A} = \frac{1}{1 + \hat{\rho}} = 0.999.$$

However, we use an F distribution with $(20, 20)$ degrees of freedom to calculate the confidence interval for A :

$$f_{20,20;\alpha/2} = f_{20,20;0.025} = 2.4645,$$

$$f_{20,20;1-\alpha/2} = f_{20,20;0.975} = 0.4058;$$

thus the confidence interval for A is narrowed to $(0.9975, 0.9996)$.

Next we derive the upper one-sided confidence interval for A . In the first case ($n = 1$), we calculate

$$f_{2,2;1-\alpha} = f_{2,2;0.95} = 0.05263.$$

So the 95% upper one-sided confidence interval for A is $(0.9813, 1)$.

In the second case ($n = 10$), we calculate

$$f_{20,20;1-\alpha} = f_{20,20;0.95} = 0.4708;$$

thus the confidence interval for A is narrowed to $(0.9979, 1)$.

‡

Example 10.28

In this example, we investigate how to achieve “5 nines” availability with 95% confidence. First, it is obvious that the point estimate of A should be above “5 nines”:

$$\hat{A} > 0.99999.$$

Second, because we have seen the width of the confidence interval narrows as the number of samples increases, we need a sufficient number of samples to make $A_L > 0.99999$. Consider the 95% upper one-sided confidence interval. From equation (10.20), we have

$$A_L = \frac{1}{1 + \frac{\frac{1}{\hat{A}} - 1}{f_{2n,2n;0.95}}}.$$

We plot the number of samples n against the lower boundary of the interval A_L for different point estimates \hat{A} in Figure 10.8. We observe that to achieve 95% upper one-sided confidence interval as $(0.99999, 1)$, the least number of samples required is

$$n = \begin{cases} 445 & \text{when } \hat{A} = 0.999991 \\ 105 & \text{when } \hat{A} = 0.999992 \\ 12 & \text{when } \hat{A} = 0.999995 \end{cases}.$$

In other words, the lower the point estimate of availability, larger must be the number of samples from which this estimate is computed in order for the given availability confidence interval to be ascertained.

‡

10.2.4.4 Estimation for a Semi-Markov Process. One way to describe a semi-Markov chain, which applies only to independent SMPs (Semi-Markov Processes) [ROSS 1983], is to define a transition probability matrix P and a vector $\mathbf{H}(t)$. This is called the *two-stage method* since the transitions of the SMP can be assumed to take place in two stages. Consider a transition from state i to state j . In the first stage, the chain stays in state i for some amount of time described by the sojourn time distribution $H_i(t)$. In the second stage, the chain moves to state j determined by the probability p_{ij} .

In a study of software aging [VAID 1999], a measurement-based system workload model is constructed based on this two-stage SMP method. Using a monitoring tool, time-ordered values for a number of variables pertaining

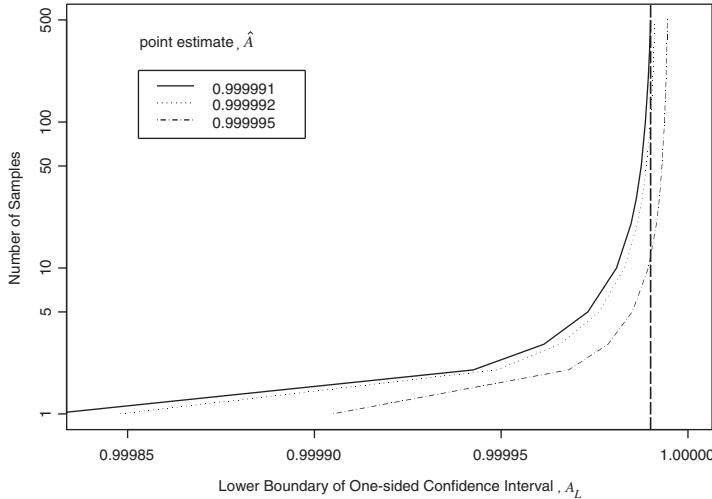


Figure 10.8. Number of samples n versus lower boundary, A_L

to CPU activity and file system I/O—*cpuContextSwitch*, *sysCall*, *pageIn*, and *pageOut*—are obtained at regular intervals over a period of time from a single workstation. Therefore, during any time interval, a point in a four-dimensional space, $x = (\textit{cpuContextSwitch}, \textit{sysCall}, \textit{pageIn}, \textit{pageOut})$, represents the measured workload.

Cluster analysis Next, the data points are partitioned into clusters that contain similar points based on some predefined criteria. A statistical clustering algorithm is used to achieve this. The goal of cluster analysis is to determine a partition of a given set of points into groups or clusters such that the points in a single cluster are more similar, according to a certain criterion, to each other than to points in different clusters. In our case, we have used an iterative nonhierarchical clustering algorithm called *Hartigan's k-means clustering algorithm* [HART 1975]. This is one of the most popular iterative nonhierarchical clustering algorithms and has been used in workload characterization and is known to work well [DEVA 1989, FERR 1983]. The objective of this algorithm is to divide a given set of points into k clusters so that the intracluster sum of squares is minimized. The algorithm starts with k initial points (which may be random or prespecified) that are taken to be centroids of k clusters. New points are assigned to these clusters according to the closest centroid, and the centroids are recomputed when all the points have been assigned. This procedure is repeated several times with the means from one iteration taken to be the initial points for the next iteration. In other words, the algorithm finds a partition $\Pi = (C_1, C_2, \dots, C_k)$ with k nonempty clusters such that the sum of the squares, S , from each point to its corresponding

centroid (center of mass) is minimized:

$$\text{minimize } S = \sum_{i=1}^k \sum_{x_j \in C_i} |x_j - \bar{x}_i|^2$$

where \bar{x}_i is the centroid of cluster C_i .

If the variables for clustering are not expressed in homogeneous units, a scale change or normalization must be performed. In this case, the normalization method based on the following transformation [FERR 1983] is used:

$$x'_{ij} = \frac{x_{ij} - \min_i\{x_{ij}\}}{\max_i\{x_{ij}\} - \min_i\{x_{ij}\}}$$

where x_{ij} is the value of the j th parameter for the i th component and x'_{ij} is the normalized value. This transformation ensures that the values of all the variables are in the range 0 to 1. We need to eliminate the outliers in the data before applying the scaling technique since they tend to distort the transformation. Outliers in the data were identified and eliminated by an analysis of the cumulative distribution of each parameter, although they are assigned to clusters in the final stage. The statistics for the workload variables measured are shown in Table 10.2. The first and the third quantiles are the 25th percentiles and 75th percentiles, respectively.

The k -means clustering algorithm was applied to the workload data, and this resulted in 11 clusters. The statistics for the workload clusters are shown in Table 10.3. Also shown in the table are the percentage of the sample data points in each cluster. It can be observed that more than 75% of the points belong to clusters 7, 8, and 10, which are relatively light workload states. Cluster 1, 2, 3, and 11 are high workload states that contain significantly fewer data points.

The transition probability matrix The next step, after the clusters and centroids are identified, is to estimate the transition probabilities from one state to another. The transition probability p_{ij} from a state i to a state j can be estimated from the sample data using formula (10.18) [DEVA 1989]. Before

TABLE 10.2. Statistics for the workload variables measured

Variable	1st			3rd		
	Min	quantile	Median	Mean	quantile	Max
<i>cpuContextSwitch</i>	5,598	10,300	10,990	20,650	18,990	386,100
<i>sysCall</i>	19,170	33,980	37,960	41,580	39,630	672,900
<i>pageIn</i>	0	9	9	26.17	12	2,522
<i>pageOut</i>	0	0	0	5.426	0	6,227

TABLE 10.3. Statistics for the workload clusters

Cluster	Cluster centroid				% of data
	no.	cpuContextSwitch	sysCall	pageOut	
1	48405.16	94194.66	5.16	677.83	0.98
2	54184.56	122229.68	5.39	81.41	0.76
3	34059.61	193927.00	0.02	136.73	0.93
4	20479.21	45811.71	0.53	243.40	1.89
5	21361.38	37027.41	0.26	12.64	7.17
6	15734.65	54056.27	0.27	14.45	6.55
7	37825.76	40912.18	0.91	12.21	11.77
8	11013.22	38682.46	0.03	10.43	42.87
9	67290.83	37246.76	7.58	19.88	4.93
10	10003.94	32067.20	0.01	9.61	21.23
11	197934.42	67822.48	415.71	184.38	0.93

this, the number of workload states is reduced to eight by merging clusters $\{1, 2, 3\}$ and $\{4, 5\}$. Thus we get $W_1 = \{1, 2, 3\}$, $W_2 = \{4, 5\}$, $W_3 = \{6\}$, $W_4 = \{7\}$, $W_5 = \{8\}$, $W_6 = \{9\}$, $W_7 = \{10\}$, and $W_8 = \{11\}$. Clusters considered for merging were clusters whose centroids were relatively close to each other and clusters with a small percentage of data points in them. This is done mainly to reduce and simplify computations. State-transition probabilities were estimated for the workload states, and the resulting transition probability matrix, P , of the embedded discrete-time Markov chain of the SMP, is shown below:

$$\hat{P} = \begin{bmatrix} 0.000 & 0.155 & 0.224 & 0.129 & 0.259 & 0.034 & 0.165 & 0.034 \\ 0.071 & 0.000 & 0.136 & 0.140 & 0.316 & 0.026 & 0.307 & 0.004 \\ 0.122 & 0.226 & 0.000 & 0.096 & 0.426 & 0.000 & 0.113 & 0.017 \\ 0.147 & 0.363 & 0.059 & 0.000 & 0.098 & 0.216 & 0.088 & 0.029 \\ 0.033 & 0.068 & 0.037 & 0.011 & 0.000 & 0.004 & 0.847 & 0.000 \\ 0.070 & 0.163 & 0.023 & 0.535 & 0.116 & 0.000 & 0.023 & 0.070 \\ 0.022 & 0.049 & 0.003 & 0.003 & 0.920 & 0.003 & 0.000 & 0.000 \\ 0.307 & 0.077 & 0.154 & 0.231 & 0.077 & 0.154 & 0.000 & 0.000 \end{bmatrix}.$$

Sojourn time distributions To completely specify the semi-Markov process, the distribution of sojourn time in each workload state needs to be estimated (see Table 10.4). The empirical distributions for all the workload states are fitted to either two-stage hyperexponential, two-stage hypoexponential, or simple exponential distribution functions.

TABLE 10.4. Sojourn time distributions in the workload states

Workload state	Sojourn time distribution $F(t)$	Distribution type
W_1	$1 - 1.602919e^{-0.9t} + 0.6029185e^{-2.392739t}$	Hypoexponential
W_2	$1 - 0.9995e^{-0.4459902t} - 0.0005e^{-0.007110071t}$	Hyperexponential
W_3	$1 - 0.9952e^{-0.3274977t} - 0.0048e^{-0.0175027t}$	Hyperexponential
W_4	$1 - 0.841362e^{-0.3275372t} - 0.158638e^{-0.03825429t}$	Hyperexponential
W_5	$1 - 1.425856e^{-0.56t} + 0.4258555e^{-1.875t}$	Hypoexponential
W_6	$1 - 0.80694e^{-0.5509307t} - 0.19306e^{-0.03705756t}$	Hyperexponential
W_7	$1 - 2.86533e^{-1.302t} + 1.86533e^{-2t}$	Hypoexponential
W_8	$1 - 0.9883e^{-0.2655196t} - 0.0117e^{-0.02710147t}$	Hyperexponential

To estimate the rate parameter of the exponential distribution, we use formula (10.5). For the two parameters λ_1 and λ_2 of the hyperexponential distribution, we use equations (10.1) and (10.2). For the two parameters λ_1 and λ_2 of the hypoexponential distribution, we use formula (10.3). The fitted distributions are tested using the Kolmogorov–Smirnov test (Section 10.3.4) at a significance level of 0.01.

Model validation The semi-Markov model for the system workload needs to be validated. The steady-state probability of occupying a particular workload state computed from the model was compared to the estimated probability from the observed data. The steady-state probabilities for the semi-Markov chain are computed as follows. First, the steady-state probabilities $[\nu_0, \nu_1, \dots]$ for the embedded discrete-time Markov chain are computed using the linear system of equations (7.18) and (7.19):

$$\mathbf{v} = \mathbf{v}P$$

$$\mathbf{v} \cdot \mathbf{e} = 1$$

where P is the transition probability matrix and \mathbf{e} is a column vector with all 1s. Next, the steady-state probabilities for the semi-Markov chain are computed as [CINL 1975]:

$$\pi_i = \frac{v_i h_i}{\sum_j v_j h_j}$$

where π_i is the steady state probability for the SMP for state i and h_i is the mean sojourn time in state i . The probabilities from the measured data are estimated as the ratio of the length of time the system was in that workload state to the total length of the period of observation. The results are shown in Table 10.5.

TABLE 10.5. Comparison of state occupancy probabilities (expressed as percentage)

<i>State</i>	<i>Observed value</i>	<i>Value from model</i>	<i>% difference</i>
W_1	2.664146	2.8110	5.512235
W_2	9.058096	8.3464	7.857015
W_3	6.548642	6.0576	7.498379
W_4	11.77381	10.8480	7.863300
W_5	42.86696	44.4310	3.648591
W_6	4.932967	4.5767	7.222165
W_7	21.22723	22.1030	4.125691
W_8	0.928154	0.82577	11.030928

It can be seen that the computed values from the model and the actual observed values match quite closely. This validates the model building methodology, and so the semi-Markov process obtained can be taken to model the real-system workload reasonably well.

Problems

- For an $M/M/1$ queue, 955 arrivals were observed in a period of 1000 time units, and the server was found to be busy for 660 time units. Compute 95% confidence intervals for the following quantities:
 - The arrival rate λ .
 - The average service time $1/\mu$.
 - The server utilization ρ .
 - The average queue length $E[N]$.
 - The average response time $E[R]$.

- Give an argument for determining a $100(1 - \alpha)\%$ confidence interval for the server utilization ρ as

$$\rho_L \leq \rho \leq \rho_U$$

in the following cases:

- $M/E_k/1$ queuing system:

$$\rho_L = \frac{\hat{\rho}}{f_{2mk,2n;\alpha/2}} \quad \text{and} \quad \rho_U = \frac{\hat{\rho}}{f_{2mk,2n;1-\alpha/2}}.$$

- $E_k/M/1$ queuing system:

$$\rho_L = \frac{\hat{\rho}}{f_{2m,2nk;\alpha/2}} \quad \text{and} \quad \rho_U = \frac{\hat{\rho}}{f_{2m,2nk;1-\alpha/2}}.$$

3. We have noted (in Chapter 8) that in an $M/M/1$ queue the response time is exponentially distributed. Given a sample of observations of response times for n successive jobs, can we use methods of Section 10.2.3.2 to obtain confidence intervals for the parameter $\delta [= \mu(1 - \rho)]$ of the response time distribution?

10.2.5 Estimation with Dependent Samples

So far, we have assumed that the sample random variables X_1, X_2, \dots, X_n are mutually independent. Measurements obtained from real systems, however, often exhibit dependencies. For example, there is a high correlation between the response times of consecutive requests to a file server. Although observations taken from such a system do not satisfy the definition of a random sample, the behavior of the system can be modeled as a stochastic process, and the observations made are then a portion of one particular realization of the process. If the stochastic process is a Markov chain, then, by noting special properties of such a process, we can make use of the methods discussed in Section 10.2.4. We consider the more general case here.

Consider a discrete-time stochastic process (or a stochastic sequence) $\{X_i \mid i = 1, 2, \dots\}$ (the treatment can also be generalized to the case of a continuous-time stochastic process). We observe the sequence for n time units to obtain the values x_1, x_2, \dots, x_n . The observed quantities are values of dependent random variables X_1, X_2, \dots, X_n . Assume that the process has an index-invariant mean, $\mu = E[X_i]$, and an index-invariant variance, $\sigma^2 = \text{Var}[X_i]$.

As before, the sample mean

$$\bar{X} = \sum_{i=1}^n \frac{X_i}{n}$$

is a consistent unbiased point estimator of the population mean. However, derivation of confidence intervals for μ poses a problem, since the variance of \bar{X} is not σ^2/n any longer. Assume that the sequence $\{X_i\}$ is wide-sense stationary, so that the autocovariance function

$$K_{j-i} = E[(X_i - \mu)(X_j - \mu)] = \text{Cov}(X_i, X_j)$$

is finite and is a function only of $|i - j|$. Then the variance of the sample mean is given by

$$\begin{aligned} \text{Var}[\bar{X}] &= \frac{1}{n^2} \left\{ \sum_{i=1}^n \text{Var}[X_i] + \sum_{\substack{i,j=1 \\ i \neq j}}^n \text{Cov}(X_i, X_j) \right\} \\ &= \frac{\sigma^2}{n} + \frac{2}{n} \sum_{j=1}^{n-1} \left(1 - \frac{j}{n}\right) K_j. \end{aligned}$$

As n approaches infinity

$$\lim_{n \rightarrow \infty} n \text{Var}[\bar{X}] = \sigma^2 + 2 \sum_{j=1}^{\infty} K_j = \sigma^2 a, \quad \text{where } a = 1 + 2 \sum_{j=1}^{\infty} \frac{K_j}{\sigma^2}.$$

It can be shown under rather general conditions that the statistic

$$\frac{\bar{X} - \mu}{\sigma \sqrt{a/n}}$$

of the correlated data approaches the standard normal distribution as n approaches infinity. Therefore, an approximate $100(1 - \alpha)\%$ confidence interval for μ is given by

$$\bar{X} \pm \sigma z_{\alpha/2} \sqrt{\frac{a}{n}}.$$

It is for this reason that the quantity n/a is called the *effective size* of the independent samples when the correlated sample size is n . The quantity $\sigma^2 a$ is an unknown, however, and it must be estimated from the observed data.

The need to estimate $\sigma^2 a$ can be avoided by using the method of **independent replications**. (For other methods and additional details, see Fishman [FISH 1978], and Kleijnen and van Groenendaal [KLEI 1992].) We replicate the experiment m times, with each experiment containing n observations. If the initial state of the stochastic sequence is chosen randomly in each of the m experiments, then the results of the experiments will be independent although n observations in a single experiment are dependent.

Let the i th observation in the j th experiment be the value $x_i(j)$ of a random variable $X_i(j)$. Let the sample mean and the sample variance of the j th experiment be denoted by $\bar{X}(j)$ and $S^2(j)$, respectively, where

$$\bar{X}(j) = \frac{1}{n} \sum_{i=1}^n X_i(j)$$

and

$$S^2(j) = \frac{1}{n-1} \left(\sum_{i=1}^n [X_i(j) - \bar{X}(j)]^2 \right).$$

From the individual sample means, we obtain a point estimator of the population mean μ to be

$$\bar{X} = \frac{1}{m} \sum_{j=1}^m \bar{X}(j) = \frac{1}{mn} \sum_{j=1}^m \sum_{i=1}^n X_i(j).$$

Note that $\bar{X}(1), \bar{X}(2), \dots, \bar{X}(m)$ are m independent and identically distributed random variables (hence they define a random sample of size m). Let the common variance of $\bar{X}(j)$ be denoted by v^2 . The variance v^2 can be estimated as

$$V^2 = \frac{1}{m-1} \sum_{j=1}^m [\bar{X}(j) - \bar{X}]^2 = \frac{1}{m-1} \sum_{j=1}^m \bar{X}^2(j) - \frac{m}{m-1} \bar{X}^2.$$

Since an estimate of the variance is used, the statistic $(\bar{X} - \mu)\sqrt{m}/V$ is approximately t -distributed with $(m - 1)$ degrees of freedom. Therefore, a $100(1 - \alpha)\%$ confidence interval for μ is given by

$$\bar{x} \pm \frac{t_{m-1;\alpha/2}v}{\sqrt{m}}.$$

Example 10.29

We are interested in estimating the average response time of a Web server. For this purpose 16 independent experiments are conducted, with each experiment measuring 20 successive response times. The following data are recorded:

j	$\bar{x}(j)$ (seconds)	$\bar{x}^2(j)$
1	0.52	0.2704
2	1.03	1.0609
3	0.41	0.1681
4	0.62	0.3844
5	0.55	0.3025
6	0.43	0.1849
7	0.92	0.8464
8	0.88	0.7744
9	0.67	0.4489
10	0.29	0.0841
11	0.87	0.7569
12	0.72	0.5184
13	0.61	0.3721
14	0.45	0.2025
15	0.98	0.9604
16	0.89	0.7921
		8.1274

The point estimate of the average response time is

$$\bar{x} = \frac{1}{16} \sum_{j=1}^{16} \bar{x}(j) = \frac{10.84}{16} = 0.6775 \text{ s.}$$

Also

$$\begin{aligned} v^2 &= \frac{1}{15} \sum_{j=1}^{16} \bar{x}^2(j) - \frac{16}{15}(0.6775)^2 \\ &= 0.5418 - 0.4896 \\ &= 0.052. \end{aligned}$$

Now, for the t distribution with 15 degrees of freedom, we have

$$t_{15;0.025} = 2.131.$$

Therefore

$$\bar{x} \pm \frac{2.131v}{\sqrt{m}} = 0.6775 \pm 2.131 \cdot \sqrt{\frac{0.052}{16}},$$

or $(0.556, 0.799)$ is a 95% confidence interval for the average response time. ‡

Problems

1. The sample mean and sample variance of response times for 10 sets of 1000 jobs were measured. For the first set, the total CPU busy time and total time of completion were also measured.

Sample no.	Mean response time for 1000 jobs (s)	Sample variance of response time (s^2)	Total time of completion (s)	CPU busy time (s)
1	1.8	1.5	1010	640
2	1.6	1.4		
3	1.83	2.18		
4	1.37	0.65		
5	1.67	1.52		
6	1.62	1.59		
7	1.84	2.10		
8	1.52	0.92		
9	1.59	1.01		
10	1.73	1.30		

Applying the method of Section 10.2.4.2, and assuming that the system is $M/M/1$, derive a 90% confidence interval for the average response time using the first sample. Next, using all 10 samples and the method described in this section, obtain a 90% confidence interval for the average response time.

10.3 HYPOTHESIS TESTING

Many practical problems require us to make decisions about populations on the basis of limited information contained in a sample. For instance, a system administrator may have to decide whether to upgrade the capacity of the installation. The choice is binary in nature—an upgrade either takes place or does not. In order to arrive at a decision, we often make an assumption or guess about the nature of the underlying population. Such an assertion, which may or may not be valid, is called a **statistical hypothesis**—a statement about one or more probability distributions associated with the population.

Procedures that enable us to decide whether to reject or accept hypotheses, based on the information contained in a sample, are called **statistical tests**. We typically form a **null hypothesis**, H_0 , which is a claim (about a probability distribution) that we are interested in rejecting or refuting. The contradictory hypothesis is called the **alternative hypothesis**, H_1 .

For example, on the basis of experimental evidence, we may be interested in testing the hypothesis (H_0) that MTTF of a certain system exceeds a threshold value θ_0 hours. The alternate hypothesis may be $\text{MTTF} < \theta_0$. Similarly, we may be interested in testing the hypothesis that the job arrival rate λ for a certain server satisfies $\lambda = \lambda_0$.

The experimental evidence, on which the test is based, will consist of a random sample X_1, X_2, \dots, X_n , of size n as in the parameter estimation problem. Since X_i is a random variable for each i , the totality of all n -tuples will span the Euclidean n -space \Re^n . The hypothesis testing procedure consists of dividing the n -space of observations into two regions, $R(H_0)$ and $R(H_1)$. If the observed vector (x_1, x_2, \dots, x_n) lies in $R(H_1)$, we reject the null hypothesis H_0 . On the other hand, if the observed n -tuple lies in $R(H_0)$, we fail to reject H_0 . The region $R(H_0)$ is known as the **acceptance region** and the region $R(H_1)$, as the **critical** or the **rejection region**.

The possibility always exists that the null hypothesis is true but the sample lies in the rejection region, leading us to reject H_0 . This is known as the **type I error**. The corresponding probability is denoted by α and is known as the **level of significance** of the test. Similarly, if the null hypothesis is false and the sample lies in the acceptance region, leading us to a rejection of H_1 , a **type II error** is committed. The probability of type II error is denoted by β , and $1 - \beta$ is known as the **power** of the test. When we say that $P(\text{type I error}) = \alpha$ and $P(\text{type II error}) = \beta$, we mean that if a test is performed a large number of times, α proportion of the time we will reject H_0 when it is true, and β proportion of the time we will fail to reject H_0 , when in fact it is false.

An error of type I or type II leads to a wrong decision, so we must attempt to minimize these errors. If we fix the sample size n , a decrease in one type of error leads to an increase in the other type. One can also associate a cost (or a penalty) with a wrong decision and minimize the total cost of the decision.

The only way to simultaneously reduce both types of error is to increase the sample size n .

A hypothesis is said to be *simple* if all the parameters in the test are specified exactly. Thus, for example, a test of the form $H_0 : \lambda = \lambda_0$ versus $H_1 : \lambda = \lambda_1$ is a test concerning two simple hypotheses. A hypothesis such as $H_0 : \lambda \in (\lambda_L, \lambda_U)$ is a *composite* hypothesis.

10.3.1 Tests on the Population Mean

Hypothesis testing is closely related to the procedure of interval estimation. Assume that we wish to test a simple hypothesis $H_0 : \theta = \theta_0$ regarding a parameter θ of the population distribution based on a random sample X_1, X_2, \dots, X_n . The following steps can be used for this purpose:

1. Find a random variable (called a test statistic) that is a function of X_1, X_2, \dots, X_n :

$$W = W(X_1, X_2, \dots, X_n; \theta),$$

such that the distribution of W is known.

2. Choose an interval (a, b) such that

$$P[W \notin (a, b) \mid H_0 \text{ is true}] = \alpha.$$

Note that then

$$P[a < W < b \mid H_0 \text{ is true}] = 1 - \alpha,$$

that is

$$P[a < W(X_1, X_2, \dots, X_n; \theta_0) < b] = 1 - \alpha. \quad (10.21)$$

3. The actual test is then as follows. Take a sample x_1, x_2, \dots, x_n and compute $w = W(x_1, x_2, \dots, x_n; \theta_0)$; if $w \notin (a, b)$, reject H_0 in favor of H_1 ; otherwise fail to reject H_0 .

The implication is that if H_0 is true, we have $100(1 - \alpha)\%$ confidence that the observed value of the test statistic will lie in the interval (a, b) . If the observed value lies outside this interval, we know that such an event could occur with probability α (given H_0 is true). In this case, we conclude that the observations differ *significantly* (at the level of significance α) from what would be expected if H_0 were true, and we are inclined to reject H_0 .

If α is prespecified, then a and b can be determined so that the relation in equation (10.21) is satisfied. Alternatively, if a and b are specified, then α can be determined from equation (10.21).

The similarity between the four-step procedure presented above and the procedure described earlier for obtaining confidence intervals should be noted.

Specifically, let θ be an unknown parameter of the population distribution. Let (a, b) be a confidence interval for parameter θ with confidence coefficient γ . Now, while testing the hypothesis $H_0 : \theta = \theta_0$, if we accept H_0 whenever $\theta \in (a, b)$ and reject it otherwise, then the significance level α of this test is related to confidence coefficient γ by $\alpha = 1 - \gamma$.

Assume that we wish to test a hypothesis regarding the population mean μ based on a random sample of size n taken from a normal population with a known variance σ^2 :

$$H_0 : \mu = \mu_0.$$

A required statistic is easily obtained, for if \bar{X} is the sample mean; then

$$Z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}}$$

is known to be standard normal. Also let

$$Z_0 = \frac{\bar{X} - \mu_0}{\sigma / \sqrt{n}}.$$

Assume that the alternative hypothesis is

$$H_1 : \mu \neq \mu_0.$$

Since the test statistic is symmetric about zero, we choose $a = -b$. The acceptance region in terms of the test statistic will then be $(-b, b)$. As a result, the type I error probability is specified by

$$\alpha = 1 - P(-b < Z < b \mid \mu = \mu_0)$$

or

$$P(-b < Z_0 < b) = 1 - \alpha.$$

But this implies that $b = z_{\alpha/2}$. Thus the acceptance region for a level of significance α is given by

$$\{(X_1, X_2, \dots, X_n) \in \Re^n \mid -z_{\alpha/2} < Z_0 < z_{\alpha/2}\},$$

which will be abbreviated as (see Figure 10.9)

$$-z_{\alpha/2} < Z_0 < z_{\alpha/2}$$

or in terms of the sample mean as follows:

$$\mu_0 - \frac{z_{\alpha/2}\sigma}{\sqrt{n}} < \bar{X} < \mu_0 + \frac{z_{\alpha/2}\sigma}{\sqrt{n}}.$$

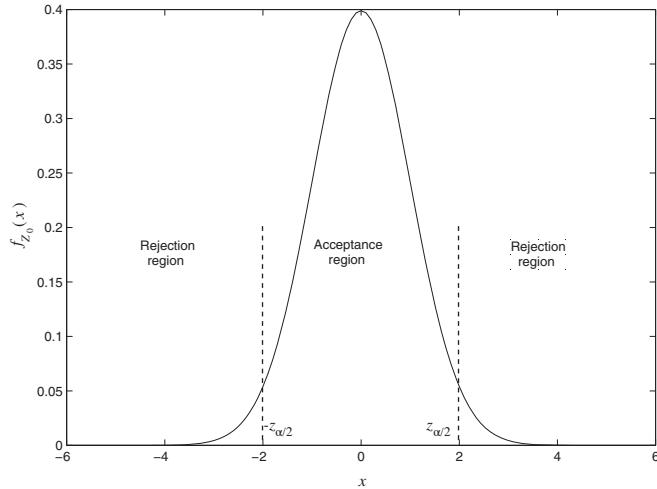


Figure 10.9. Acceptance and rejection regions for a two-sided test

The corresponding rejection (or critical) region is

$$\{(X_1, X_2, \dots, X_n) \in \mathbb{R}^n \mid |\bar{X} - \mu_0| > \frac{z_{\alpha/2}\sigma}{\sqrt{n}}\},$$

abbreviated as

$$|\bar{X} - \mu_0| > \frac{z_{\alpha/2}\sigma}{\sqrt{n}}.$$

If the alternative hypothesis is of the form

$$H_1 : \mu < \mu_0,$$

then we adopt an asymmetric acceptance region (such tests are known as *one-tailed* or *one-sided* tests) (see Figure 10.10):

$$Z_0 > -z_\alpha \quad \text{or} \quad \bar{X} > \mu_0 - \frac{z_\alpha\sigma}{\sqrt{n}},$$

and the rejection region is $\bar{X} < \mu_0 - z_\alpha\sigma/\sqrt{n}$. Similarly, if the alternative hypothesis is

$$H_1 : \mu > \mu_0,$$

then the rejection region is

$$\bar{X} > \mu_0 + \frac{z_\alpha\sigma}{\sqrt{n}}.$$

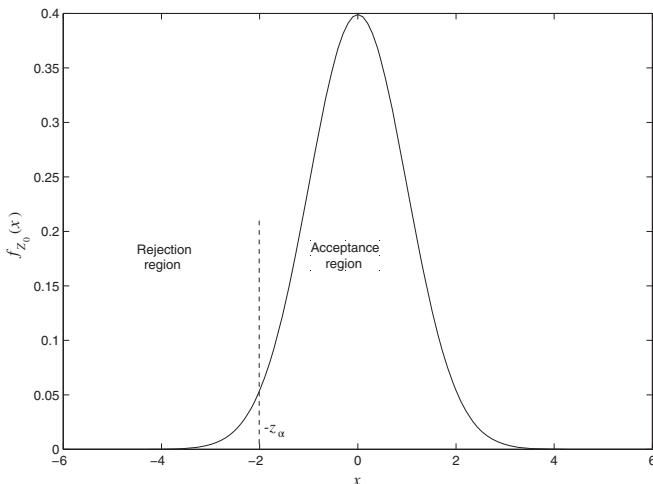


Figure 10.10. Acceptance and rejection regions for a one-sided test

Example 10.30

A program's average working-set size was found to be $\mu_0 = 50$ pages with a variance of $\sigma^2 = 900$ pages². A reorganization of the program's address space was suspected to have improved its locality and hence decreased its average working-set size. In order to judge the locality-improvement procedure, we test the hypothesis:

$$H_0 : \mu = \mu_0 \text{ versus } H_1 : \mu < \mu_0.$$

Now, since $z_\alpha = z_{0.05} = 1.645$, we have that at 5% level of significance, the critical region is

$$\bar{X} < \mu_0 - \frac{\sigma}{\sqrt{n}} z_\alpha = 50 - \frac{30}{\sqrt{n}} 1.645$$

or

$$\bar{X} < 50 - \frac{49.35}{\sqrt{n}}.$$

Thus, if 100 samples of the “improved” version of the program’s working-set size were taken and the sample average was found to be less than 45 pages, we would have reason to believe that the reorganization indeed improved program locality.

#

Instead of fixing the significance level at a value α , we may be interested in computing the probability of getting a result as extreme as, or more extreme than, the observed result under the null hypothesis. Such a probability is known as the **descriptive level** (also called the P value) of the test.

Definition (Descriptive Level). The descriptive level of a test H_0 is the smallest level of significance α at which the observed test result would be declared significant—that is, would be declared indicative of rejection of H_0 .

For instance, in Example 10.30, the descriptive level δ for an observed sample mean \bar{x} is given by

$$\delta = P(\bar{X} \leq \bar{x} | H_0) = F_{\bar{X}}(\bar{x} | H_0) = F_{Z_0}\left(\frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}\right).$$

Hence for $n = 100$ and $\bar{x} = 45$

$$\delta = F_{Z_0}\left(\frac{45 - 50}{30/10}\right) = 1 - F_{Z_0}\left(\frac{5}{3}\right) = 0.0475.$$

If the observed value of $\bar{x} = 40$, then

$$\delta = F_{Z_0}\left(\frac{40 - 50}{3}\right) = 1 - F_{Z_0}(3.33) = 0.00045.$$

Thus, if H_0 holds, observation $\bar{x} = 40$ is an extremely unlikely event, and we will be inclined to reject H_0 . On the other hand, if the observed value of $\bar{x} = 47$, then

$$\delta = F_{Z_0}\left(\frac{47 - 50}{3}\right) = F_{Z_0}(-1) = 1 - F_{Z_0}(1) = 0.1587.$$

The corresponding event can occur with about one chance in six under H_0 , and in this case we are likely not to reject H_0 .

The assumption that the population variance (σ^2) is known is very unrealistic. We can use the sample variance S^2 in place of σ^2 and derive a critical region, using the fact that the statistic

$$\frac{\bar{X} - \mu}{S/\sqrt{n}}$$

possesses the t distribution with $n - 1$ degrees of freedom. Thus, in Example 10.30, if the observed sample variance is $s^2 = 900$, then the critical region for the test $H_0 : \mu = \mu_0$ versus $H_1 : \mu < \mu_0$ is obtained as

$$\bar{X} < \mu_0 - \frac{s}{\sqrt{n}} t_{n-1; \alpha} = 50 - \frac{30}{\sqrt{16}} t_{n-1; \alpha}.$$

From the t tables, since $t_{15; 0.05} = 1.753$, the critical region for $n = 16$ and $\alpha = 0.05$ is

$$\bar{X} < 50 - \frac{52.59}{\sqrt{16}} = 36.8525,$$

while for $n = 30$ and $\alpha = 0.05$ the critical region is

$$\bar{X} < 50 - \frac{30 \cdot 1.699}{\sqrt{30}} = 50 - \frac{50.97}{\sqrt{30}} = 40.6942.$$

For $n \geq 30$, the use of the t distribution will give nearly the same results as those obtained by using the standard normal distribution.

If X does not have a normal distribution, but the sample size is sufficiently large, the procedure described above can be used to obtain an approximate critical region.

Example 10.31

Assume that we are interested in statistically testing the hypothesis that a given combinational circuit is functioning properly. Prior testing with a properly functioning circuit has shown that if its inputs are uniformly distributed over their range of values, then the probability of observing a 1 at the output is p_0 . Thus we drive the given circuit with a sequence of n randomly chosen input sets and test the hypothesis $H_0 : p = p_0$ versus $H_1 : p \neq p_0$. The test statistic X is the number of observed 1s at the output in a sample of n clock ticks. Let the critical region for the test be $|X - np_0| > n\epsilon$. The quantity ϵ is known as the *test stringency*, and the interval $(np_0 - n\epsilon, np_0 + n\epsilon)$ is the acceptance region. Assuming that H_0 is true, X has a binomial distribution with parameters n and p_0 . If n is large, and if p_0 is not close to either 0 or 1, we can use the normal approximation to the binomial distribution with mean $\mu_0 = np_0$ and variance $\sigma^2 = np_0(1 - p_0)$. Then

$$Z_0 = \frac{X - np_0}{\sqrt{np_0(1 - p_0)}}$$

has a standard normal distribution. Thus an approximate acceptance region for a level of significance α is given by

$$-z_{\alpha/2} < Z_0 < z_{\alpha/2}$$

or

$$np_0 - z_{\alpha/2}\sqrt{np_0(1 - p_0)} < X < np_0 + z_{\alpha/2}\sqrt{np_0(1 - p_0)}.$$

Thus the test stringency is derived as

$$\epsilon \simeq \frac{1}{n}z_{\alpha/2}\sqrt{np_0(1 - p_0)}.$$

Since a type I error in this circuit-testing situation implies that a properly functioning circuit is declared defective, this type of error is known as a **false alarm** in this connection. For a given test stringency ϵ , the type I error:

$$\begin{aligned} \alpha &= P(X \leq np_0 - n\epsilon \quad \text{or} \quad X \geq np_0 + n\epsilon \mid H_0) \\ &\simeq P\left(|Z_0| \geq \frac{n\epsilon}{\sqrt{np_0(1 - p_0)}}\right), \end{aligned}$$

using the normal approximation, is equal to

$$2P\left(Z_0 \geq \frac{n\epsilon}{\sqrt{np_0(1-p_0)}}\right)$$

by symmetry of the standard normal density. Now, since $p_0(1-p_0) \leq \frac{1}{4}$ for all $0 \leq p_0 \leq 1$, we have

$$\alpha \leq 2P(Z_0 \geq 2\epsilon\sqrt{n}) = 2[1 - F_{Z_0}(2\epsilon\sqrt{n})].$$

This bound will be close to the actual value of α for $0.3 \leq p_0 \leq 0.7$. Thus the bound on the probability of declaring a fault-free circuit defective depends only on the test stringency ϵ and the sample size n . It is clear that α decreases as the test stringency ϵ is increased and as the sample size is increased (i.e., α is inversely proportional to both ϵ and n) (Figure 10.11). However, as we will see later, an increase in ϵ will imply an increase in the probability of type II error (also called the **probability of escape** in this case).

‡

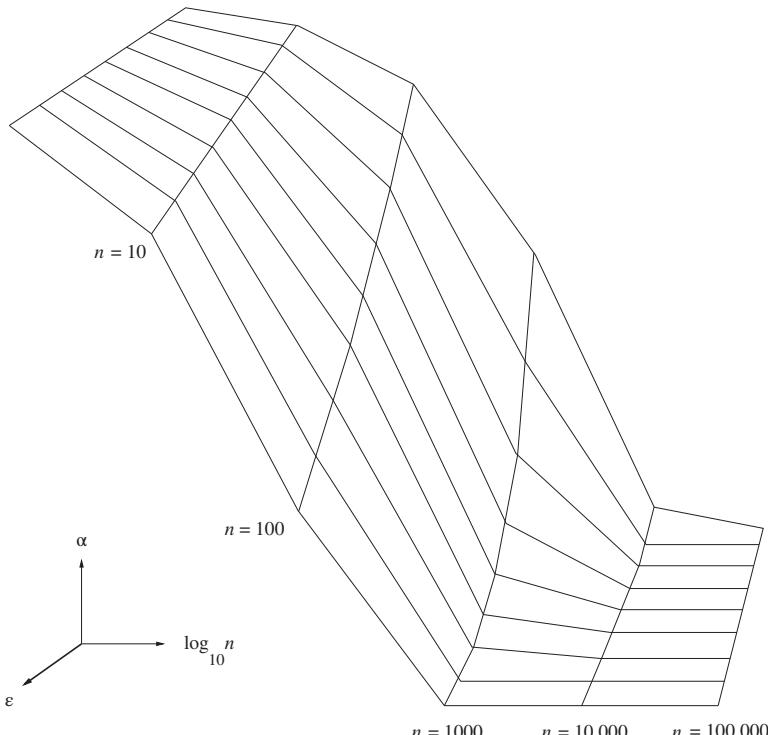


Figure 10.11. Probability of false alarm as a function of test stringency ϵ and sample size n

The inverse relationship between type I and the type II errors can be demonstrated by devising a simple test. Suppose that we always accept the null hypothesis $H_0 : \mu = \mu_0$, no matter what the outcome of sampling may be. Then clearly, the probability of rejecting the null hypothesis when it is true is zero; hence $\alpha = 0$. Simultaneously, the probability of accepting H_0 when it is false is one—that is, $\beta = 1$. Similarly, if we always reject H_0 , independent of the outcome of sampling, then $\beta = 0$ but $\alpha = 1$. In practice, we usually want to fix the probability of the type I error at some small value α , typically $\alpha = 0.01$ or 0.05 , and then devise a test that has $P(\text{type II error})$ as small as possible.

To derive an expression for the type II error probability β , consider $H_0 : \mu = \mu_0$ and $H_1 : \mu = \mu_1 > \mu_0$ for fixed values of μ_0 and μ_1 . Referring to Figure 10.12, suppose that the critical region for the test is $\bar{X} > C$. Now, if the population $X \sim N(\mu, \sigma^2)$, then $\bar{X} \sim N(\mu, \sigma^2/n)$; hence

$$\alpha = P(\bar{X} > C | H_0), \quad \beta = P(\bar{X} < C | H_1),$$

where α is the area under the pdf of the normal distribution $N(\mu_0, \sigma^2)$ from C to ∞ and β is the area under the pdf of the normal distribution $N(\mu_1, \sigma^2)$ from $-\infty$ to C . For a given level of significance α , the dividing line of the criteria can be determined as

$$C = \mu_0 + \frac{z_\alpha \sigma}{\sqrt{n}}.$$

If the allowable type II error probability β is also specified, then the minimum acceptable sample size can also be determined. We want

$$P(\bar{X} < C | H_1) \leq \beta;$$

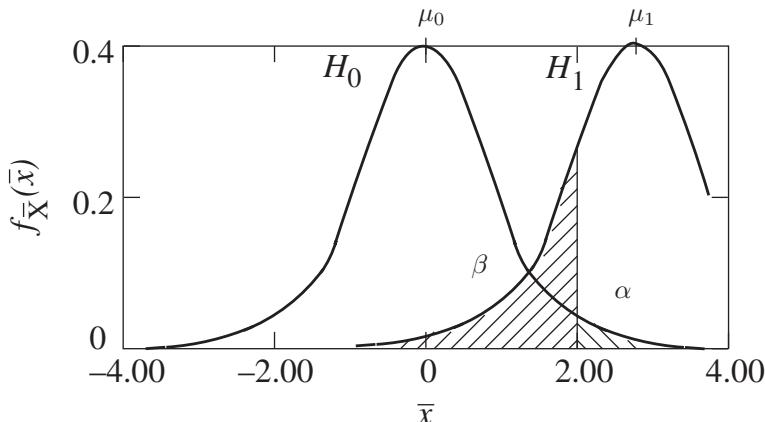


Figure 10.12. Computing types I and II errors

that is

$$P\left(Z < \frac{C - \mu_1}{\sigma/\sqrt{n}}\right) \leq \beta, \quad \frac{C - \mu_1}{\sigma/\sqrt{n}} \leq -z_\beta,$$

so we want n large enough so that $(\mu_1 - C)/(\sigma/\sqrt{n}) \geq z_\beta$; that is, for fixed α ; we have

$$\frac{\mu_1 - (\mu_0 + z_\alpha \sigma/\sqrt{n})}{\sigma/\sqrt{n}} \geq z_\beta;$$

hence

$$n \geq \frac{\sigma^2(z_\alpha + z_\beta)^2}{(\mu_1 - \mu_0)^2}.$$

Example 10.32

We wish to test the hypothesis that the response time to a trivial request (i.e., for a small file size) for a given file server is 2 s against an alternative hypothesis of 3 s. The probabilities of the two error types are specified to be $\alpha = 0.05$ and $\beta = 0.10$. Since $z_{0.05} = 1.645$ and $z_{0.10} = 1.28$ from Table 10.1, we determine the required number of response time samples to be (assuming population variance $\sigma^2 = 5.8 \text{ s}^2$):

$$\begin{aligned} n &\geq \frac{5.8 \cdot (1.645 + 1.28)^2}{(3 - 2)^2} \\ &= 50 \text{ rounded up to the nearest integer.} \end{aligned}$$

The dividing line of the criterion is

$$C = 2 + 1.645 \cdot \sqrt{\frac{5.8}{50}} = 2.556 \text{ s.}$$

Thus, if the sample mean of the observed response times exceeds 2.556 s, then the hypothesis that the system provides a 2 s response should be rejected.

#

In this analysis, we assumed that the actual mean μ was suspected to be larger than μ_0 , and therefore we used a *one-tailed* test. If we did not have such knowledge, we would use a *two-tailed* test so that the acceptance region would be $C_1 < \bar{X} < C_2$. Hence we have

$$\alpha = P(\bar{X} < C_1 \text{ or } \bar{X} > C_2 | H_0),$$

$$\beta = P(C_1 < \bar{X} < C_2 | H_1),$$

$$C_1 = \mu_0 - z_{\alpha/2} \frac{\sigma}{\sqrt{n}},$$

$$C_2 = \mu_0 + z_{\alpha/2} \frac{\sigma}{\sqrt{n}}.$$

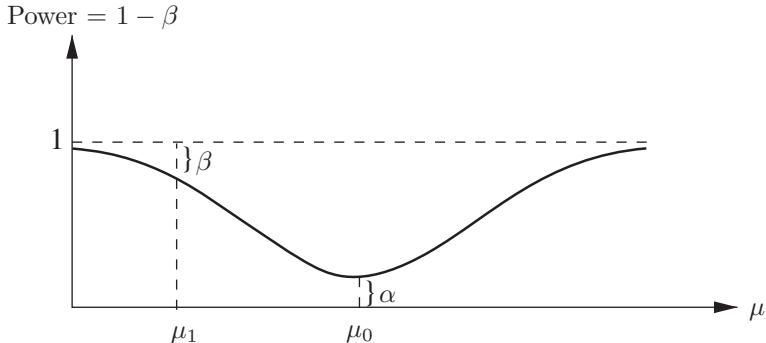


Figure 10.13. A typical power curve for a two-tailed test

Let $n(\mu, \sigma^2)$ denote the pdf of the normal random variable with mean μ and variance σ^2 . Then

$$\begin{aligned}\beta &= \int_{C_1}^{C_2} n(\mu_1, \sigma^2) dx \\ &= \int_{(C_1 - \mu_1)/(\sigma/\sqrt{n})}^{(C_2 - \mu_1)/(\sigma/\sqrt{n})} n(0, 1) dx \\ &= \int_{(\mu_0 - \mu_1)/(\sigma/\sqrt{n}) - z_{\alpha/2}}^{(\mu_0 - \mu_1)/(\sigma/\sqrt{n}) + z_{\alpha/2}} n(0, 1) dx \\ &= F_Z\left(\frac{\mu_0 - \mu_1}{\sigma/\sqrt{n}} + z_{\alpha/2}\right) - F_Z\left(\frac{\mu_0 - \mu_1}{\sigma/\sqrt{n}} - z_{\alpha/2}\right).\end{aligned}$$

Often the alternative value μ_1 of the actual mean will not be specified. In this case we could compute β as a function of μ_1 . Plot of $1 - \beta$ as a function of μ_1 is known as a **power curve**. A typical power curve for a two-tailed test is shown in Figure 10.13. Note that $1 - \beta(\mu_1)$ is the probability of rejecting $H_0 : \mu = \mu_0$ when actually $\mu = \mu_1$, so that for $\mu_1 \neq \mu_0$, the power is the probability of a correct decision. Now, if $\mu_1 = \mu_0$, then $1 - \beta(\mu_1)$ is the probability of rejecting H_0 when it should be accepted. Thus the value of the power curve at $\mu_1 = \mu_0$, $1 - \beta(\mu_0) = \alpha$. A typical power curve for a one-tailed test ($H_0 : \mu = \mu_0$, $H_1 : \mu > \mu_0$) is shown in Figure 10.14.

Example 10.33

Continuing with our example of statistically monitoring a circuit, we are testing the hypothesis $H_0 : p = p_0$ versus the alternative $H_1 : p \neq p_0$. The null hypothesis H_0 corresponds to a fault-free circuit and the alternative corresponds to a faulty circuit. Different faults may give rise to a different value of the test statistic X , the number of observed 1s at the output.

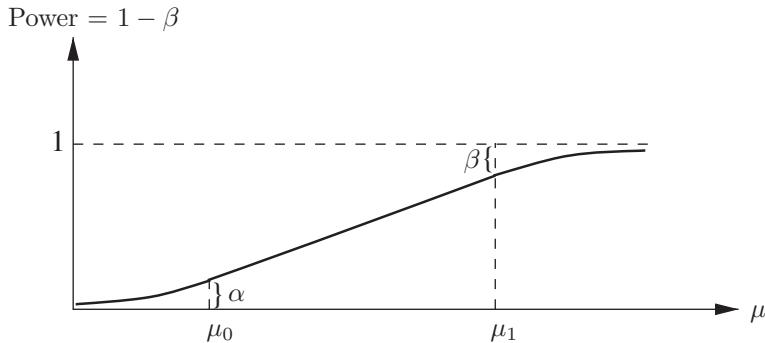


Figure 10.14. A typical power curve for a one-tailed test

If we assume that X is binomially distributed with mean np , then

$$\begin{aligned}\beta &= P(np_0 - n\epsilon < X \leq np_0 + n\epsilon \mid H_1) \\ &= \sum_{np_0 - n\epsilon < k \leq np_0 + n\epsilon} \binom{n}{k} p^k (1-p)^{n-k} \\ &= F_X(np_0 + n\epsilon) - F_X(np_0 - n\epsilon)\end{aligned}$$

Now, since X is binomially distributed with mean np and variance $np(1-p)$, it is approximately $N[np, np(1-p)]$; then

$$\beta \simeq \left[F_Z \left(\frac{\mu_0 - np}{\sigma/\sqrt{n}} + \frac{\sigma_0}{\sigma} z_{\alpha/2} \right) - F_Z \left(\frac{\mu_0 - np}{\sigma/\sqrt{n}} - \frac{\sigma_0}{\sigma} z_{\alpha/2} \right) \right],$$

where $\sigma = \sqrt{np(1-p)}$, $\sigma_0 = \sqrt{np_0(1-p_0)}$, and $\mu_0 = np_0$.

‡

Now assume that we are sampling from an exponential distribution with parameter λ and we wish to test the following hypothesis:

$$H_0 : \lambda = \lambda_0 \quad \text{versus} \quad H_1 : \lambda = \lambda_1 < \lambda_0.$$

Recall that when population $X \sim \text{EXP}(\lambda)$, $2\lambda n \bar{X}$ has chi-square distribution with $2n$ degrees of freedom. Suppose that we use the critical region $\sum_{i=1}^n X_i = n\bar{X} \geq C$ for some constant C . It follows that the probability of type I error is given by

$$\begin{aligned}\alpha &= P\left(\sum_{i=1}^n X_i \geq C \mid \lambda = \lambda_0\right) \\ &= P(2\lambda n \bar{X} \geq 2\lambda C \mid \lambda = \lambda_0) \\ &= P(X_{2n}^2 \geq 2\lambda_0 C)\end{aligned}$$

and so

$$2\lambda_0 C = \chi^2_{2n;\alpha} \quad \text{or} \quad C = \frac{\chi^2_{2n;\alpha}}{2\lambda_0}.$$

Fixing α thus fixes C , and we can compute the probability of type II error by observing that

$$\begin{aligned}\beta &= P\left(\sum_{i=1}^n X_i \leq C \mid \lambda = \lambda_1\right) \\ &= P(2\lambda n \bar{X} \leq 2\lambda C \mid \lambda = \lambda_1) \\ &= P(X_{2n}^2 \leq 2\lambda_1 C).\end{aligned}$$

Example 10.34

Returning to Example 10.19, suppose that we wish to test the hypothesis:

$$H_0 : \lambda = 0.00025 \quad \text{versus} \quad H_1 : \lambda < 0.00025.$$

Assume further that we wish to attain $\alpha = 0.05$. Then

$$\begin{aligned}C &= \frac{\chi^2_{20;0.05}}{2 \cdot 0.00025} \\ &= \frac{31.41}{0.0005} = 62,820 \text{ h.}\end{aligned}$$

We therefore reject the null hypothesis if the accumulated life on test, $S_{n;r}$, is greater than the critical value 62,820 and fail to reject H_0 otherwise. The type II error probability β can be computed for any given value of λ_1 . For example, if $\lambda_1 = 0.0002$, then $2\lambda_1 C = 25.128$, and since $\chi^2_{20;2} = 25.04$, we have $\beta(0.0002) \simeq 0.8$. On the other hand, for $\lambda_1 = 0.0001$, we have $2\lambda_1 C = 12.564$, and since $\chi^2_{20;90} = 12.44$, we have $\beta(0.0001) \simeq 0.1$, which shows a dramatic reduction in the type II error!

Suppose now that we wish to reduce the value of β to 0.1 for $\lambda = 0.0002$. This can be done either by increasing the sample size (in this case, the number of observed failures) or by allowing a larger value of α :

$$C = \frac{\chi^2_{2r;\alpha}}{2\lambda_0} = \frac{\chi^2_{2r;1-\beta}}{2\lambda_1}$$

so

$$\chi^2_{2r;\alpha} = \frac{0.00025}{0.0002} \chi^2_{2r;0.9}.$$

We can now specify either α or the sample size r ; should we choose $r = 40$, we get

$$\chi^2_{2r;\alpha} = 1.25 \cdot 64.25 = 80.35.$$

Since $\chi^2_{80;0.5} = 79.33$, the value of α is approximately 0.5. Of course, we could choose smaller α at the expense of further increase in the sample size.



Problems

- To test whether a given circuit is fault-free, we drive it for a sequence of 100 inputs and observe 37 ones at the output (63 zeros). If the circuit is fault-free, 50 ones are expected. At a significance level of 0.05 (probability of a false alarm), can we reject the hypothesis that the circuit is fault-free? Compute the descriptive level of the test.
- In statistical pattern recognition, one method used to distinguish the letter B from the numeral 8 is to compute the *straightness ratio*, defined as a value of the random variable X , which is the ratio of the symbol's height to its arc length (on the left-hand side), and perform a hypothesis test on it. Suppose that the conditional distribution of X given that the symbol is 8 is normal with mean 0.8 and variance 0.01, while the conditional distribution of X given B is normal with mean 0.96 and variance 0.01. The pattern recognition problem is now cast as a hypothesis testing problem:

$$H_0 : E[X] = 0.8 \quad \text{versus} \quad H_1 : E[X] = 0.96.$$

Suppose after measurement of the given symbol we reject H_0 if $\bar{x} > 0.90$. Compute the error probabilities α and β .

- Consider the combinational circuit in problem 1 of the review problems for Chapter 1. First compute the probability of a 1 at the output, assuming that at each of the inputs the probability of a 1 is $\frac{1}{2}$. Then test the hypothesis that the circuit is fault-free versus the hypothesis that there is a stuck-at-0 type fault at input x_1 . For this case compute the probability of false alarm (α) and the probability of escape (β), assuming the length of test $n = 400$ and test stringency $\epsilon = 0.005$. Repeat the calculation of β for each of the remaining 13 fault types.
- In selecting a computer server we are considering three alternative systems. The first criterion to be met is that the response time to a simple editing command should be less than 3 s at least 70% of the time. We would like the type I error probability to be less than 0.05. On $n = 64$ randomly chosen requests the number m of requests that met the criterion of < 3 s response were found to be as shown in the following table:

<i>Server number</i>	<i>m</i>
1	52
2	47
3	32

First determine critical value C so that if $m < C$ for a server, that server will be rejected, and then determine which of the three servers will be rejected from further consideration.

- Consider the problem of acceptance sampling from a large batch of items (VLSI chips). From a sample of size n , the number of defectives found, X , is noted.

If $X \leq k$, the batch is accepted; otherwise the batch is rejected. Let p denote the actual probability of defective items in the batch. Using the Poisson approximation to the binomial, show that the probability of accepting the batch as a function of p is obtained by solving $2np = \chi^2_{2(k+1);\beta(p)}$. The plot of $\beta(p)$ against p is known as the *operating characteristic*. Plot this curve for $n = 20$ and $k = 8$. The producer of the items demands that if $p = p_0$, where p_0 is the *acceptable quality level*, then the probability α of the batch being rejected should be small. In this connection α is called the *producer's risk*. Note that $\beta(p_0) = 1 - \alpha$. The consumer demands that if the lot is relatively bad ($p \geq p_1$), the probability of its being acceptable should be small. The probability $\beta(p_1)$ is called the *consumer's risk*. Show that for fixed values of α, β, p_0 , and p_1 , the value of the critical point k is determined by solving

$$\frac{\chi^2_{2(k+1);\beta}}{\chi^2_{2(k+1);1-\alpha}} \leq \frac{p_1}{p_0}.$$

Given $p_0 = 0.05$, $p_1 = 0.10$, $\alpha = 0.05$, and $\beta(p_1) = 0.10$, determine the values of k and n . Plot the operating-characteristic curve for this case and mark the above mentioned values on the curve.

6. *The sign test.* For a continuous population distribution, develop the test for median, $\pi_{0.5}$ (based on the random sample X_1, X_2, \dots, X_n):

$$H_0 : \pi_{0.5} = m_0 \quad \text{versus} \quad H_1 : \pi_{0.5} = m_1 > m_0.$$

Let the random variable $Z_i = 0$ if $X_i - m_0 \leq 0$, and otherwise $Z_i = 1$. Let $Z = \sum Z_i$ and show that if H_0 is true, Z is binomially distributed with parameters n and 0.5. Derive an expression for the significance level α of the test based on the critical region $Z > k$. This test is known as the **sign test**, since the statistic Z is equal to the number of positive signs among

$$X_1 - m_0, X_2 - m_0, \dots, X_n - m_0.$$

10.3.2 Hypotheses Concerning Two Means

While making a purchasing decision, two vendors offer computing systems with nearly equal costs (and with all the other important attributes except throughput differing insignificantly). After running benchmarks and measuring throughputs on the two systems, we are interested in comparing the performances and finally selecting the better system. In such cases we wish to test the null hypothesis that the difference between the two population means, $\mu_X - \mu_Y$, equals some given value d_0 . We shall discuss three separate cases.

Case 1. Suppose that we wish to test the null hypothesis, $H_0 : \mu_X - \mu_Y = d_0$, for a specified constant d_0 on the basis of independent random samples of size n_1 and n_2 , assuming that the population variances σ_X^2 and σ_Y^2 are known. The test will be based on the difference of the sample means $\bar{X} - \bar{Y}$. If we

assume that both populations are normal, then the statistic

$$Z = \frac{\bar{X} - \bar{Y} - (\mu_X - \mu_Y)}{(\text{Var}[\bar{X} - \bar{Y}])^{1/2}}$$

can be shown to have the standard normal distribution. Here

$$\text{Var}[\bar{X} - \bar{Y}] = \frac{\sigma_X^2}{n_1} + \frac{\sigma_Y^2}{n_2}.$$

Now, for a significance level α , the critical regions for the test statistic Z can be specified as

$$\begin{aligned} Z < -z_\alpha, & \quad \text{if } H_1: \mu_X - \mu_Y < d_0, \\ Z > +z_\alpha, & \quad \text{if } H_1: \mu_X - \mu_Y > d_0, \\ Z < -z_{\alpha/2} \text{ or } Z > +z_{\alpha/2}, & \quad \text{if } H_1: \mu_X - \mu_Y \neq d_0. \end{aligned}$$

Example 10.35

Two file servers are compared according to their response time for retrieving a small file. The mean response time of 50 such requests submitted to server 1 was measured to be 682 ms with a known standard deviation of 25 ms. A similar measurement on server 2 resulted in a sample mean of 675 ms with a standard deviation of 28 ms. To test the hypothesis that server 2 provides better response than server 1, we form the hypotheses:

$$\begin{aligned} H_0: \mu_X &= \mu_Y \text{ (i.e., no difference in response time)} \\ H_1: \mu_X &> \mu_Y \text{ (i.e., server 2 is better than server 1).} \end{aligned}$$

Then, the null hypothesis

$$\begin{aligned} \mu_X - \mu_Y &= 0, \\ \sigma_{\bar{X}-\bar{Y}} &= \sqrt{\frac{\sigma_X^2}{n_1} + \frac{\sigma_Y^2}{n_2}} = \sqrt{\frac{(25)^2}{50} + \frac{(28)^2}{50}} = 5.3, \end{aligned}$$

and the test statistic

$$Z = \frac{\bar{X} - \bar{Y}}{\sigma_{\bar{X}-\bar{Y}}} = \frac{\bar{X} - \bar{Y}}{5.3}$$

have the following value:

$$z = \frac{682 - 675}{5.3} = 1.32.$$

Using a one-tailed test at a 5% level of significance, we fail to reject this hypothesis, since the observed value of $z (= 1.32)$ is less than the critical value $z_\alpha = z_{0.05} = 1.645$.

Thus, on the basis of the given data, we cannot support the claim that server 2 is more responsive than server 1.

Note that the null hypothesis can be rejected in this case at a 10% (rather than 5%) level of significance. This would mean that we are willing to take a 10% chance of being wrong in rejecting the null hypothesis.

‡

Case 2. The assumption that the population variances σ_X^2 and σ_Y^2 are known rarely holds in practice. If we use sample estimates in place of population variances, then we can use the t distribution in place of the normal distribution, if we further assume that the two population variances are equal ($\sigma_X^2 = \sigma_Y^2 = \sigma^2$) and that the two populations are approximately normal.

Suppose that we select two independent random samples, one from each population, of sizes n_1 and n_2 , respectively. Using the two sample variances S_X^2 and S_Y^2 , the common population variance σ^2 is estimated by S_p^2 , where

$$S_p^2 = \frac{(n_1 - 1)S_X^2 + (n_2 - 1)S_Y^2}{(n_1 + n_2 - 2)}.$$

Now, if the null hypothesis $H_0 : \mu_X - \mu_Y = d_0$ holds, then it can be shown that the statistic:

$$T = \frac{\bar{X} - \bar{Y} - d_0}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

has a t distribution with $n_1 + n_2 - 2$ degrees of freedom.

Small departures from the assumption of equal variances may be ignored if $n_1 \simeq n_2$. If σ_X^2 is much different from σ_Y^2 , a modified t statistic can be used as described by Walpole [WALP 1968].

Example 10.36

Elapsed times for a synthetic job were measured on two different computer systems. The sample sizes for the two cases were 15 each, and the sample means and sample variances were computed to be

$$\bar{x} = 104 \text{ s}, \quad \bar{y} = 114 \text{ s}, \\ s_X^2 = 290 \text{ s}^2, \quad s_Y^2 = 510 \text{ s}^2.$$

To test the hypothesis that the population means $\mu_X = \mu_Y$ against the alternative $\mu_X < \mu_Y$, we first calculate an estimate of the (assumed) common variance:

$$s_p^2 = \frac{14 \cdot 290 + 14 \cdot 510}{28} \\ = 400.$$

Now, since

$$\frac{\bar{X} - \bar{Y}}{S_p / \sqrt{7.5}}$$

is approximately t -distributed with 28 degrees of freedom, we get the descriptive level of this test:

$$\begin{aligned}\delta &= P\left(T_{28} \leq \frac{104 - 114}{\sqrt{400/7.5}}\right) \\ &= P\left(T_{28} \leq -\frac{10}{7.30}\right) = P(T_{28} \geq 1.3693) \\ &\simeq 0.0972.\end{aligned}$$

Thus the observed results have a chance of about 1 in 10 of occurring; hence we do not reject H_0 . ‡

Case 3. The test procedures discussed in the two cases above are valid only if the populations are approximately normal. We now describe a test due to Wilcoxon, Mann, and Whitney that allows for arbitrary continuous distributions for X and Y and therefore is known as a **distribution-free** or a **nonparametric** test. Specifically, we consider the problem of testing: for all x , we have

$$H_0 : f_X(x) = f_Y(x) \quad \text{versus} \quad H_1 : f_X(x) \neq f_Y(x+c), \quad (10.22)$$

where c is a positive constant (see Figure 10.15). This test is often posed as a test for the equality of the two population medians; in case of symmetric densities f_X and f_Y , it is equivalent to a test of equality of the two population means (if both exist).

Assume that two independent random samples of respective sizes n_1 and n_2 are collected from the two populations and denoted by $x_1, x_2, x_3, \dots, x_{n_1}$ and y_1, y_2, \dots, y_{n_2} . We now combine the two samples, arrange these values in order of increasing magnitude, and assign to the $(n_1 + n_2)$ -ordered values

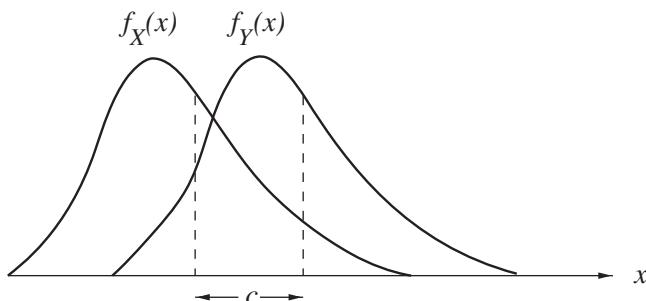


Figure 10.15. A nonparametric test

the ranks $1, 2, 3, \dots, n_1 + n_2$. In the case of ties, assign the average of the ranks associated with the tied values. Let $r(Y_i)$ denote the rank of Y_i in the combined ordered set and define the statistic

$$W = \sum_{i=1}^{n_2} r(Y_i).$$

The statistic W is a sum of ranks, hence the test we describe is commonly known as the **rank-sum test**.

Under H_1 , the density of Y is shifted to the right of the density of X and the values in the Y sample would tend to be larger than the values in the X sample. Thus, under H_1 , the statistic W would tend to be larger than expected under H_0 . Therefore, the critical region for the test (10.22) will be of the form $W > w_0$.

We can derive the distribution of W under H_0 by noting that the combined ordered set represents a random sample of size $n_1 + n_2$ from the population of X . Further, since the ranks depend on the relative (and not the absolute) magnitudes of the sample values, it is sufficient to consider the positions of the Y values in the combined set in order to evaluate $F_W(w)$. Let $\#(W = w)$ denote the set of all combinations of y ranks that will sum to w . The total ways of picking all combinations of ranks given n_1 and n_2 is

$$\binom{n_1 + n_2}{n_2},$$

and since each of these combinations is equally likely under H_0 , we get (assuming no ties)

$$P(W = w | H_0) = \frac{\#(W = w)n_1!n_2!}{(n_1 + n_2)!}.$$

Then the significance level α is determined by

$$P(W \geq w | H_0) \leq \alpha.$$

Since the distribution function $F_w(w)$ depends only on relative ranks, it can be computed by combinatorial methods. For small values of n_1 and n_2 , the w_α values have been precomputed and listed in Appendix C. For larger values of n_1 and n_2 , we use a normal approximation to the distribution of W . It can be shown that if H_0 is true, and $n_1 \geq 10$ and $n_2 \geq 10$, the statistic W possesses an approximate normal distribution with

$$E[W] = \frac{n_1(n_1 + n_2 + 1)}{2}$$

and

$$\text{Var}[W] = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12}.$$

Example 10.37

The times between two successive crashes are recorded for two competing computer systems as follows (time in weeks):

$$\begin{array}{ll} \text{System } X : & 1.8 \quad 0.4 \quad 2.7 \quad 3.0 \\ \text{System } Y : & 2.0 \quad 5.4 \quad 1.3 \quad 4.5 \quad 0.8 \end{array}$$

In order to test the hypothesis that both systems have the same mean time between crashes against the alternative that system X has a shorter mean time between crashes, we first arrange the combined data in ascending order and assign ranks (y ranks are underlined for easy identification):

Original data	0.4	0.8	1.3	1.8	2.0	2.7	3.0	4.5	5.4
Ranks	1	<u>2</u>	<u>3</u>	4	<u>5</u>	6	7	<u>8</u>	<u>9</u>

The rank sum $W = 2 + 3 + 5 + 8 + 9 = 27$. Looking up the table of rank-sum critical values with $n_1 = 4$ and $n_2 = 5$, we find that

$$P(W \geq 27 | H_0) = 0.056.$$

Therefore, we reject the null hypothesis at 0.056 level of significance.

‡

Noether [NOET 1967] points out several reasons why in general the rank-sum test is preferable to the t test. Since the rank-sum test is distribution free, whatever the true population distribution, as long as both samples come from the same population (i.e., $f_X = f_Y$), the significance level of the test is known. On the other hand, for nonnormal populations, the significance level of the t test may differ considerably from the calculated value. In contrast to the t test, the rank-sum test is not overly affected by large deviations (so called **outliers**). On the other hand, when there is sufficient justification for assuming that the population distribution is normal, it would be a mistake not to use that information [HOEL 1971].

Problems

1. Returning to the server-selection problem considered in problem 4 of Section 10.3.1, the sample means of the response times for the first two servers are 2.28 and 2.52 s, respectively. The sample size in both cases is 64, and the variances can be assumed to be 0.6 and 0.8 s^2 , respectively. Test the hypothesis that the mean response times of the two servers are the same against the alternative that server 1 has a smaller mean response time.
2. In this section we assumed that X and Y samples were chosen independently. In practice, the observations often occur in pairs, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. On

the basis of pairwise differences $d_i = x_i - y_i$, construct a t test using the statistic

$$T = \frac{\bar{D} - d_0}{S_D/\sqrt{n}}$$

to test the hypothesis $H_0 : \mu_X - \mu_Y = d_0$. Also show that in case of nonnormal populations, the sign test of problem 6 in Section 10.3.1, can be adapted to test the null hypothesis $H_0 : \mu_X = \mu_Y$. Apply these two tests to the claim that two computer systems are about equal in their processing speeds, based on the following data:

<i>Benchmark program</i>	<i>Run time in seconds</i>	
	<i>Server A</i>	<i>Server B</i>
Payroll	42	55
Linear programming (simplex)	201	195
Least squares	192	204
Queueing network solver	52	40
Puzzle	10	12
Simulation	305	290
Statistical test	10	13
Synthetic 1	1	1
Synthetic 2	350	320
Synthetic 3	59	65

10.3.3 Hypotheses Concerning Variances

First consider the problem of testing the null hypothesis that a population variance σ^2 equals some fixed value σ_0^2 against a suitable one-sided or two-sided alternative. For instance, many computer users may not mind a relatively long average response time as long as the system is consistent. In other words, they wish the variance of the response time to be small.

Assuming that we are sampling from a normal population $N(\mu, \sigma_0^2)$, we have shown in Chapter 3 that the statistic

$$X_{n-1}^2 = \frac{(n-1)S^2}{\sigma_0^2}$$

is chi-square distributed with $n - 1$ degrees of freedom. From this, the critical regions for testing $H_0 : \sigma^2 = \sigma_0^2$ are

<i>Reject H_0 if, for</i>	
$X_{n-1}^2 = (n-1)S^2/\sigma_0^2$	<i>when H_1 is</i>
$X_{n-1}^2 < \chi_{n-1;1-\alpha}^2$	$\sigma^2 < \sigma_0^2$
$X_{n-1}^2 > \chi_{n-1;\alpha}^2$	$\sigma^2 > \sigma_0^2$
$X_{n-1}^2 < \chi_{n-1;\alpha/2}^2$ or $X_{n-1}^2 < \chi_{n-1;1-\alpha/2}^2$	$\sigma^2 \neq \sigma_0^2$

Example 10.38

In the past, the standard deviation of the response time to simple commands to a compute server was 25 ms and the mean response time was 400 ms. A new version of the operating system was installed, and it was claimed to be biased against simple commands. With the new system, a random sample of 21 simple commands experienced a standard deviation of response times of 32 ms. Is this increase in variability significant at a 5% level of significance? Is it significant at a 1% level? Suppose that we wish to test

$$H_0 : \sigma^2 = (25)^2 \quad \text{versus} \quad \sigma^2 > (25)^2.$$

The observed value of the chi-square statistic is

$$\frac{(n-1)s^2}{\sigma_0^2} = \frac{20(32)^2}{(25)^2} = 32.8.$$

Since $\chi_{20;0.05}^2 = 31.41$, we conclude at the 5% level of significance that the new version of the system is unfair to simple commands. On the other hand, since $\chi_{20;0.01}^2 = 37.566$, we cannot reject H_0 at the 1% level of significance. ‡

We should caution the reader that the test described above is known to give poor results if the population distribution deviates appreciably from the normal distribution. The reader is advised to use a suitable nonparametric test in such cases [NOET 1967].

If we wish to compare the variances of two normal populations, with variances σ_X^2 and σ_Y^2 respectively, then we test the following hypothesis:

$$H_0 : \sigma_X^2 = \sigma_Y^2.$$

In this case we apply the fact that the ratio $(S_X^2 \sigma_Y^2)/(S_Y^2 \sigma_X^2)$ has an F distribution with $(n_1 - 1, n_2 - 1)$ degrees of freedom. This statistic is simply the ratio of sample variances if H_0 is true. Here the sample size for the first population is n_1 , while the sample size for the second population is n_2 . Therefore, the critical region for testing H_0 against the alternative $H_1 : \sigma_X^2 > \sigma_Y^2$ is

that the observed value of the F statistic satisfies $F_{n_1-1, n_2-1} > f_{n_1-1, n_2-1; \alpha}$, where $f_{n_1-1, n_2-1; \alpha}$ is determined by the F distribution to be that value such that $P(F_{n_1-1, n_2-1} > f_{n_1-1, n_2-1; \alpha}) = \alpha$. Similarly, if $H_1 : \sigma_X^2 < \sigma_Y^2$, we use the critical region $F_{n_1-1, n_2-1} < f_{n_1-1, n_2-1; 1-\alpha}$.

Example 10.39

In comparing two compute servers based on sample means, suppose that at the desired level of significance the hypothesis $\mu_X = \mu_Y$ cannot be rejected; in other words, the difference in the average response times is not statistically significant. The next level of comparison is then the difference in the variances of the response times. Recalling Example 10.27, we have $s_X^2 = (25)^2 = 625$ and $s_Y^2 = (28)^2 = 784$. The observed value of the F statistic under H_0 is $\frac{625}{784} = 0.797$. If we wish to test H_0 against $H_1 : \sigma_X^2 < \sigma_Y^2$, we need to obtain the critical value of $f_{49, 49; 1-\alpha}$, so we use a table of the F distribution with (49, 49) degrees of freedom. Then $f_{49, 49; \alpha} = f_{49, 49; 0.05} = 1.62$, which we invert to obtain $f_{49, 49; 1-\alpha} = 0.617$. Since the observed value is larger than the critical value, we fail to reject H_0 . In other words, server 1 does not provide a statistically lower variability in response time at the 5% level of significance.

#

Problems

1. Returning to the data in Example 10.18, test the hypothesis at significance level 0.05 that the variance of response time is 20 ms^2 against the alternative that the variance is greater than 20 ms^2 .

10.3.4 Goodness-of-fit Tests

Most of the methods in the preceding sections require the type of the distribution function of X to be known and either its parameters to be estimated or a hypothesis concerning its parameters to be tested. It is important to have some type of test that can establish the “goodness of fit” between the postulated distribution type of X and the evidence contained in the experimental observations. Such experimental data are likely to be in the same basic form as the data used to estimate parameters of the distribution. Graphical methods are also used to establish goodness of fit as well as analytical methods.

First assume that X is a discrete random variable with true (but unknown) pmf given by $p_X(i) = p_i$. We wish to test the null hypothesis that X possesses a certain specific pmf given by $p_i = p_{i_0}$, $0 \leq i \leq k - 1$. Our problem then is to test H_0 versus H_1 , where

$$\begin{aligned} H_0 : \quad p_i &= p_{i_0}, \quad i = 0, 1, \dots, k - 1, \\ H_1 : \quad &\text{not } H_0. \end{aligned}$$

Assume that we make n observations and let N_i be the observed number of times (out of n) that the measured value of X takes the value i . N_i is clearly a

binomial random variable with parameters n and p_i so that $E[N_i] = np_i$ and $\text{Var}[N_i] = np_i(1 - p_i)$. Wilks [WILK 1962] shows that the statistic

$$Q = \sum_{i=0}^{k-1} \frac{(N_i - np_i)^2}{np_i} \quad (10.23)$$

is approximately chi-square distributed with $(k - 1)$ degrees of freedom. One degree of freedom is lost because only $k - 1$ of the N_i are independent owing to the relation

$$n = \sum_{i=0}^{k-1} N_i = \sum_{i=0}^{k-1} np_i.$$

Under the assumption $H_0 : p_i = p_{i_0}$, the statistic (10.23) is just

$$X_{k-1}^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}.$$

Example 10.40

A large enterprise server has six I/O channels and the system personnel are reasonably certain that the load on the channels is balanced. If X is the random variable denoting the index of the channel to which a given I/O operation is directed, then its pmf is assumed to be

$$p_X(i) = p_i = \frac{1}{6}, \quad i = 0, 1, \dots, 5.$$

Out of $n = 150$ I/O operations observed, the numbers of operations directed to various channels were

$$n_0 = 22, \quad n_1 = 23, \quad n_2 = 29, \quad n_3 = 31, \quad n_4 = 26, \quad n_5 = 19.$$

We wish to test the hypothesis that the load on the channels is balanced; that is, $H_0 : p_i = \frac{1}{6}, i = 0, 1, \dots, 5$. Using the chi-square statistic, we obtain

$$\begin{aligned} \chi^2 &= \frac{(22 - 25)^2}{25} + \frac{(23 - 25)^2}{25} + \frac{(29 - 25)^2}{25} \\ &\quad + \frac{(31 - 25)^2}{25} + \frac{(26 - 25)^2}{25} + \frac{(19 - 25)^2}{25} \\ &= 4.08. \end{aligned}$$

For the chi-square distribution with 5 degrees of freedom, the 55% critical value is

$$\chi_{5,0.55}^2 \simeq 4.00.$$

In other words, there is a high probability under H_0 of observing such a small deviation; hence we cannot reject the null hypothesis that the channels are load-balanced.

#

In deriving the distribution of test statistic (10.23), the multivariate normal approximation to the multinomial distribution is employed [WILK 1962]. For the approximation to be accurate, each np_i value should be moderately large (as a rule of thumb, $np_i \geq 5$). When the random variable X takes a large (perhaps infinite) number of values, the condition $np_i \geq 5$ for all i will be difficult to meet even with a large value of n . If the expected numbers in several categories are small, then these categories should be combined to form a single category. Note that this process of combination of categories implies a concomitant loss in power of the test [WILK 1962].

While performing a goodness-of-fit test, often a null hypothesis specifies only that the population distribution belongs to a family of distributions $F_X(x; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a vector of unknown parameters. For example, we may want to test whether X has a Poisson distribution. This amounts to a null hypothesis:

$$H_0 : p_i = \frac{\lambda^i e^{-\lambda}}{i!}, \quad i = 0, 1, \dots,$$

which we cannot test without some specification of λ . In such situations, the unknown parameters of the population (such as λ in the Poisson example) must first be estimated from the collected sample of size n . We then use a test statistic:

$$\hat{Q} = \sum_{i=0}^{k-1} \frac{(N_i - n\hat{p}_i)^2}{n\hat{p}_i},$$

where $\hat{p}_i = p_x(i; \hat{\boldsymbol{\theta}})$ is obtained using the maximum-likelihood estimates of the parameters $\boldsymbol{\theta}$. It can be shown that the statistic \hat{Q} is approximately chi-square distributed with $k - m - 1$ degrees of freedom. Thus, if m population parameters are to be estimated, the chi-square statistic loses m degrees of freedom.

Example 10.41

It is suspected that the number of errors discovered in a system program is Poisson distributed. The number of errors discovered in each one-week period is given in Table 10.6. The total number of errors observed in the 50 weeks was 95.

To compute the Poisson probabilities above, we must have an estimate of the rate parameter λ , which is computed from the data

$$\hat{\lambda} = \frac{\text{total number of observed errors}}{\text{total number of weeks observed}} = \frac{95}{50} = 1.9 \text{ per week}$$

$$\chi^2 = \sum \frac{(N_i - n\hat{p}_i)^2}{n\hat{p}_i}$$

$$\begin{aligned}
&= \frac{(14 - 7.50)^2}{7.50} + \frac{(11 - 14.20)^2}{14.20} + \frac{(9 - 13.5)^2}{13.5} + \frac{(6 - 8.55)^2}{8.55} \\
&\quad + \frac{(5 - 4.05)^2}{4.05} + \frac{(5 - 2.20)^2}{2.20} \\
&= 5.633 + 0.7211 + 1.5 + 0.7605 + 0.2228 + 3.564 \\
&= 12.401
\end{aligned}$$

is the value of a chi-square random variable with $k - 2 = 6 - 2 = 4$ degrees of freedom. Since $\chi^2_{4,0.05} = 9.488$ is lower than the observed value of 12.401, we conclude that the null hypothesis of Poisson distribution should be rejected at a 5% level of significance. The descriptive level for this test is approximately 0.016, indicating that the observations would be highly unlikely to occur under the Poisson assumption.

‡

Now suppose that X is a continuous random variable and we wish to test the hypothesis that the distribution function of X is a specified function:

$$H_0 : \text{ for all } x \ F_X(x) = F_0(x),$$

versus

$$H_1 : \text{ there exist } x \text{ such that } F_X(x) \neq F_0(x).$$

The chi-square test described above is applicable in this case, but we will be required to divide the image of X into a finite number of categories. The subsequent loss of information results in a loss of power of the test [WILK 1962]. The Kolmogorov–Smirnov test to be described is the preferred goodness-of-fit test in case of a continuous population distribution. Conversely, when applied to discrete population distributions, the Kolmogorov–Smirnov test is known to produce conservative results. Thus, the actual probability of type I error

TABLE 10.6. Fitting a Poisson model

Number of errors (i) in one-week period	Number of one-week periods with i errors	Poisson probabilities \hat{p}_{i_0}	Expected frequencies $n\hat{p}_{i_0}$
0	14	0.150	7.50
1	11	0.284	14.20
2	9	0.270	13.50
3	6	0.171	8.55
4	5	0.081	4.05
5+	5	0.044	2.20

will be at most equal to the chosen value α but this advantage is offset by a corresponding loss of power (or increase in the probability of type II error).

The given random sample is first arranged in order of magnitude so that the values are assumed to satisfy $x_1 \leq x_2 \leq \cdots \leq x_n$. Then the empirical distribution functions $\hat{F}_n(x)$ is defined by

$$\hat{F}_n(x) = \begin{cases} 0, & x < x_1 \\ \frac{i}{n}, & x_i \leq x < x_{i+1} \\ 1, & x_n \leq x. \end{cases} \quad (10.24)$$

The alternative definition of $\hat{F}_n(x)$ is

$$\hat{F}_n(x) = \frac{\text{number of values in the sample that are } \leq x}{n}.$$

A logical measure of deviation of the empirical distribution function from $F_0(x)$ is the absolute value of the following difference:

$$d_n(x) = |\hat{F}_n(x) - F_0(x)|.$$

Since $F_0(x)$ is known, the deviation $d_n(x)$ can be computed for each value of x . The largest among these values as x varies over its full range is an indicator of how well $\hat{F}_n(x)$ approximates $F_0(x)$. Since $\hat{F}_n(x)$ is a step function (see Figure 10.16) with n steps and $F_0(n)$ is continuous and nondecreasing, it suffices to evaluate $d_n(x)$ at the left and right endpoints of the intervals $[x_i, x_{i+1}]$. The maximum value of $d_n(x)$ is then the value of the Kolmogorov–Smirnov statistic defined by

$$D_n = \sup_x |\hat{F}_n(x) - F_0(x)|. \quad (10.25)$$

[The reason for the use of supremum rather than maximum in the preceding definition is that by definition (10.24), $\hat{F}_n(x)$ is a discontinuous function.] The definition (10.25) simply says that we evaluate $|\hat{F}_n(x) - F_0(x)|$ at the endpoints of each interval $[x_i, x_{i+1}]$, treating $\hat{F}_n(x)$ as having a constant value in that interval, and then choosing the largest of these values as the value of D_n .

The usefulness of the statistic (10.25) is that it is **distribution-free**; hence its exact distribution can be derived. In other words, for a continuous $F_0(x)$, the sampling distribution of D_n depends only on n and not on $F_0(x)$. Thus the D_n statistic possesses the advantage that its exact distribution is known even for small n whereas the Q statistic is only approximately chi-square distributed, and a fairly large sample size is needed in order to justify the approximation. We shall not derive the distribution function of D_n , but we give a table of critical values $d_{n;\alpha}$ in Appendix C.

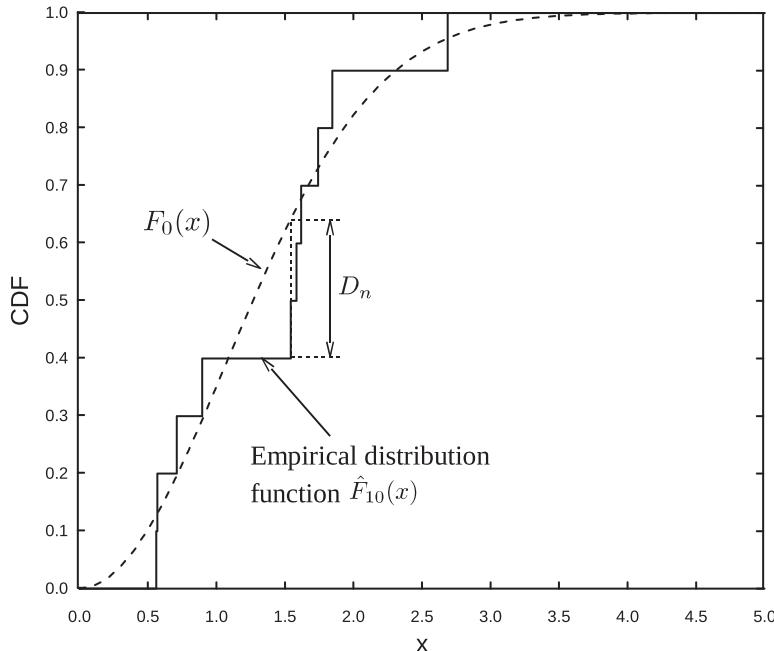


Figure 10.16. The Kolmogorov–Smirnov test for a small sample size

We reject the null hypothesis at a level of significance α if the observed value of the statistic D_n exceeds the critical value $d_{n;\alpha}$; otherwise we reject the alternative hypothesis H_1 .

Example 10.42

In Example 3.12 we used the inverse transform method to generate a random deviate with Weibull distribution. Suppose that we generate 10 Weibull distributed random deviates with shape parameter 2 and $\lambda = 0.4325$:

$$\begin{array}{cccccccc} 1.8453 & 0.5616 & 1.6178 & 2.6884 & 1.7416 & 0.7111 & 1.5430 \\ & 1.5831 & 0.5688 & 0.8961. \end{array}$$

We wish to test the hypothesis that the population distribution function is

$$F_X(x) = 1 - e^{-\lambda x^2}, \quad (10.26)$$

with $\lambda = 0.4325$. $F_{10}(x)$ and $F_0(x)$ are plotted in Figure 10.16. The observed value of the D_n statistic is 0.2429. Now, using the table of critical values in Appendix C, we find that at $\alpha = 0.05$, $d_{10;\alpha} = 0.41$, and hence the rejection region is $\{D_{10} > 0.41\}$. We therefore accept the null hypothesis at the 5% level of significance.

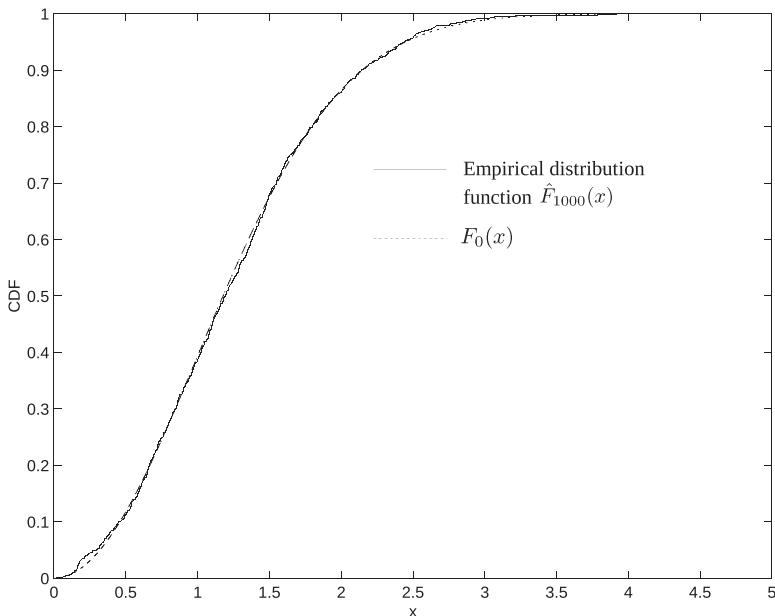


Figure 10.17. The Kolmogorov–Smirnov test for a large sample size

The empirical distribution function does not match very well with the theoretical distribution function. This is because too few points are used to plot the empirical distribution function. Figure 10.17 shows the result for 1000 sample points. We see that the two distribution functions match each other much better. The observed value for D_n statistic is 0.0273. Using the table of critical values in Appendix C, at $\alpha = 0.5$, $d_{1000;\alpha} = 0.043$. Therefore, the null hypothesis is accepted at the 5% level of significance.

#

Now consider the problem of obtaining a confidence interval for the unknown function $F_0(x)$. For a fixed value of x , $n\hat{F}_n(x)$ is a binomial random variable with parameters n and $p = F_0(x)$. Hence we can use the procedure of Section 10.2.3.4. Here we are considering a confidence interval not just for $F_0(x)$ at a number of isolated points but for $F_0(x)$ as a whole. A **confidence band** with confidence coefficient γ for $F_0(x)$ is obtained using the D_n statistic by using

$$\begin{aligned}\gamma &= P(D_n \leq d_{n;1-\gamma}) \\ &= P(\sup_x |\hat{F}_n(x) - F_0(x)| \leq d_{n;1-\gamma}) \\ &= P(|\hat{F}_n(x) - F_0(x)| \leq d_{n;1-\gamma} \text{ for all } x) \\ &= P(\hat{F}_n(x) - d_{n;1-\gamma} \leq F_0(x) \leq \hat{F}_n(x) + d_{n;1-\gamma} \text{ for all } x).\end{aligned}$$

Noting that $0 \leq F_0(x) \leq 1$, we have a confidence band for $F_0(x)$ with confidence coefficient γ as follows; for all x ,

$$\max\{0, \hat{F}_n(x) - d_{n;1-\gamma}\} \leq F_0(x) \leq \min\{1, \hat{F}_n(x) + d_{n;1-\gamma}\}. \quad (10.27)$$

Suppose that the null hypothesis does not specify the function $F_0(x)$ completely but specifies some parametric family of functions $F_0(x; \boldsymbol{\theta})$ where parameters $\boldsymbol{\theta}$ are to be estimated from the given sample. Analogous to the chi-square test, we will use the test statistic

$$\hat{D}_n = \sup_x |\hat{F}_n(x) - F_0(x, \hat{\boldsymbol{\theta}})|,$$

where $\hat{\boldsymbol{\theta}}$ is an appropriate estimate of unknown vector of parameters $\boldsymbol{\theta}$. Unfortunately, there is no simple modification as in the case of the chi-square test. The sampling distribution of \hat{D}_n must be separately studied for each family of population distribution functions. Lilliefors has studied the distribution of \hat{D}_n in case $F_0(x; \theta)$ is the family of exponential distributions with unknown mean θ [LILL 1969] and in case $F_0(x; \mu, \sigma^2)$ is the family of normal distributions with unknown mean μ and unknown variance σ^2 [LILL 1967]. In Appendix C we give tables of critical values for the statistic \hat{D}_n in these two cases.

Having studied two analytic methods of goodness of fit, now we consider a graphical method. The probability plot is one of the methods used to graphically analyze reliability data. It is based simply on the concept of transforming the data in such a way that approximately straight lines can be generated when the data is plotted. Then, the graph can be quickly checked to determine whether a straight line can reasonably fit the data. If not, the assumed distribution is rejected and another may be tried.

We will first do the probability plot for the exponential distribution. Then the technique will be extended to the Weibull distribution, following a similar procedure.

The CDF of the exponential distribution is

$$F(t) = 1 - e^{-\lambda t}.$$

Rewriting this equation and taking natural logarithms, we get

$$\begin{aligned} 1 - F(t) &= e^{-\lambda t} \\ \ln\left[\frac{1}{1 - F(t)}\right] &= \lambda t. \end{aligned}$$

This is a linear equation in time variable t . If the exponential distribution applies, the data plotted should approximately fall on a straight line.

Now we need to estimate the CDF $F(t)$ from the data of failure time t . A simple approach would be to use the empirical CDF:

$$\hat{F}(t_i) = \frac{i}{n} \quad i = 1, 2, 3, \dots, n$$

This is a good approximation of the true CDF when n is large. However, when we have only one data point ($n = 1$), (e.g., we have only one unit and it fails at time t_1), we do not expect the observed time to failure to represent the 100th percentile [i.e., $F(t_1) = 1$] of the population distribution. So we can use the following alternative definition of the empirical CDF:

$$\hat{F}(t_i) = \frac{i - 0.3}{n + 0.4} \quad i = 1, 2, 3, \dots, n$$

Next we look at the CDF of the Weibull distribution,

$$F(t) = 1 - e^{-\lambda t^\alpha}.$$

We rewrite it and take natural logarithms twice:

$$1 - F(t) = e^{-\lambda t^\alpha},$$

$$\ln\left[\frac{1}{1 - F(t)}\right] = \lambda t^\alpha,$$

$$\ln\{\ln\left[\frac{1}{1 - F(t)}\right]\} = \alpha \ln t + \ln \lambda.$$

This equation is linear in $\ln t$. If the Weibull distribution applies, the data plotted should approximately fall on a straight line with slope α and intercept $\ln \lambda$.

Consider the data used in Example 10.22. The probability plot of exponential distribution is shown in Figure 10.18. Obviously it is far from a straight line. So we want to try the probability plot of the Weibull distribution, which results in Figure 10.19. It is shown that a straight line can be approximately fitted into the data plot, which indicates that the Weibull distribution assumption is appropriate for the data set. The line we plotted in the graph has a slope 1.924 and intercept -13.5. We can easily compute the parameters α and λ from the slope and the intercept. From this graph, we can also see the exponential distribution (the straight line with slope equal to 1) is inappropriate for this data set.

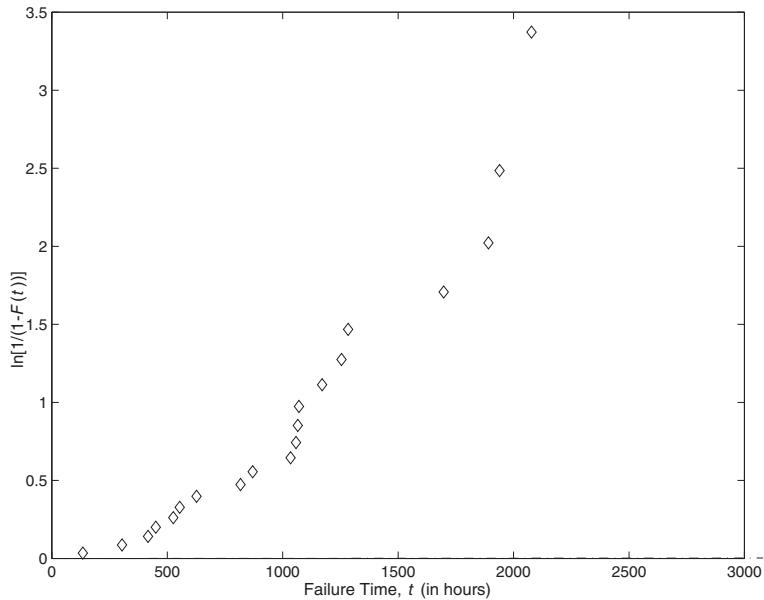


Figure 10.18. Probability plot with exponential distribution assumption

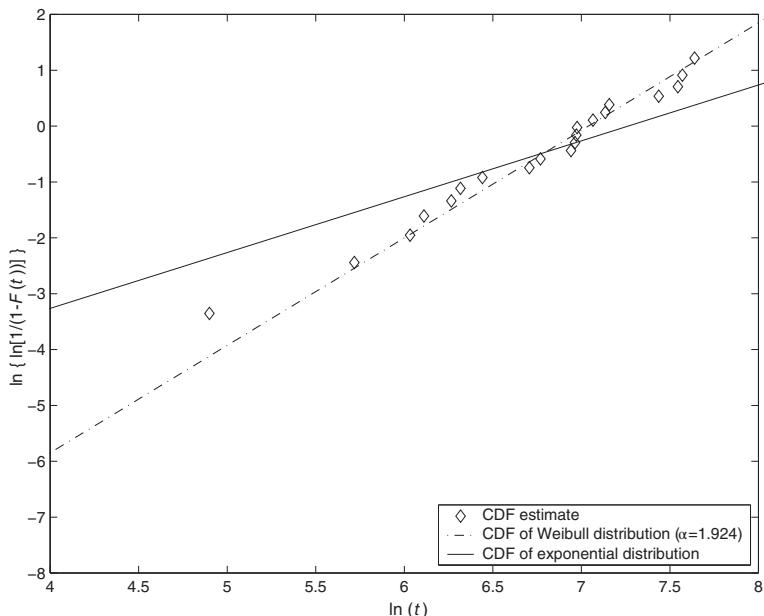


Figure 10.19. Probability plot with Weibull distribution assumption

Problems

1. The number of busy senders in a panel-type switching machine of a telephone exchange was observed as follows:

<i>Number busy</i>	<i>Observed frequency</i>	<i>Number busy</i>	<i>Observed frequency</i>
0	0	12	413
1	5	13	358
2	14	14	219
3	24	15	145
4	57	16	109
5	111	17	57
6	197	18	43
7	278	19	16
8	378	20	7
9	418	21	8
10	461	22	3
11	433		

Test whether the corresponding theoretical distribution is Poissonian.

2. Perform a goodness of fit test at significance level 0.05 for the binomial model on the data of Example 2.4, assuming that the population parameter p is known to be 0.1.
3. Return to the question of memory residence times considered in problem 3 in Section 10.2.1. The empirical distribution function for the memory residence times was measured and is shown below:

<i>Time in milliseconds</i>	<i>Count</i>	<i>Cumulative percent</i>
0–31	7540	57.24
32–63	2938	79.55
64–95	1088	87.81
96–127	495	91.57
128–191	449	94.98
192–319	480	98.62
319–1727	181	100.00

Graphically compare the empirical distribution with three theoretical distributions, normal, gamma, and hyperexponential (for the last two distributions, parameters were estimated in problem 3 in Section 10.2.1, whereas for the normal distribution, parameters μ and σ^2 are readily estimated). Now perform chi-square goodness-of-fit tests against the three theoretical distributions at the 5% level of significance.

4. Since the Poisson model was found to be improper for the data in Example 10.41, try the (modified) geometric model:

$$H_0 : p_i = p(1 - p)^i, \quad i = 0, 1, \dots$$

Note that the unknown parameter p must be estimated before the chi-square test can be performed.

5. Using formula (10.27), construct a 90% confidence band for the distribution function $F_0(x)$ based on the data in Example 10.42. Plot your results together with the distribution of F_0 under the null hypothesis.
6. Observed times between successive crashes of a computer system were noted for a 6-month period as follows (time in hours):

1, 10, 20, 30, 40, 52, 63, 70, 80, 90, 100, 102, 130, 140, 190,

210, 266, 310, 530, 590, 640, 1340

Using the \hat{D}_n statistic, test a goodness of fit against an exponential model and a normal model for the population distribution.

7. We wish to verify the analytical results of review problems 2 and 3 at the end of Chapter 3 by means of a discrete-event simulation. Suppose that the mantissas X and Y of two floating-point numbers are independent random variables with uniform density [over $[1/\beta, 1]$] and reciprocal pdf $1/(y \ln \beta)$, respectively. Generate n random deviates of X and Y [for the random deviate of Y , use the formula derived in problem 6(a), Section 3.5], and compute the mantissas of the normalized product Z_N and the normalized quotient Q_N . From these values, obtain the empirical distributions of Z_N and Q_N . Now perform goodness-of-fit tests against the reciprocal distribution, using the Kolmogorov–Smirnov D_n statistic. Use $n = 10, 20, 30$ and $\beta = 2, 10, 16$.

REFERENCES

- [BHAT 1984] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd ed., Wiley, New York, 1984.
- [CINL 1975] E. Cinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [CONS 1995] C. Constantinescu, “Using multi-stage & stratified sampling for inferring fault coverage probabilities,” *IEEE Trans. Reliability* **44**(4), 632–639 (1995).
- [CONS 1999] C. Constantinescu, “Using physical and simulated fault injection to evaluate error detection mechanisms,” *Proc. Pacific Rim Int. Symp. Dependable Computing*, Hong Kong, Dec. 1999, pp. 186–192.
- [DEVA 1989] M. V. Devarakonda and R. K. Iyer, “Predictability of process resource usage: A measurement-based study on UNIX,” *IEEE Trans. Software Eng.* **15**(12), 1579–1586 (Dec. 1989).

- [FERR 1983] D. Ferrari, G. Serazzi, and A. Zeigner, *Measurement and Tuning of Computer Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [FISH 1978] G. S. Fishman, *Principles of Discrete Event Digital Simulation*, Wiley, New York, 1978.
- [GAY 1978] F. A. Gay, "Evaluation of maintenance software in real-time systems," *IEEE Trans. Comput.* **27**(6), 576–582 (June 1978).
- [GOEL 1979] A. L. Goel and K. Okumoto, "Time-dependant error-detection rate models for software reliability and other performance measures," *IEEE Trans. Reliability* **28**(3), (206–211) (Aug. 1979).
- [HART 1975] J. A. Hartigan, *Clustering Algorithms*, Wiley, New York, 1975.
- [HOEL 1971] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Statistical Theory*, Houghton Mifflin, Boston, 1971.
- [KEND 1961] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 2, *Inference and Relationship*, Hafner, New York, 1961.
- [KLEI 1992] J. P. C. Kleijnen and W. Van Groenendaal, *Simulation: a Statistical Perspective*, Wiley, New York, 1992.
- [LEEM 1995] L. M. Leemis, *Reliability—Probabilistic Models and Statistical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [LEEM 1996] L. M. Leemis and K. S. Trivedi, "A comparison of approximate interval estimators for the Bernoulli parameter," *Am. Stats.* **50**(1), 63–68 (Feb. 1996).
- [LILL 1967] H. W. Lilliefors, "On the Kolmogorov–Smirnov test for normality with mean and variance unknown," *J. Am. Stat. Assoc.* **62**, 399–402 (1967).
- [LILL 1969] H. W. Lilliefors, "On the Kolmogorov–Smirnov test for the exponential distribution with mean unknown," *J. Am. Stat. Assoc.* **64**, 387–389 (1969).
- [NOET 1967] G. E. Noether, *Elements of Nonparametric Statistics*, Wiley, New York, 1967.
- [OHBA 1984] M. Ohba, "Software reliability analysis models," *IBM J. Res. Develop.* **28**(4), 428–442 (July 1984).
- [POWE 1995] D. Powell, E. Martins, J. Arlat, and Y. Crouzet, "Estimators for fault tolerance coverage evaluation," *IEEE Trans. Comput.* 261–274 (Feb. 1995).
- [ROSS 1983] S. M. Ross, *Stochastic Processes*, Wiley, New York, 1983.
- [VAID 1999] K. Vaidyanathan and K. S. Trivedi, "A measurement-based model for estimation of resource exhaustion in operational software systems," *Proc. 10th Int. Symp. Software Reliability Engineering*, Boca Raton, FL, Nov. 1999.
- [WALP 1968] R. E. Walpole, *Introduction to Statistics*, Macmillan, New York, 1968.
- [WILK 1962] S. S. Wilks, *Mathematical Statistics*, Wiley, New York, 1962.
- [WOLF 1999] S. Wolfram, *The Mathematica Book*, 4th ed., Wolfram Media/Cambridge Univ. Press, 1999

Chapter 11

Regression and Analysis of Variance

11.1 INTRODUCTION

In this chapter, we study aspects of **statistical relationships** between two or more random variables. For example, in a computer system the throughput Y and the degree of multiprogramming X might well be related to each other. One indicator of the association (interdependence) between two random variables is their correlation coefficient $\rho(X, Y)$ and its estimator $\hat{\rho}(X, Y)$. Correlation analysis will be considered in Section 11.6.

A related problem is that of predicting a value of system throughput y at a given degree of multiprogramming x . In other words, we are interested here in studying the dependence of Y on X . The problem then is to find a **regression line** or a **regression curve** that describes the dependence of Y on X . Conversely, we may also study the inverse regression problem of dependence of X on Y . In the remainder of this section we consider regression when the needed parameters of the population distribution are known exactly. Commonly, though, we are required to obtain a regression curve that best approximates the dependence on the basis of sampled information. This topic will be covered in Sections 11.3 and 11.4.

Another related problem is that of **least-squares curve fitting**. Suppose that we have two variables (not necessarily random) and we hypothesize a relationship (e.g., linear) between the two variables. From a collection of n pairs of measurements of the two variables, we wish to determine the equation of a line (curve, in general) of closest fit to the data. Out of the many possible criteria in choosing the “best” fit, the criterion leading to the simplest calculations is the least-squares criterion, which will be discussed in the next section.

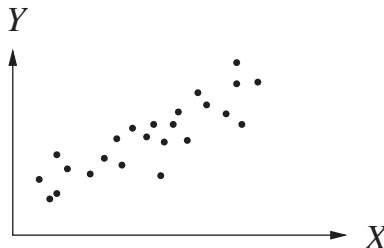


Figure 11.1. A scatter diagram showing a linear dependence

Consider two random variables, X and Y , possessing a joint density, $f(x, y)$. We would like to design a function, $d(x)$, so that the random variable, $d(X)$, will be as close as possible to Y in an appropriate sense. When $d(x)$ is used to predict a value of Y , a modeling error will usually result so that the actual value $y = d(x) + \epsilon$.

Such errors are introduced, of course, because we are attempting to simplify the joint distribution of X and Y by postulating an elementary functional dependence, d , of Y on X . The extent of our error is thus the extent to which the random variable Y differs from the random variable $d(X)$, the most common measure of which is undoubtedly the expected value of the squared difference, $E[D^2]$, where $D = Y - d(X)$.

The function $d(x)$ for which $E[D^2]$ is at a minimum is commonly called the **least-squares regression curve** of Y on X . Now it is easy to show that this regression curve is necessarily given by $d(x) = E[Y|x]$; nevertheless, conditional distributions are, in practice, difficult to obtain, so most often we simply restrict our choices for $d(x)$ to a specific class of functions and minimize $E[D^2]$ over that class. In a similar fashion $g(y) = E[X|y]$ will give us the dependence of X on Y .

A common choice, of course, is the linear predictor function, $d(x) = a + bx$. In general, the appropriate functional class may be inferred from observed data. The n pairs of measurements of the variables X and Y may be plotted as points, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, on the (x, y) plane. The resulting set of points is called a **scatter diagram**. If the observations are clustered about some curve in the (x, y) plane, then we may infer that the curve describes the dependence of Y on X . Such a curve is called an **approximating curve**. The inferred relationship could be **linear** as in Figure 11.1, or **nonlinear** as in Figure 11.2. On the other hand, no particular functional dependence of Y on X can be inferred from the scatter diagram in Figure 11.3.

Assume that a linear model of the dependence of Y on X is acceptable; that is, we restrict d to the class of functions of the form $d(x) = a + bx$. The problem of regression (or optimal prediction) then reduces to the problem of choosing the parameters a and b to minimize the following:

$$G(a, b) = E[D^2] = E[(Y - d(X))^2] = E[(Y - a - bX)^2].$$

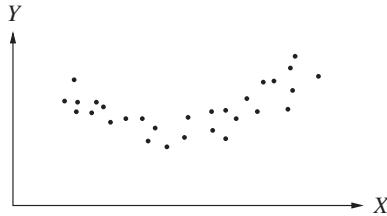


Figure 11.2. A scatter diagram showing a nonlinear dependence

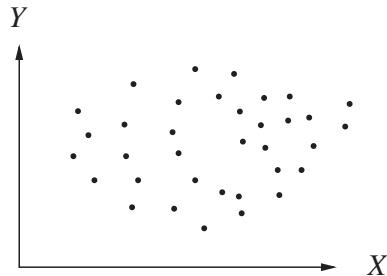


Figure 11.3. A scatter diagram showing no specific dependence

Let μ_X and μ_Y denote the respective expectations $E[X]$ and $E[Y]$, and the mean squared error may be rewritten as

$$\begin{aligned}
 G(a, b) &= E[(Y - a - bX)^2] \\
 &= E[((Y - \mu_Y) + (\mu_Y - a) - b(X - \mu_X) - b\mu_X)^2] \\
 &= E[(Y - \mu_Y)^2 + (\mu_Y - a)^2 + b^2(X - \mu_X)^2 + b^2\mu_X^2 \\
 &\quad + 2(Y - \mu_Y)(\mu_Y - a) - 2b(Y - \mu_Y)(X - \mu_X) \\
 &\quad - 2b(Y - \mu_Y)\mu_X - 2b(\mu_Y - a)(X - \mu_X) \\
 &\quad - 2b(\mu_Y - a)\mu_X + 2b^2(X - \mu_X)\mu_X].
 \end{aligned} \tag{11.1}$$

Note that $E[Y - \mu_Y] = E[X - \mu_X] = 0$, and let

$$\sigma_X^2 = \text{Var}[X], \sigma_Y^2 = \text{Var}[Y], \text{ and } \rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}.$$

Then

$$\begin{aligned}
 G(a, b) &= \sigma_Y^2 + b^2\sigma_X^2 + (\mu_Y - a)^2 + b^2\mu_X^2 \\
 &\quad - 2b\rho\sigma_X\sigma_Y - 2b\mu_X(\mu_Y - a) \\
 &= \sigma_Y^2 + b^2\sigma_X^2 + (\mu_Y - a - b\mu_X)^2 - 2b\rho\sigma_X\sigma_Y.
 \end{aligned}$$

To minimize $G(a, b)$, we take its partial derivatives with respect to a and b and set them equal to zero:

$$\begin{aligned}\frac{\partial G}{\partial a} &= -2(\mu_Y - a - b\mu_X) = 0, \\ \frac{\partial G}{\partial b} &= 2b\sigma_X^2 - 2\mu_X(\mu_Y - a - b\mu_X) - 2\rho\sigma_X\sigma_Y = 0.\end{aligned}$$

Thus the optimal values of a and b are given by

$$b = \rho \frac{\sigma_Y}{\sigma_X} \quad \text{and} \quad a = \mu_Y - b\mu_X = \mu_Y - \rho \frac{\sigma_Y}{\sigma_X} \mu_X$$

or

$$b = \frac{\text{Cov}(X, Y)}{\text{Var}[X]}, \quad (11.2)$$

and

$$a = E[Y] - \frac{\text{Cov}(X, Y)}{\text{Var}[X]} E[X]. \quad (11.3)$$

The corresponding linear regression curve is

$$y = E[Y] - \frac{\text{Cov}(X, Y)}{\text{Var}[X]} (E[X] - x), \quad (11.4)$$

which can be rewritten as

$$\frac{y - \mu_Y}{\sigma_Y} = \rho \frac{x - \mu_X}{\sigma_X}. \quad (11.5)$$

The regression line of Y on X can be derived in a similar manner:

$$\frac{x - \mu_X}{\sigma_X} = \rho \frac{y - \mu_Y}{\sigma_Y}. \quad (11.6)$$

The square of the minimum prediction error is

$$\begin{aligned}E[D^2] &= G(a, b) = G\left(\rho \frac{\sigma_Y}{\sigma_X}, \mu_Y - \rho \frac{\sigma_Y}{\sigma_X} \mu_X\right) \\ &= \sigma_Y^2 + \rho^2 \frac{\sigma_Y^2}{\sigma_X^2} \sigma_X^2 - 2\rho \frac{\sigma_Y}{\sigma_X} \rho \sigma_X \sigma_Y \\ &= \sigma_Y^2 + \rho^2 \sigma_Y^2 - 2\rho^2 \sigma_Y^2 \\ &= \sigma_Y^2 - \rho^2 \sigma_Y^2 = (1 - \rho^2) \sigma_Y^2.\end{aligned} \quad (11.7)$$

Recall that $-1 \leq \rho \leq 1$. From (11.7), if $\rho = \pm 1$, $E[D^2] = 0$. In this case the regression line of Y on X (11.5) and the regression line of X on Y (11.6) coincide with each other; hence X and Y are completely dependent and are said to have a **functional relationship** with each other. (A note of caution: a statistical relationship such as this, however strong, cannot logically imply a causal relationship.) In this case when $\rho^2 = 1$, the linear model is a perfect fit, and in terms of the joint distribution function of X and Y , this means that the entire probability mass is concentrated on the regression line. Also note that the best linear model may well be the best model, even when the fit is not perfect (viz., $\rho^2 \neq 1$). For example, it can be shown that when X and Y have a joint normal distribution, the regression curve of Y on X is necessarily linear.

If $\rho = 0$, then X and Y are uncorrelated and the two regression lines [equations (11.5) and (11.6)] are at right angles to each other. If X and Y are independent, then $\rho = 0$, but the converse does not hold in general. In the special case of bivariate normal distribution, it is true that $\rho = 0$ implies the independence of X and Y . For this reason ρ can be used as a measure of interdependence of X and Y only in cases of normal or near-normal variation. Otherwise ρ should be used as an *indicator* rather than a *measure* of interdependence. Furthermore, ρ is essentially a coefficient of *linear* interdependence, and more complex forms of interdependence lie outside its scope. If we suspect that $E[Y|x]$ is far from being linear in x , a linear predictor is of little value, and an appropriate nonlinear predictor should be sought.

In the case $\rho^2 \neq 1$, from (11.7) we conclude that there is a nonzero prediction error even when the parameters a and b are known exactly. For the minimizing values of a and b , from (11.7) we have

$$\sigma_Y^2 = \rho^2 \sigma_Y^2 + E[D^2]. \quad (11.8)$$

The term $\rho^2 \sigma_Y^2$ is interpreted as the variance of Y *attributable* to a linear dependence of Y on X , and the term $E[D^2] = (1 - \rho^2)\sigma_Y^2$ is the **residual variance** that cannot be “explained” by the linear relationship. Additional errors will occur if the parameters are estimated from observed data.

The predictor discussed above is attractive, since it requires only the knowledge of the first two moments and the correlation coefficient of the random variables X and Y . If true (population) values of these quantities are unknown, then they have to be estimated on the basis of a random sample of n pairs of observations of the two random variables.

More generally, suppose we are interested in modeling the input–output behavior of a stochastic system with m inputs (also called *independent variables*) $\mathbf{X} = (X_1, X_2, \dots, X_m)$ and output (or dependent variable) Y . The appropriate predictor of Y , $d(\mathbf{x}; \mathbf{a})$, is the conditional expectation:

$$E[Y|x_1, x_2, \dots, x_m].$$

The predictor $d(\mathbf{x}; \mathbf{a})$ is a parameterized family of functions with a vector of parameters \mathbf{a} . For example, in a model of a moving-head disk, the response

time to an I/O request may be the dependent variable; the record size and the request arrival rate, the independent variables; and the device transfer rate and average seek time, the parameters. If the internal system behavior is well understood (i.e., if the conditional distribution of Y is known), then the function $d(\mathbf{x}; \mathbf{a})$ can be derived analytically. Otherwise, the form of the function must be inferred empirically from observed data.

Once we have determined an appropriate functional class for the predictor d , the parameters \mathbf{a} have to be determined just as a and b were determined in our linear model $d(x) = a + bx$. Nonlinear models are, of course, more difficult to handle than linear models, because the equations

$$\frac{\partial}{\partial a_i} (E [(Y - d(\mathbf{X}; \mathbf{a}))^2]) = 0, \quad \text{for all } i$$

are more difficult to solve. We note that even if the regression function is nonlinear in the independent variable \mathbf{X} , it could still be a linear function of the parameters. The regression model will be linear in this case. Thus for instance the regression functions $d = a + b \log x$ or $d = a + b \sin x$ give rise to linear models.

11.2 LEAST-SQUARES CURVE FITTING

Suppose, after reviewing price lists of various manufacturers of high-speed random access memories, we have gathered the data plotted in Figure 11.4. Since price seems to be nearly a linear function of size, we might wish to present our findings in a compact form: the best line through these points. Of course, the term “best” is open to interpretation, and the “eyeball” method has a great following. Nevertheless, when a more precise analysis is required,

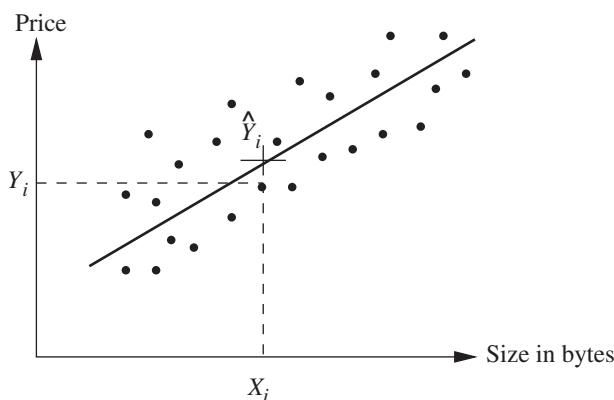


Figure 11.4. Least-squares linear fit

“best” is usually defined to be that line which minimizes the sum of the squares of the y -coordinate deviations from it.

To be more specific, if our points are $\{(x_i, y_i) | i = 1, \dots, n\}$, then we choose a and b so as to minimize

$$\sum_{i=1}^n [y_i - (a + bx_i)]^2. \quad (11.9)$$

Taking partial derivatives with respect to a and b and setting them equal to 0, we obtain

$$b = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sum_{i=1}^n x_i^2 - n (\bar{x})^2} \quad (11.10)$$

and

$$a = \bar{y} - b \bar{x},$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ and } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Such a technique does have a probabilistic counterpart. Consider two discrete random variables, X and Y , having joint pmf given by our plot of Figure 11.4, that is, $P(X = x_i, Y = y_i) = 1/n, i = 1, 2, \dots, n$. It is meaningful to ask: What is the regression line of Y on X ?

From the definition of regression line, we want to minimize:

$$E[(Y - (a + bX))^2] = \sum_{i=1}^n \frac{1}{n} [y_i - (a + bx_i)]^2. \quad (11.11)$$

Obviously the values of a and b that minimize (11.11) also minimize (11.9) and thus are given by (11.10).

This empirical approach to regression is known as the **method of least squares** and equations (11.10) are known as the **normal equations of least squares**.

Example 11.1

The failure rate of a certain electronic device is suspected to increase linearly with its temperature. Fit a least-squares linear line through the data in Table 11.1 (two measurements were taken for each given temperature, and hence we have twelve pairs of measurements).

The sample mean of the temperature is $\bar{x} = 80^\circ\text{F}$ and the sample mean of failure rates is $\bar{y} = 1.98 \cdot 10^{-6}$ failures per hour. From these values we get $a = 1.80 \cdot 10^{-6}$ failures per hour and $b = 0.00226$ failures per hour per degree Fahrenheit. Thus,

TABLE 11.1. The failure rate versus temperature

T (°F)	55	65	75	85	95	105
Failure rate · 10 ⁶	1.90	1.93	1.97	2.00	2.01	2.01
	1.94	1.95	1.97	2.02	2.02	2.04

$y = 1.80 + 0.00226x$ is the desired least-squares line. From this line we can obtain a predicted value of the failure rate for a specified temperature. For example, the predicted failure rate is $1.9582 \cdot 10^{-6}$ at 70°F .

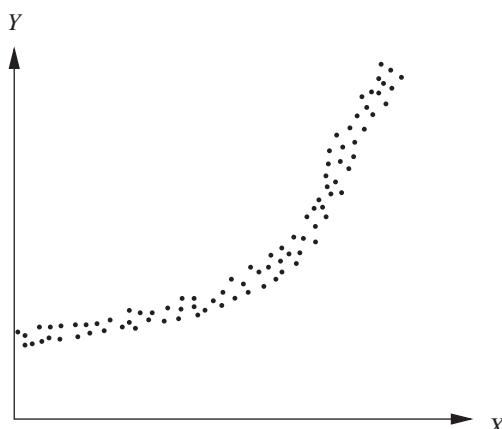
‡

Suppose now that we were faced with fitting a summary curve to the data of Figure 11.5, rather than those of Figure 11.4. Although we could still perform a least-squares linear fit, the result would be of at best questionable value, and we would certainly prefer to fit a nonlinear curve.

In general, as mentioned earlier in our discussion of regression, nonlinear fitting is much more difficult; nevertheless, in certain circumstances we can consider using our linear results to fit nonlinear curves. For example, suppose that we want to fit an exponential curve, $y = ae^{bx}$ to the data of Figure 11.5. We might reason as follows. If in each case $y \simeq ae^{bx}$, then $\ln y \simeq \ln a + bx$, so if we transform our data into pairs (x_i, z_i) , where $z_i = \ln y_i$, and we perform a least-squares linear fit to obtain the line $z = a' + b'x$, then

$$y = e^z = e^{a' + b'x} = e^{a'} e^{b'x} = ae^{bx},$$

where $a = e^{a'}$ and $b = b'$.

**Figure 11.5.** Nonlinear curve fitting

Example 11.2

Suppose that we are interested in fitting the price of a CPU of an IBM 370 series as a function of its speed x [measured in millions of additions per second (MAPS)]. Grosch's law suggests that price, y , will be roughly proportional to the square root of the speed, x . We use a general power function

$$y = a \cdot x^b$$

to model the relation between the price and the speed. Using the data from Hatfield and Sliski [HATF 1977] and using a transformation to $\ln y = \ln a + b \ln x$, Kinicki [KINI 1978] obtained the least-squares fit:

$$\text{Estimated price, } \hat{y} = \$1,147,835 \cdot x^{0.55309}.$$

¶

We must point out that the easy transformations illustrated here, although analytically precise, do not necessarily preserve the least-squares property; more specifically, consider the $y = ae^{bx}$ model: the values of a and b that minimize

$$\sum_{i=1}^n (y_i - ae^{bx_i})^2$$

are not necessarily the same as those we have chosen to use, which minimize

$$\sum_{i=1}^n [\ln y_i - (\ln a + bx_i)]^2.$$

(For an easy example, consider the data $\{(-1, 1), (0, e), (1, 1)\}$). Nevertheless, the transformation fit may be preferable, owing to ease of application.

Problem

- Consider an arithmetic unit of a computer system with a modulo- m online fault detector. As the modulus m varies, the average detection latency y also varies. Given the following data, with two observations of y for each value of m :

m_i	y_i (μs)	
3	1.45	1.5
5	1.30	1.26
7	1.20	1.23
11	1.10	1.08
13	1.05	1.03

Determine parameters a and b by performing a least-squares fit of the curve $y = am^b$ to the given data.

11.3 THE COEFFICIENTS OF DETERMINATION

Having obtained a least-squares linear fit to data, $y = a + bx$, we next consider the **goodness of fit** between this line and the data. For a point x_i , the value predicted by the fitted line is

$$\hat{y}_i = a + bx_i.$$

The difference $|y_i - \hat{y}_i|$ between the observed and the predicted values should be low for a good fit. Observe that

$$|y_i - \bar{y}| = |(y_i - \hat{y}_i) + (\hat{y}_i - \bar{y})|.$$

Squaring both sides and then summing, we get

$$\begin{aligned} \sum_{i=1}^n (y_i - \bar{y})^2 &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \\ &\quad + 2 \sum_{i=1}^n (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}). \end{aligned}$$

The last sum can be shown to equal zero if we substitute the linear predictor for \hat{y}_i :

$$\begin{aligned} &\sum_{i=1}^n (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) \\ &= \sum [(y_i - a - bx_i)(a + bx_i - \bar{y})] \\ &= a \sum (y_i - a - bx_i) + b \sum x_i (y_i - a - bx_i) \\ &\quad - \bar{y} \sum (y_i - a - bx_i) \\ &= 0, \end{aligned}$$

since a and b are defined to have those values for which

$$\frac{\partial}{\partial a} \sum_{i=1}^n (y_i - a - bx_i)^2 = 0$$

and

$$\frac{\partial}{\partial b} \sum_{i=1}^n (y_i - a - bx_i)^2 = 0.$$

It follows that

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2. \quad (11.12)$$

Note the similarity between this equation and equation (11.8). Here $y_i - \bar{y}$ is the deviation of the i th observed value of Y from its sample mean; therefore,

the left-hand side is the **sum of squares about the mean** (also called the **total variation**), and $\hat{y}_i - \bar{y}$ is the difference between the predicted value and the sample mean. This quantity is “explained” by the least-squares line, since $\hat{y}_i - \bar{y} = b(x_i - \bar{x})$. Therefore, $\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$ is called the **explained variation**. The quantity $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ is the **residual sum of squares** (also called the **unexplained variation**) and should be as small as possible. In fact, this sum would be zero if all the actual observations were to lie on the fitted line. This shows that the total variation can be partitioned into two components:

$$\text{Total variation} = \text{unexplained variation} + \text{explained variation}.$$

The ratio of the explained variation to the total variation, called the **coefficient of determination**, is a good measure of how well the line $y = ax + b$ fits the given data:

$$\begin{aligned} \text{Coefficient of determination} &= \text{explained variation}/\text{total variation} \\ &= \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}. \end{aligned} \quad (11.13)$$

The coefficient of determination must lie between 0 and 1, and the closer it is to 1, the better the fit.

Returning to Example 11.3, we compute the variation explained by regression to be 0.01788428 and the total variation to be 0.0206; hence the coefficient of determination is 0.8682, indicating a fairly good fit, since 86.82% of the variation in Y values is explained by the fitted line.

Example 11.3

The failure rate (hazard rate) $h(t)$ of a system is thought to be a power function of its age. In other words a Weibull model seems appropriate, so that

$$h(t) = ct^d, \quad t \geq 0.$$

A large number of systems are put on test, and we divide the number of failures in each hourly interval by the number of surviving components to obtain an observed value, h_i , of the hazard rate in the interval. The data for this example are shown in Table 11.2.

$$\begin{aligned} \sum y_i &= 13.1343, & \sum x_i &= 6.5598, \\ \sum y_i^2 &= 20.86416, & \sum x_i^2 &= 5.2152, \\ \sum x_i y_i &= 10.42965. \end{aligned}$$

Therefore

$$a = (\log c) = 0.008930 \text{ or } c \simeq 1.0208$$

TABLE 11.2. The failure rate data for a Weibull model

$t, \text{ h}$	$h_i \cdot 10^3$	$Y_i = \log h_i$	$x_i = \log t_i$	\hat{y}_i
0.5-1.5	1.05	0.02119	0.0000	0.008946
1.5-2.5	3.95	0.59600	0.3010	0.607515
2.5-3.5	8.20	0.91381	0.4771	0.957707
3.5-4.5	17.60	1.24551	0.6021	1.206282
4.5-5.5	25.00	1.39794	0.6990	1.398977
5.5-6.5	38.00	1.57978	0.7782	1.556474
6.5-7.5	49.00	1.69020	0.8451	1.689512
7.5-8.5	59.00	1.77085	0.9031	1.804850
8.5-9.5	85.00	1.92942	0.9542	1.906468
9.5-10.5	97.50	1.98900	1.0000	1.997546
		13.1343	6.5598	

and

$$b = d = 1.9886.$$

Also, $\sum (\hat{y}_i - \bar{y})^2 = 3.607$ and $\sum (y_i - \bar{y})^2 = 3.613$, which imply that the coefficient of determination is $3.607/3.613 = 0.9983$. Hence the quality of fit as indicated by the coefficient of determination is near perfect. The estimated hazard-rate function is $h(t) = 1.0208 \cdot t^{1.9886}$.

#

Problems

- Compute the coefficient of determination for the least-squares fit of the data in problem 1 in Section 11.2.
- Consider a network which uses the Transport Layer (layer 4 in the seven-layer OSI Reference Model) Security Protocol for information exchange. The real time needed for the information exchange increases linearly with increase in the data set size. Perform a least squares fit for the following data:

i	<i>Data set size</i>		<i>Real time</i>
	y_i	x_i	(s)
1	128		0.8145
2	256		0.7957
3	512		0.8002
4	1024		0.8016
5	2048		0.7698
6	4096		0.9112
7	8192		0.8306

- (a) Predict the real time for data transfer for data-set sizes of 524288 and 4194304 bytes.
- (b) Determine the coefficient of determination for the given data.

11.4 CONFIDENCE INTERVALS IN LINEAR REGRESSION

We return to the problem of linear regression. Now we assume, not that required parameters of the population distributions are known, but rather that they are estimated from a random sample of n pairs of observations. Because the conditional expectation of Y given X minimizes the mean-squared error, we usually consider X to be a controlled or nonrandom variable; that is, in sampling we can restrict ourselves to those points at which X takes on an *a priori* specified value. If we let Y_i denote the random variable Y restricted to those points at which $X = x_i$, then Y_i will be assumed to be normal with mean $\mu_i = a + bx_i$ and a common variance $\sigma^2 = \text{Var}[Y_i]$. In order to derive maximum-likelihood estimates of the two population parameters, a and b , we form the likelihood function of a and b :

$$\begin{aligned} L(a, b) &= \prod_{i=1}^n f(y_i) \\ &= \frac{\exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - a - bx_i)^2 \right]}{(2\pi)^{n/2} \sigma^n}. \end{aligned} \quad (11.14)$$

Taking logarithms and setting the derivatives with respect to a and b to zero, we get

$$\begin{aligned} \sum_{i=1}^n (y_i - a - bx_i) &= 0, \\ \sum_{i=1}^n (y_i - a - bx_i)x_i &= 0. \end{aligned}$$

The resulting maximum-likelihood estimates of a and b , denoted by \hat{a} and \hat{b} are then given by the normal equations (11.10). The corresponding estimators are

$$\hat{A} = \bar{Y} - \hat{B}\bar{x}, \quad (11.15)$$

and

$$\hat{B} = \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2}. \quad (11.16)$$

Now, if Y_i is not normally distributed, we can still obtain the point estimates of the parameters a and b by the method of least squares. Similarly, the discussion about the coefficient of determination holds in the general case. However, in order to derive confidence intervals for these estimates,

we need to make assumptions similar to those we made while deriving maximum-likelihood estimates of a and b . Thus we continue to assume that X is a controlled variable. We further assume that Y_1, Y_2, \dots, Y_n are mutually independent, and that the conditional distribution of Y_i is normal with mean $E[Y_i] = a + bx_i$, and that variance of Y_i is equal to σ^2 , that is, $\text{Var}[Y_i] = \sigma_i^2 = \sigma^2$.

If we write

$$Y_i = a + bx_i + \Delta_i, \quad (11.17)$$

then $\Delta_1, \Delta_2, \dots, \Delta_n$ are mutually independent normal random variables with zero mean and a common variance σ^2 , that is, $E[\Delta_i] = 0$ and $\text{Var}[\Delta_i] = \sigma^2$, for all i .

THEOREM 11.1.

- (a) \hat{A} is an unbiased estimator of a : $E[\hat{A}] = a$.
- (b) \hat{B} is an unbiased estimator of b : $E[\hat{B}] = b$.
- (c) $\text{Var}[\hat{A}] = \frac{\sigma^2}{n} \left[1 + \frac{n\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]$.
- (d) $\text{Var}[\hat{B}] = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$.

Proof: First we show part (b):

$$\begin{aligned} E[\hat{B}] &= E \left[\frac{\sum(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2} \right] \\ &= \frac{\sum(x_i - \bar{x})E[Y_i - \bar{Y}]}{\sum(x_i - \bar{x})^2}. \end{aligned}$$

Now since $E[Y_i] = a + bx_i$ and

$$E[\bar{Y}] = \frac{1}{n} \sum_{i=1}^n E[Y_i] = a + b\bar{x},$$

we have

$$\begin{aligned} E[\hat{B}] &= \frac{\sum(x_i - \bar{x})(a + bx_i - a - b\bar{x})}{\sum(x_i - \bar{x})^2} \\ &= b. \end{aligned}$$

To show part (a), we write

$$\begin{aligned} E[\hat{A}] &= E[\bar{Y} - \hat{B}\bar{x}] \\ &= E[\bar{Y}] - E[\hat{B}]\bar{x} \end{aligned}$$

$$\begin{aligned}
 &= a + b\bar{x} - b\bar{x} \\
 &= a.
 \end{aligned}$$

Next to show part (d), we rewrite

$$\hat{B} = \frac{\sum(x_i - \bar{x})Y_i}{\sum(x_i - \bar{x})^2}.$$

Since the $\{Y_i\}$ are independent, we have

$$\begin{aligned}
 \text{Var}[\hat{B}] &= \frac{1}{[\sum(x_i - \bar{x})^2]^2} \sum \text{Var}[(x_i - \bar{x})Y_i] \\
 &= \sum \frac{(x_i - \bar{x})^2 \sigma^2}{[\sum(x_i - \bar{x})^2]^2} \\
 &= \frac{\sigma^2}{\sum(x_i - \bar{x})^2}.
 \end{aligned}$$

The result of part (c) follows in a similar fashion.

It is also clear that \hat{A} and \hat{B} are both normally distributed. If the variance σ^2 of the error term Δ_i is known, we can use the $\text{Var}[\hat{A}]$ and $\text{Var}[\hat{B}]$ above to obtain confidence intervals for a and b . Usually, however, σ^2 will not be known in advance and it must be estimated by s^2 :

$$\begin{aligned}
 s^2 &= \frac{\sum_{i=1}^n (y_i - \hat{a} - \hat{b}x_i)^2}{n-2} \\
 &= \frac{\sum (y_i - \hat{y}_i)^2}{n-2} \\
 &= \frac{\text{unexplained variation}}{n-2}. \tag{11.18}
 \end{aligned}$$

The denominator $(n - 2)$ reflects the fact that 2 degrees of freedom have been lost since \hat{a} and \hat{b} have been estimated from the given data.

If we use s^2 in place of σ^2 in estimating $\text{Var}[\hat{A}]$ and $\text{Var}[\hat{B}]$, we must use the Student t distribution with $(n - 2)$ degrees of freedom in place of a normal distribution. Now a $100(1 - \alpha)$ percent confidence interval for b is given by

$$\hat{b} \pm t_{n-2;\alpha/2} \cdot s \cdot \left[\sum_{i=1}^n (x_i - \bar{x})^2 \right]^{-\frac{1}{2}} \tag{11.19}$$

and for a by

$$\hat{a} \pm t_{n-2;\alpha/2} \cdot s \cdot \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]^{\frac{1}{2}}. \tag{11.20}$$

Example 11.4

Returning to the Weibull model and the corresponding failure rate data of Example 11.3, we compute the unexplained variation to be 0.006 and hence

$$s^2 = 0.006/8 \quad \text{or} \quad s = 0.02738.$$

Also

$$\sum_i (x_i - \bar{x})^2 = 0.91211 \quad \text{and} \quad \sqrt{\sum_i (x_i - \bar{x})^2} = 0.95504.$$

Thus, the required confidence intervals are obtained using the t distribution with $n - 2 = 8$ degrees of freedom. Since

$$t_{8;0.05} = 1.860,$$

a 90% confidence interval for the exponent d is given by

$$1.98864 \pm 0.05332.$$

Similarly, a 90% confidence interval for $a (= \log c)$ is given by 0.008930 ± 0.03851 .

#

Problems

1. Complete the proof of Theorem 11.1 by showing part (c).
2. Compute 90% confidence intervals for the parameters a and b from the data in problem 1 of Section 11.2.

11.5 TREND DETECTION AND SLOPE ESTIMATION

In many problems, the important objective may be to detect and estimate trends in monitored parameters over time. For example, Garg et al. [GARG 1998] analyzed operating system resource usage data collected over time for increasing or decreasing trends due to software aging. The slope of the trend is then estimated to give a measure of the effect of aging on these resources. This section describes the Mann–Kendall test for detecting trends and Sen’s nonparametric slope (trend) estimator.

11.5.1 Mann–Kendall Test

Smoothing techniques provide good visual clues for finding trends in time series data (data observed sequentially over time), but it is difficult to make conclusive statements regarding presence or absence of trend since no statistical significance levels are attached. The Mann–Kendall test [GILB 1987], a nonparametric test for zero slope of the linear regression of time-ordered data

versus time, is particularly useful in such cases. Since it uses only the relative magnitudes of the data and not the actual measured values, it can be used for data reported as trace or less than the detection limit. The main objective in the Mann–Kendall test is to test the null hypothesis H_0 that there is no trend, against the alternative hypothesis H_1 that there is an upward or a downward trend. To that end, we compute the Mann–Kendall statistic, S_i , where

$$S = \sum_{k=1}^{n-1} \sum_{l=k+1}^n sgn(y_l - y_k) \quad (11.21)$$

for $l > k$ and $sgn(x)$ is the signum function:

$$sgn(x) = \begin{cases} -1, & \text{if } x < 0 \\ 0, & \text{if } x = 0 \\ 1, & \text{if } x > 0 \end{cases}$$

n is the number of data points and y_l is the datum for the l th timepoint. To test the null hypothesis, H_0 is rejected in favor of H_1 upward (resp. downward) trend if S is positive (negative) and if the critical value in Table C.11 (of Appendix C) corresponding to the absolute value of the computed S is less than the specified significance level of the test, α . For a two-tailed test, the tabled probability level corresponding to the absolute value of S is doubled before comparison with the significance level, α . The above test is used if n , the number of data points, is 40 or less. For n larger than 40, the normal approximation test described later is used. The Mann–Kendall test is simple, efficient and robust against any missing values in the data. Furthermore, the data do not have to conform to any particular distribution; thus, it is a nonparametric test.

Example 11.5

Suppose that we wish to test the null hypothesis H_0 of no trend versus the alternative hypothesis H_1 , of an upward trend at the significance level, $\alpha = 0.10$, and the measurements collected over equal time intervals are: 25, 36, 33, and 51, in that order. For simplicity, we have assumed only a single observation per time period and that there are no ties in the data values. For the multiple observations and tied data values, refer to Gilbert [GILB 1987]. Here $n = 4$, and hence there are six differences to consider: 36–25, 33–25, 51–25, 33–36, 51–36 and 51–33. From equation (11.21), $S = +1 + 1 + 1 - 1 + 1 + 1 = 4$. The tabled probability for $S = +4$ and $n = 4$ is 0.167. Since this value is greater than 0.10, we cannot reject H_0 .

If the data had been 45, 56, 57, and 89, then $S = +6$ and the tabled probability would be 0.042 for $n = 4$ and $\alpha = 0.1$. Hence, we reject H_0 and accept the alternative hypothesis H_1 of an upward trend.

#

For $n > 40$, we use the normal approximation test. S is first computed using equation (11.21) and then the variance of S is computed using the

following equation, which takes into account ties that may be present in the data:

$$\text{Var}[S] = \frac{1}{18} \left[n(n-1)(2n+5) - \sum_{p=1}^g t_p(t_p-1)(2t_p+5) \right], \quad (11.22)$$

where g is the number of tied groups and t_p is the number of data points in the p th group. For example, if the data are 12, 14, 15, 19, 14, 14, 15, and 12, the number of tied groups $g = 3$, $t_1 = 2$ for value 12, $t_2 = 3$ for value 14 and $t_3 = 2$ for value 15.

Both, S and $\text{Var}[S]$ are then used to compute the Z statistic as follows:

$$Z = \begin{cases} \frac{S-1}{[\text{Var}[S]]^{1/2}} & \text{if } S > 0 \\ 0 & \text{if } S = 0 \\ \frac{S+1}{[\text{Var}[S]]^{1/2}} & \text{if } S < 0 \end{cases} \quad (11.23)$$

If Z is positive (resp. negative), it indicates an upward (resp. downward) trend. To decide whether to accept or reject hypothesis H_0 of no trend for a significance level α , we compare the absolute computed value of the Z statistic to the critical value of Z obtained from the normal distribution table (Table C.3), in a similar manner as described for $n \leq 40$.

11.5.2 Sen's Slope Estimator

Once the presence of a trend is confirmed by the procedure described above, its true slope may be estimated by computing the least-squares estimate of the slope by linear regression methods. These, however, deviate greatly from the true value if there are gross errors or outliers in the data. To overcome this limitation, a nonparametric procedure developed by Sen [SEN 1968] can be used. This method is not greatly affected by outliers and is also robust against missing values.

First, N' slopes are calculated for all pairs of points at l and k for which $l > k$, as $Q = (y_l - y_k)/(l - k)$. These N' slopes are then ranked and their median is calculated. This median is the required slope estimate, N . A $100(1-\alpha)\%$ confidence interval about the true slope can also be computed [GILB 1987].

Example 11.6

Suppose that we wish to estimate the slope of the following data: 34, 45, 51, 49, observed at equally spaced time intervals. Here, $n = 4$ and there are $N' = 6$ slope estimates: 9, 8.5, 6, 5, 2, and -2 ($(45-34)/2$, $(51-34)/2$, $(49-34)/3$, $(49-45)/2$ and $49-51$). Ranking these slopes from the smallest to the largest, we get -2, 2, 5, 6, 8.5, and 9. The median of these slopes, $N = (5+6)/2 = 5.5$, is the Sen estimate of the true slope.

11.6 CORRELATION ANALYSIS

In the last section we assumed that X was a controlled or nonrandom variable. Consider again the case when X and Y are both random variables. A random sample will then consist of pairs of observations $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ from a bivariate distribution.

Recall from Chapter 4 that the correlation coefficient $\rho(X, Y)$ gives an indication of the linearity of the relationship between X and Y . If X and Y are independent, then they are uncorrelated; that is, $\rho(X, Y) = 0$. The converse does not hold, in general. However, if X and Y are jointly normal, then $\rho(X, Y) = 0$ (X and Y are uncorrelated) implies that they are also independent. We will assume that $f(x, y)$ is a bivariate normal pdf, so that

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \cdot \exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X} \right)^2 - 2\rho \left(\frac{x-\mu_X}{\sigma_X} \right) \left(\frac{y-\mu_Y}{\sigma_Y} \right) + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right] \right\} \quad (11.24)$$

where

$$\mu_X = E[X], \mu_Y = E[Y], \sigma_X^2 = \text{Var}[X], \sigma_Y^2 = \text{Var}[Y]$$

and $\rho = \rho(X, Y)$.

We can form the likelihood function for a random sample of size n as

$$L(\mu_X, \mu_Y, \sigma_X, \sigma_Y, \rho) = \prod_{i=1}^n f(x_i, y_i).$$

Taking natural log on both sides, we obtain

$$\begin{aligned} \ln L &= \sum_{i=1}^n \ln f(x_i, y_i) \\ &= -n \ln (2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}) - \frac{1}{2(1-\rho^2)} \left\{ \sum_{i=1}^n \left[\left(\frac{x_i-\mu_X}{\sigma_X} \right)^2 - 2\rho \left(\frac{x_i-\mu_X}{\sigma_X} \right) \left(\frac{y_i-\mu_Y}{\sigma_Y} \right) + \left(\frac{y_i-\mu_Y}{\sigma_Y} \right)^2 \right] \right\}. \end{aligned}$$

Taking the partial derivatives with respect to the five parameters, and setting them equal to zero, we get the maximum-likelihood estimates

$$\hat{\mu}_X = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}, \quad \hat{\mu}_Y = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y},$$

$$s_X = \left(\frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_X)^2 \right)^{1/2}, \quad s_Y = \left(\frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mu}_Y)^2 \right)^{1/2} \quad (11.25)$$

and

$$\hat{\rho}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left(\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right)^{1/2}}. \quad (11.26)$$

The estimate $\hat{\rho}(X, Y)$ is also referred to as the *sample correlation coefficient* (let the corresponding estimator be denoted by \hat{R}). The distribution of \hat{R} is difficult to obtain. When $\rho = \rho(X, Y) = 0$ (X and Y are independent), it can be shown [KEND 1961] that the statistic

$$T = \hat{R} \left(\frac{n-2}{1-\hat{R}^2} \right)^{1/2} \quad (11.27)$$

has a Student t distribution with $(n-2)$ degrees of freedom. The test for the hypothesis (of *independence* of normal random variables X and Y)

$$H_0 : \rho(X, Y) = 0 \quad \text{versus the alternative} \quad H_1 : \rho(X, Y) \neq 0$$

with level of significance α is to reject the hypothesis H_0 when $|T| > t_{n-2;\alpha/2}$, where $t_{n-2;\alpha/2}$ denotes the critical value of the t distribution with $(n-2)$ degrees of freedom in the usual way. To test the hypothesis

$$H_0 : \rho(X, Y) = 0 \quad \text{versus the alternative} \quad H_1 : \rho(X, Y) > 0,$$

with a level of significance α , we reject H_0 if $T > t_{n-2;\alpha}$.

Example 11.7

It is believed that the number of breakdowns of a computing center is related to the number of jobs processed. Data were collected for two centers, A and B, as follows:

1. For center A, data on the number of breakdowns per month (y_i) and the number of jobs completed per month (x_i) were collected for 10 months.
2. For center B, data of x_i and y_i were collected for 20 months.

The maximum-likelihood estimates of the correlation coefficients for the two cases were computed using formula (11.26) to be $\hat{\rho}_A = 0.49$ and $\hat{\rho}_B = 0.55$. The corresponding values of the t statistic were computed using formula (11.27): $t_A = 1.5899$ and $t_B = 2.794$. Recalling that a t distribution with 8 degrees of freedom is applicable for center A, and that $t_{8;0.05} = 1.86$, we see that we cannot reject the hypothesis $H_0 : \rho_A = 0$ in favor of $H_1 : \rho_A \neq 0$ at the significance level 0.1.

On the other hand, for center B, we use a t distribution with 18 degrees of freedom and $t_{18;0.01} = 2.552$ to reject $H_0 : \rho_B = 0$ in favor of $H_1 : \rho_B \neq 0$ at the 0.02 level of significance; that is, there is a significant correlation between breakdowns and workload.

‡

Problems

- Associated with a job are two random variables: CPU time required (Y) and the number of disk I/O operations (X). Given the following data, compute the sample correlation coefficient:

i	<i>Time</i> (s)	<i>Number</i>
	y_i	x_i
1	40	398
2	38	390
3	42	410
4	50	502
5	60	590
6	30	305
7	20	210
8	25	252
9	40	398
10	39	392

Draw a scatter diagram from these data. Does a linear fit seem reasonable? Assuming that we wish to predict the CPU time requirement given an I/O request count, perform a linear regression:

$$y = a + bx.$$

Compute point estimates of a and b as well as 90% confidence intervals.

Next suppose that we want to predict a value of I/O request count, given a CPU time requirement. Thus, perform a linear regression of X on Y . Calculate 90% confidence intervals for c and d with the regression line:

$$x = c + dy.$$

In both cases compute the coefficients of determination.

- Since the method described for testing the hypothesis of no correlation in this section is based on a stringent assumption that the joint density of X and Y is bivariate normal, it is desirable to design a *nonparametric* alternative to this test in case the distributional assumption is not satisfied. First, we arrange both the X and Y sample values separately in increasing order of magnitudes and assign ranks to these values. Now let δ_i be the difference between the ranks of the paired

observations (x_i, y_i) . The Spearman **rank correlation coefficient** is defined to be [NOET 1976]:

$$r_s = 1 - \frac{6 \sum_{i=1}^n \delta_i^2}{n(n^2 - 1)}.$$

Compute r_s for the data in problem 1 above. Note that r_s is a value of a random variable R_S . Suppose that we want to perform a distribution-free test of the null hypothesis of no correlation. Then we can use the fact that under the null hypothesis, the distribution of R_S is approximately normal with:

$$\begin{aligned} E[R_S | H_0] &= 0, \\ \text{Var}[R_S | H_0] &= \frac{1}{n-1}. \end{aligned}$$

Carry out this test for the data in problem 1 above, and compute the descriptive level of the test.

11.7 SIMPLE NONLINEAR REGRESSION

Nonlinear regression presents special difficulties. The first and foremost of these, of course, is that the basic system of equations, given by

$$\frac{\partial}{\partial a_i} E[(Y - d(X; a_1, a_2, \dots, a_k))^2] = 0, \quad i = 1, 2, \dots, k,$$

is, in general, nonlinear in the unknown parameters. Nonlinear systems of equations are considerably more difficult to solve than linear systems. The difficulty is compounded when we realize that we are attempting to locate the lowest point (global minimum) of a multidimensional surface. If the surface has multiple valleys (i.e., it is not unimodal), most numerical techniques can do no better than to find a local minimum (one of the valleys) rather than the global minimum.

Example 11.8

The price y of a semiconductor memory module is suspected to be a nonlinear function of its capacity x :

$$y = a + bx^c.$$

Even the logarithmic transformation that we used earlier does not help here. The empirical mean squared error is given by

$$G(a, b, c) = \sum_{i=1}^n (y_i - a - bx_i^c)^2.$$

Using either a nonlinear (unconstrained) minimization routine directly or the derivative method and solving the resulting system of nonlinear equations is quite complex. We observe that if a is known, the problem can be transformed to a linear regression

situation. Since a is not known, we can iterate on different values of a to find an optimum point.

On the basis of data collected from manufacturers (117 data points), Maieli [MAIE 1980] obtained the following least-squares fit for the above model of semiconductor memory: $b = 0.75608$ (indicating a price per bit of about $\frac{3}{4}$ cents), $c = 0.72739$ (indicating a significant economy of scale), and $a = -7488$ cents. The coefficient of determination was found to be 0.982, indicating a very good fit.

‡

Problems

- Consider a computer system that is subject to periodic diagnosis and maintenance every 1000 h. The diagnosis-maintenance service is assumed not to be perfect, and the probability of its being able to correctly diagnose and correct the fault (if it exists) is c . The expected life y of the system is to be fitted as a power function of the coverage factor c :

$$y = a + be^c.$$

(Here e denotes the base of the natural logarithm.) Using the following data (adapted from Ingle and Siewiorek [INGL 1976]), estimate the parameters a , b , and compute the coefficient of determination (if this is found to be unsatisfactory, then plotting a scatter diagram may help you find a more appropriate function class):

c_i	y_i (h)
0.2	11,960
0.4	15,950
0.6	23,920
0.8	47,830
0.9	95,670
0.92	120,000
0.94	159,500
0.96	239,200
0.98	478,340

- Refit the data of problem 1 of Section 11.2, to the curve

$$y = a_1 m^{b_1} + c_1$$

Compare the quality of two fits by comparing the error sum of squares in the two cases.

11.8 HIGHER-DIMENSIONAL LEAST-SQUARES FIT

The treatment of least-squares linear fit to data can be extended in a simple way to cover problems involving several independent variables [DRAP 1966].

For example, suppose we wish to fit a three-dimensional plane

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3,$$

to a set of n points:

$$(y_1, x_{11}, x_{21}, x_{31}), \dots, (y_n, x_{1n}, x_{2n}, x_{3n})$$

in four dimensions. The empirical mean squared error is given by

$$G(a_0, a_1, a_2, a_3) = \sum_{i=1}^n (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i} - a_3 x_{3i})^2. \quad (11.28)$$

Setting partial derivatives with respect to the four parameters equal to zero, we get the following solution for $\mathbf{a} = (a_0, a_1, a_2, a_3)$, given in matrix form:

$$\mathbf{a} = (X^T X)^{-1} X^T \mathbf{y} \quad (11.29)$$

where $\mathbf{y} = (y_1, y_2, y_3, \dots, y_n)$,

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} \\ 1 & x_{12} & x_{22} & x_{32} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{3n} \end{bmatrix}$$

and superscript T denotes matrix transpose.

Example 11.9 (Overhead Regression)

In a given interval of time, the CPU usage may be divided into idle time, user time, and supervisory overhead time. To improve performance, we would like to investigate how to reduce the supervisory overhead. This overhead is caused by various calls to the supervisor. If x_i denotes the number of calls of type i and y denotes the overhead per unit of real time, then a linear model of overhead might be

$$y = a_0 + \sum a_i x_i.$$

From the given measurement data, $(y_1, x_{11}, x_{21}, \dots), (y_2, x_{12}, x_{22}, \dots), \dots, (y_n, x_{1n}, x_{2n}, \dots)$, we can use our techniques [equation (11.29)] to estimate the parameters a_0, a_1, \dots . A large value of a_i indicates that the i th type of supervisor call contributes heavily to the overhead. We could try to reduce the number of calls of this type or reduce the execution time of the corresponding service routine to reduce the value of a_i .

For example, this method was used in a study of CP-67 system [BARD 1978]. Letting x_1 = number of virtual selector I/O, x_2 = number of pages read, and x_3 =

number of spool I/O, corresponding coefficients were obtained to be $a_1 = 9.7$, $a_2 = 6.1$, and $a_3 = 6.0$. A redesign of the supervisory routines led to a reduction in these numbers, $a_1 = 7.9$, $a_2 = 2.0$, and $a_3 = 4.6$.

‡

As a special case of the discussion in this section, consider polynomials of the form $y = \sum_{j=0}^k a_j x^j$, which are often useful in practice. These polynomials are linear in the unknown parameters and, hence, produce a linear system of equations for the unknowns. In fact, these equations can be obtained by substituting $(x_i)^j$ for x_{ji} , in equation (11.29) where x_i is the i th observation of the independent variable.

Problems

- Refer to Putnam [PUTN 1978]. Let K denote the total man-years required for a software project over its lifecycle. Let T_d be the development time until it is put into operation. The ratio $Y = K/T_d$ is to be fitted as a linear function of the number of report types (x_1) to be generated and the number of application subprograms (x_2) from the following data regarding 19 software systems, where the y_i values are in person-years per year. Obtain the parameters of the linear fit and compute the coefficient of determination.

x_{1i}	x_{2i}	y_i
45	52	32.28
44	31	56.76
74	39	19.76
34	23	35.00
41	35	19.10
5	5	5.00
14	12	3.97
10	27	36.40
95	109	48.09
109	229	57.95
179	256	191.78
101	144	115.43
192	223	216.48
215	365	240.75
200	398	342.82
59	75	98.85
228	241	224.55
151	120	55.66
101	130	50.35

11.9 ANALYSIS OF VARIANCE

In many practical situations we wish to determine which of the many independent variables affect the particular dependent variable. Regression analysis can be used to answer such questions, if the independent variables take numeric values. Situations often arise where qualitative variables are encountered. For instance, we may wish to study the effect of disk scheduling algorithm on disk request throughput. The independent variable representing the scheduling algorithm may take FCFS, SCAN, SSTF, and so on as its values. (For a discussion of disk scheduling algorithms, see Coffman and Denning [COFF 1973].) In such problems, regression is not directly applicable.

The technique known as **analysis of variance** is usually cast in the framework of **design and analysis of experiments**. The first step in planning a measurement experiment is to formulate a clear statement of the objectives. The second step is the choice of dependent variable (also called the **response variable**).

The third step is to identify the set of all independent variables (called **factors**) that can potentially affect the value of the response variable. Some of these factors may be quantitative (e.g., the number of I/O operations performed by a job) while others may be qualitative (e.g., the disk scheduling algorithm). A particular value of a factor is called a **level**.

A factor is said to be **controllable** if its levels can be set by the experimenter, while the levels of an **uncontrollable** (or **observable**) factor cannot be set but only observed. For example, if we wish to test a logic circuit and if it is being tested on line, then its input variables are observable factors. On the other hand, if the circuit is being tested off line, then it can be driven by a chosen set of inputs, and these factors are then controllable. Similarly, when a performance measurement is conducted on a computer system under its production workload, its workload factors are uncontrollable. In order to make the workload factors controllable, the experiment can be conducted using synthetic jobs. If there are m controllable factors, an m -tuple of assignments of a level to each of those factors is called a **treatment**.

The purpose in applying analysis of variance to an experimental situation is to compare the effect of several simultaneously applied factors on the response variable. This technique allows us to separate the effects of interest (those due to controllable factors) from the **uncontrolled** or **residual variation**. It allows us not only to gauge the effects of individual factors but also to study the interactions between the factors.

In order to study the effects of factors on the response variable, we must start with an empirical model. It is usual to assume that the effects of various factors and the interactions are **additive**.

First we deal with one-way analysis of variance, where we wish to study the effect of one controllable factor on the response variable. Assume that the factor can take c different levels. For the i th level we assume that the response

variable Y_i takes the form

$$Y_i = \mu_i + \Delta_i, \quad (11.30)$$

where Δ_i is a random variable that captures the effect of all uncontrollable factors. We will assume that $\Delta_1, \Delta_2, \dots, \Delta_c$ are mutually independent, normally distributed random variables with zero means and a common variance σ^2 ; that is, $E[\Delta_i] = 0$ and $\text{Var}[\Delta_i] = \sigma^2$ for all i . Even though this last assumption may not hold in practice, it has been shown that the technique of analysis of variance is fairly robust, in that it is relatively insensitive to violations of the assumption of normality as well as the assumption of equal variances.

Suppose that n observations are taken on each of the c levels for a total of $n_T = nc$ observations. Let y_{ij} be the j th observed value of the response variable at the i th level. The random variable Y_{ij} possesses the same distribution as Y_i :

$$Y_{ij} = \mu_i + \Delta_{ij}, \quad i = 1, 2, \dots, c, \quad j = 1, 2, \dots, n. \quad (11.31)$$

It is convenient to rewrite the above equation so that:

$$Y_{ij} = \mu + \alpha_i + \Delta_{ij}, \quad i = 1, 2, \dots, c, \quad j = 1, 2, \dots, n, \quad (11.32)$$

where μ is the overall average $\frac{1}{c} \sum_{i=1}^c \mu_i$, α_i is the effect of the i th level, and Δ_{ij} is the random error term. Note that $\mu_i = \mu + \alpha_i$, and that $\sum_{i=1}^c \alpha_i = 0$.

Such data are usually organized in the form of a table, as shown in Table 11.3. It is common to think of level i defining a separate population with corresponding population mean μ_i . Let the i th sample mean $\bar{Y}_{i\cdot} = \frac{1}{n} \sum_{j=1}^n Y_{ij}$ where a dot in the subscript indicates the index being summed over. Then $\bar{Y}_{i\cdot}$ is the minimum-variance unbiased estimator of the population mean μ_i .

Our aim is to compare the observed sample means. If the observed means are close together, then the differences can be attributed to residual or chance variation. On the other hand, if the observed sample means are dispersed, then there is reason to believe that the effects of different treatments are significantly different. The problem can be formulated as a hypothesis test:

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_c \quad \text{versus} \quad H_1 : \mu_i \neq \mu_j \text{ for some } i \neq j. \quad (11.33)$$

TABLE 11.3. Observations for one-way analysis of variance

Level	Observations	Sample mean	Population mean	Sample variance
1	$y_{11}, y_{12}, \dots, y_{1n}$	$\bar{y}_{1\cdot}$	μ_1	s_1^2
2	$y_{21}, y_{22}, \dots, y_{2n}$	$\bar{y}_{2\cdot}$	μ_2	s_2^2
\vdots	\vdots	\vdots	\vdots	\vdots
c	$y_{c1}, y_{c2}, \dots, y_{cn}$	$\bar{y}_{c\cdot}$	μ_c	s_c^2

If there are only two levels, then this is a special case of hypothesis test on two means as discussed in Chapter 10. In that case a t test can be used to compare \bar{y}_1 and \bar{y}_2 . In the general case of c levels, we may be tempted to perform a series of t tests between each pair of sample means. As the number of levels c increases, however, this procedure not only encounters the problem of combinatorial growth but also suffers a loss of significance. For example, suppose there are 6 levels and hence $\binom{6}{2} = 15$ t tests need to be performed. Assume that each test is conducted at the 5% level of significance (i.e., the probability of a type I error in each test is 0.05). Assuming that the composite hypothesis H_0 is true, the probability that each individual t test will lead us to accept the hypothesis $\mu_i = \mu_j$ is $1 - 0.05 = 0.95$. All fifteen individual hypotheses must be accepted in order for us to accept H_0 . Assuming that these tests are mutually independent, the probability associated with this event is $(0.95)^{15}$; hence the probability of an overall type I error is $1 - (0.95)^{15} \simeq 0.537$. Clearly, this approach is not feasible for problems of reasonable size.

The method we develop for testing hypothesis (11.33) is based on comparing different estimates of the population variance σ^2 . This variance can be estimated by any one of the sample variances:

$$S_i^2 = \frac{1}{n-1} \sum_{j=1}^n (Y_{ij} - \bar{Y}_{i.})^2 \quad (11.34)$$

and, hence, also by their mean:

$$S_W^2 = \sum_{i=1}^c \frac{S_i^2}{c} = \frac{\sum_{i=1}^c \sum_{j=1}^n (Y_{ij} - \bar{Y}_{i.})^2}{n_T - c}. \quad (11.35)$$

Note that $E[S_W^2] = \sigma^2$. The subscript W reminds us that this is a **within-sample variance**. By our assumptions, $Y_{ij} \sim N(\mu_i, \sigma^2)$ hence $\bar{Y}_{i.} \sim N(\mu_i, \sigma^2/n)$. It follows that

$$\frac{(n-1)S_i^2}{\sigma^2}$$

has a chi-square distribution with $(n-1)$ degrees of freedom, and

$$\frac{(n_T - c)S_W^2}{\sigma^2}$$

has a chi-square distribution with $(n_T - c)$ degrees of freedom.

An alternative method of estimating σ^2 is to obtain the variance of the c sample means:

$$S_{\bar{Y}_{..}}^2 = \frac{\sum_{i=1}^c (\bar{Y}_{i.} - \bar{Y}_{..})^2}{c-1}, \quad (11.36)$$

where the overall sample mean is defined by

$$\bar{Y}_{..} = \frac{\sum_{i=1}^c \bar{Y}_{i..}}{c} = \frac{\sum_{i=1}^c \sum_{j=1}^n Y_{ij}}{n_T}. \quad (11.37)$$

Noting that

$$\begin{aligned} E[\bar{Y}_{i..}] &= \mu_i, & E[\bar{Y}_{..}] &= \mu, \\ \text{Var}[\bar{Y}_{i..}] &= \frac{\sigma^2}{n}, & \text{Var}[\bar{Y}_{..}] &= \frac{\sigma^2}{n_T}, \end{aligned}$$

we have

$$E[\bar{Y}_{i..}^2] = \text{Var}[\bar{Y}_{i..}] + \mu_i^2 = \frac{\sigma^2}{n} + \mu_i^2$$

and

$$E[\bar{Y}_{..}^2] = \text{Var}[\bar{Y}_{..}] + \mu^2 = \frac{\sigma^2}{n_T} + \mu^2.$$

Now, taking expectations on both sides of equation (11.36), we have

$$\begin{aligned} E[S_{\bar{Y}_{..}}^2] &= E\left[\frac{\sum \bar{Y}_{i..}^2}{c-1} - \frac{c}{c-1}\bar{Y}_{..}^2\right] \\ &= \frac{1}{c-1} \left(\frac{c\sigma^2}{n} + \sum_{i=1}^c \mu_i^2 \right) - \frac{c}{c-1} \left(\frac{\sigma^2}{nc} + \mu^2 \right) \\ &= \frac{\sigma^2}{n} + \frac{\sum_{i=1}^c \mu_i^2 - c\mu^2}{c-1}. \end{aligned} \quad (11.38)$$

Thus, if the null hypothesis H_0 is true (i.e., $\mu_i = \mu$ for all i), then the right-hand side in equation (11.38) reduces to σ^2/n . Therefore, if the null hypothesis is true, then

$$S_B^2 = nS_Y^2 = \frac{n}{c-1} \sum_{i=1}^c (\bar{Y}_{i..} - \bar{Y}_{..})^2 \quad (11.39)$$

is an unbiased estimator of σ^2 . The subscript B refers to the fact that S_B^2 is a measure of the **between-sample (intersample) variance**. Furthermore, under H_0 , the equation

$$\frac{(c-1)}{\sigma^2/n} S_Y^2 = \frac{(c-1)S_B^2}{\sigma^2}$$

has a chi-square distribution with $(c-1)$ degrees of freedom.

If H_0 is true, S_W^2 and S_B^2 should have comparable values since they both estimate σ^2 . It is reasonable, therefore, to base our test on their ratio. The statistic (often called the **variance ratio**):

$$F = \frac{\frac{(c-1)S_B^2}{\sigma^2(c-1)}}{\frac{(n_T - c)S_W^2}{\sigma^2(n_T - c)}} = \frac{S_B^2}{S_W^2} \quad (11.40)$$

has an F distribution with $(c-1, n_T - c)$ degrees of freedom, by Theorem 3.9. Since between-sample variance S_B^2 is expected to be larger than within-sample variance S_W^2 when H_0 is false, we reject H_0 at α percent level of significance if the observed variance ratio exceeds the critical value $f_{c-1, n_T - c; \alpha}$ of the F distribution.

To gain further insight into the problem, we observe that we can partition the total sum of squares (SST)

$$SST = \sum_{i=1}^c \sum_{j=1}^n (Y_{ij} - \bar{Y}_{..})^2 \quad (11.41)$$

as follows:

$$\begin{aligned} SST &= \sum \sum (Y_{ij} - \bar{Y}_{..})^2 \\ &= n \sum_{i=1}^c (\bar{Y}_{i.} - \bar{Y}_{..})^2 + \sum_{i=1}^c \sum_{j=1}^n (Y_{ij} - \bar{Y}_{i.})^2. \end{aligned} \quad (11.42)$$

The first term on the right-hand side is S_B^2 times its degrees of freedom; hence it is known as the **sum of squares between treatments** (or groups), $SS(Tr)$. The second term on the right-hand side is S_W^2 times its degrees of freedom; hence it is a measure of the chance error. This sum is referred to as the **residual variation, or error sum of squares, SSE**. Thus, an alternative form of equation (11.42) is

$$SST = SS(Tr) + SSE \quad (11.43)$$

and the variance ratio is expressed by

$$F = \frac{\left(\frac{SS(Tr)}{c-1}\right)}{\left(\frac{SSE}{n_T - c}\right)}. \quad (11.44)$$

Numerical calculation of this F statistic is usually expressed in terms of a so-called analysis of variance (ANOVA) table as shown in Table 11.4.

TABLE 11.4. One-way ANOVA table

<i>Source of variation</i>	<i>Sum of squares</i>	<i>Degrees of freedom</i>	<i>Mean square</i>
Between treatments	$SS(Tr) = n \sum_{i=1}^c (\bar{Y}_{i\cdot} - \bar{Y}_{..})^2$	$c - 1$	S_B^2
Error (within groups)	$SSE = \sum_{i=1}^c \sum_{j=1}^n (Y_{ij} - \bar{Y}_{i\cdot})^2$	$n_T - c$	S_W^2
Total variation	$SST = \sum_{i=1}^c \sum_{j=1}^n (Y_{ij} - \bar{Y}_{..})^2$	$n_T - 1$	

Example 11.10

Three different interactive systems are to be compared with respect to their response times to an editing request. Owing to chance fluctuations among the other transactions in process, it was decided to take 10 sets of samples at randomly chosen times for each system and record the mean response time as follows:

<i>Session</i>	<i>System response time (s)</i>		
	A	B	C
1	0.96	0.82	0.75
2	1.03	0.68	0.56
3	0.77	1.08	0.63
4	0.88	0.76	0.69
5	1.06	0.83	0.73
6	0.99	0.74	0.75
7	0.72	0.77	0.60
8	0.86	0.85	0.63
9	0.97	0.79	0.59
10	0.90	0.71	0.61

The ANOVA table for this problem can be formulated as follows:

<i>Source of variation</i>	<i>Sum of squares</i>	<i>Degrees of freedom</i>	<i>Mean square</i>	<i>F</i>
Between treatments	0.34041	2	0.1702033	17.4276
Error	0.26369	27	0.0097663	
Total	0.60410	29		

Since $f_{2,27;0.01}$ is 5.49, and the observed value is 17.4276, we reject the null hypothesis at the 1% level of significance. In other words, there is a significant difference in the responsiveness of the three systems.



Next we consider a two-way analysis of variance so that we have two controllable factors, and the response variable Y_{ij} is modeled as follows:

$$\begin{aligned} Y_{ij} &= \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \Delta_{ij}, \\ i &= 1, 2, \dots, c_1, j = 1, 2, \dots, c_2. \end{aligned} \quad (11.45)$$

There are c_1 levels of the first factor, c_2 levels of the second factor, and the total number of **cells** or treatments is $c_T = c_1 c_2$. The term μ is the overall mean. The term α_i is the main effect of the first factor at level i , the term β_j accounts for the main effect of the second factor at level j , and $(\alpha\beta)_{ij}$ denotes the **interaction** (or joint) effect of the first factor at level i and the second factor at level j . As before, we will assume that Δ_{ij} 's ($i = 1, 2, \dots, c_1; j = 1, 2, \dots, c_2$) are mutually independent, normally distributed random variables with zero means and the common variance σ^2 . We will further assume without loss of generality that

$$\sum_{i=1}^{c_1} \alpha_i = \sum_{j=1}^{c_2} \beta_j = \sum_{i=1}^{c_1} (\alpha\beta)_{ij} = \sum_{j=1}^{c_2} (\alpha\beta)_{ij} = 0. \quad (11.46)$$

These restrictions allow us to uniquely estimate the parameters μ, α_i, β_j , and $(\alpha\beta)_{ij}$.

As in the case of one-way analysis of variance, we assume that n independent observations are taken for each i and j so that

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \Delta_{ijk} \quad (11.47)$$

($i = 1, 2, \dots, c_1, j = 1, 2, \dots, c_2, k = 1, 2, \dots, n$) denote the random variables corresponding to $n_T = c_1 c_2 n$ total observations of the response variable. These observations are usually organized as shown in Table 11.5.

The following quantities are calculated:

$$\begin{aligned} \bar{Y}_{ij.} &= \frac{\sum_{k=1}^n Y_{ijk}}{n}, \quad \text{the sample average in } (i, j)\text{th cell,} \\ \bar{Y}_{i..} &= \frac{\sum_{j=1}^{c_2} \bar{Y}_{ij.}}{c_2} = \frac{\sum_{j=1}^{c_2} \sum_{k=1}^n Y_{ijk}}{nc_2}, \quad \text{sample average of row } i, \\ \bar{Y}_{.j.} &= \frac{\sum_{i=1}^{c_1} \bar{Y}_{ij.}}{c_1} = \frac{\sum_{i=1}^{c_1} \sum_{k=1}^n Y_{ijk}}{nc_1}, \quad \text{sample average of column } j, \\ \bar{Y}_{...} &= \frac{\sum_{i=1}^{c_1} \bar{Y}_{i..}}{c_1} = \frac{\sum_{j=1}^{c_2} \bar{Y}_{.j.}}{c_2}, \\ &= \frac{\sum_{i=1}^{c_1} \sum_{j=1}^{c_2} \sum_{k=1}^n Y_{ijk}}{nc_1 c_2}, \quad \text{overall sample average.} \end{aligned}$$

TABLE 11.5. Observations for two-way analysis of variance

Factor A	Factor B				Row average
	Level 1	Level 2	...	Level c_2	
Level 1	y_{111}, \dots, y_{11n}	y_{121}, \dots, y_{12n}	...	$y_{1c_21}, \dots, y_{1c_2n}$	$\bar{y}_{1..}$
Level 2	y_{211}, \dots, y_{21n}	y_{221}, \dots, y_{22n}	...	$y_{2c_21}, \dots, y_{2c_2n}$	$\bar{y}_{2..}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
Level c_1	$y_{c_111}, \dots, y_{c_11n}$	$y_{c_121}, \dots, y_{c_12n}$...	$y_{c_1c_21}, \dots, y_{c_1c_2n}$	$\bar{y}_{c_1..}$
Column average	$\bar{y}_{.1..}$	$\bar{y}_{.2..}$...	$\bar{y}_{.c_2..}$	$\bar{y}_{...}$

It should be clear that these averages allow us to compute the best unbiased estimates of the model parameters

$$\hat{\mu} = \bar{y}_{...}, \hat{\alpha}_i = \bar{y}_{i..} - \bar{y}_{...}, \hat{\beta}_j = \bar{y}_{.j..} - \bar{y}_{...},$$

$$(\hat{\alpha}\hat{\beta})_{ij} = (\bar{y}_{ij..} - \bar{y}_{...}) - [(\bar{y}_{i..} - \bar{y}_{...}) + (\bar{y}_{.j..} - \bar{y}_{...})].$$

We are usually interested in testing one or more hypotheses regarding model parameters. For example, one hypothesis could be that the main effect of factor A is zero:

$$H_{01} : \alpha_i = 0 \quad \text{for all } i.$$

Another hypothesis relates the main effect of factor B:

$$H_{02} : \beta_j = 0 \quad \text{for all } j.$$

Finally, we may be interested in testing the hypothesis that the effects of the two factors are additive; that is, there is no interaction:

$$H_{03} : (\alpha\beta)_{ij} = 0 \quad \text{for all } i \text{ and } j.$$

To proceed further, we break down the total sum of squares SST into four components: row sum of squares SSA , the main effect of factor A; column sum of squares SSB , the main effect of factor B; the interaction sum of squares $SSAB$; and the residual or error sum of squares SSE . Thus we have

$$SST = SSA + SSB + SSAB + SSE, \tag{11.48}$$

where

$$SST = \sum_i \sum_j \sum_k (Y_{ijk} - \bar{Y}_{...})^2,$$

$$\begin{aligned} SSA &= nc_2 \sum_i (\bar{Y}_{i..} - \bar{Y}_{...})^2, \\ SSB &= nc_1 \sum_j (\bar{Y}_{.j.} - \bar{Y}_{...})^2, \\ SSAB &= n \sum_i \sum_j (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...})^2, \\ SSE &= \sum_i \sum_j \sum_k (Y_{ijk} - \bar{Y}_{ij.})^2. \end{aligned}$$

These quantities are usually tabulated as shown in Table 11.6.

In order to test the hypothesis H_{01} , we compute the F statistic:

$$F_1 = \frac{\frac{SSA}{c_1 - 1}}{\frac{SSE}{n_T - c_T}}.$$

We reject H_{01} at the α percent level of significance, provided that the computed value of F_1 exceeds the critical value $f_{c_1-1, n_T - c_T; \alpha}$ of the F distribution. Similarly, we compute:

$$F_2 = \frac{\frac{SSB}{c_2 - 1}}{\frac{SSE}{n_T - c_T}} \text{ and } F_3 = \frac{\frac{SSAB}{(c_1 - 1)(c_2 - 1)}}{\frac{SSE}{n_T - c_T}}$$

and respectively reject H_{02} and H_{03} at the α percent level of significance if the computed values exceed the critical values of F distributions with $(c_2 - 1, n_T - c_T)$ and $[(c_1 - 1)(c_2 - 1), n_T - c_T]$ degrees of freedom. Clearly, this procedure can be extended to the case of multifactor analysis of variance [NETE 1974].

The following example is a slightly modified version of an example in another study [LIU 1978].

Example 11.11

Suppose that we wish to study the variation in the throughput of a paged multiprogramming system with the following five factors:

1. POL—the memory-partitioning policy at 1 of 14 possible levels
2. ML—multiprogramming level at levels 3, 4, and 5
3. MPP—memory allotment per program at levels 20, 26, and 40 pages per program
4. DRM—drum rotation time at three levels: 10, 35, and 70 ms
5. WRK—four different workload types

TABLE 11.6. table

<i>Source</i>	<i>Sum of squares</i>	<i>Mean squared deviation</i>	<i>Degrees of freedom</i>	<i>Test statistic</i>	<i>E[mean squared deviation]</i>
Main effect A (rows)	SS_A	$S_A^2 = \frac{SS_A}{c_1 - 1}$	$c_1 - 1$	$F_1 = \frac{S_A^2}{S_W^2}$	$\sigma^2 + nc_2 \sum \frac{\alpha_i^2}{c_1 - 1}$
Main effect B (columns)	SS_B	$S_B^2 = \frac{SS_B}{c_2 - 1}$	$c_2 - 1$	$F_2 = \frac{S_B^2}{S_W^2}$	$\sigma^2 + nc_1 \sum \frac{\beta_j^2}{c_2 - 1}$
Interaction	SS_{AB}	$S_{AB}^2 = \frac{SS_{AB}}{(c_1 - 1)(c_2 - 1)}$	$(c_1 - 1)(c_2 - 1)$	$F_3 = \frac{S_{AB}^2}{S_W^2}$	$\sigma^2 + nc^2 \sum \frac{(\alpha\beta)_{ij}^2}{(c_1 - 1)(c_2 - 1)}$
Residual	SSE	$S_W^2 = \frac{SSE}{n_T - c_T}$	$c_1 c_2 (n - 1)$		σ^2
Total				$= n_T - c_T$	$n_T - 1$

For each one of the $14 \cdot 3 \cdot 3 \cdot 3 \cdot 4 = 1512$ cells, a simulation is conducted with a sample size of $n = 25$. Selecting system throughput as the response variable, the ANOVA is shown in Table 11.7.

From this table we conclude that all main effects and two-way interactions are significant at the 1% level. An ordering of factors from the largest to the smallest effect is as follows: drum rotation time, memory allotment, workload type, the degree of multiprogramming, and the memory-partitioning policy. The largest two-way interaction effect is the one due to the combination of workload levels and drum

TABLE 11.7. ANOVA Table for Example 11.11

<i>Source of variation</i>	<i>Sum of squares</i>	<i>Degrees of freedom</i>	<i>Mean-squared deviation</i>	<i>Test statistic</i>	$f_{.,.,0.01}$
Main effects	21.238				
POL	0.401	13	0.03085	136.5	2.21
ML	0.253	2	0.12650	559.7	4.60
MPP	4.128	2	2.064	9132.7	4.60
DRM	15.053	2	7.5265	33303.1	4.60
WRK	1.403	3	0.4677	2069.5	3.78
Two-way interaction	2.206				
POL-ML	insig.	26	—	—	—
POL-MPP	0.261	26	0.01004	44.425	1.76
POL-DRM	0.043	26	0.001654	7.319	1.76
POL-WRK	0.067	39	0.001718	7.602	1.60
ML-MPP	insig.	4	—	—	—
ML-DRM	0.022	4	0.0055	24.337	3.32
ML-WRK	0.013	6	0.002167	9.588	2.80
MPP-DRM	0.204	4	0.051	225.66	3.32
MPP-WRK	0.179	6	0.029834	132.01	2.80
DRM-WRK	1.417	6	0.236167	1045.00	2.80
Three-way interaction	0.411				
MPP-DRM-WRK	0.230	12	0.019167	84.81	2.18
Other	0.181				
Residual	00.82	36,288	0.000226		
Total	24.775	37,799			

rotation times. The information gained from such an analysis can be used to improve system performance.

‡

Problems

- Suppose that we wish to study the effect of computer-aided instruction (CAI) on the performance of college students. We randomly divide the incoming class into three sections. Section A is taught in a conventional way, Section B is (nearly) completely automated, and in Section C a mixed approach is used. The following test scores are observed:

<i>Section A</i>	<i>Section B</i>	<i>Section C</i>
77	70	79
68	69	74
72	73	77
75	74	80
60	59	73
59	63	60
82	80	79

Perform an analysis of variance and determine whether the differences among the means obtained for the three sections are significant at $\alpha = 0.05$.

- The **Kruskal–Wallis H test** is a nonparametric analog of test (11.33) in one-way analysis of variance [NOET 1976]. This test is a generalization of the Wilcoxon rank sum test considered in Section 10.3.2. Suppose that n_i independent observations for treatment level i have been taken. We combine $k = \sum_{i=1}^c n_i$ observations and order them by increasing magnitude. Now we assign ranks to the combined sample values and let R_i be the sum of the ranks assigned to the n_i observations of treatment level i . The test is based on statistic H , defined by

$$H = \frac{12}{k(k+1)} \sum_{i=1}^c \frac{R_i^2}{n_i} - 3(k+1).$$

If $n_i > 5$ for all i and if the null hypothesis that the k samples come from identical populations holds, then the distribution of the H statistic is approximately chi-square with $(c - 1)$ degrees of freedom. Apply the H test to the data in Example 11.10 and to the data in problem 1 above.

REFERENCES

- [BARD 1978] Y. Bard and M. Schatzoff, “Statistical methods in computer performance analysis,” in K. M. Chandy and R. T. Yeh (eds.), *Current Trends in Programming Methodology*, Vol.III, Prentice-Hall, Englewood Cliffs, NJ, 1978.

- [COFF 1973] E. G. Coffman, Jr. and P. J. Denning, *Operating System Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [DRAP 1966] N. R. Draper and H. Smith, *Applied Regression Analysis*, Wiley, New York, 1966.
- [GARG 1998] S. Garg, A. Van Moorsel, K. Vaidyanathan and K. S. Trivedi, "A methodology for detection and estimation of software aging", *Proc. 9th Intl. Symp. Software Reliability Engineering*, Paderborn, Germany, Nov. 1998, pp. 282–292
- [GILB 1987] R. O. Gilbert, *Statistical Methods for Environmental Pollution Monitoring*, Van Nostrand Reinhold, New York, 1987.
- [HATF 1977] M. A. Hatfield and D. J. Sliski (eds.), *Computer Review*, GML Corp., Lexington, MA, 1977.
- [INGL 1976] A. D. Ingle and D. P. Siewiorek, *Reliability Models for Multiprocessor Systems with and without Periodic Maintenance*, Technical Report, Department of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1976.
- [KEND 1961] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 2, *Inference and Relationship*, Hafner, New York, 1961.
- [KINI 1978] R. E. Kinicki, *Queuing Models of Computer Configuration Planning*, Ph.D. dissertation, Department of Computer Science, Duke Univ., Durham, NC, 1978.
- [LIU 1978] M. Liu, "A simulation study of memory partitioning algorithms," in D. Ferrari (ed.), *Performance of Computer Installations*, North-Holland, Amsterdam, 1978.
- [MAIE 1980] M. V. Maieli, *The Significant Parameters Affecting the Price of Matrix-Configured Random Access Computer Memory and Their Functional Relationship*, A.M. dissertation, Department of Computer Science, Duke Univ., Durham, NC, 1980.
- [NETE 1974] J. Neter and W. Wasserman, *Applied Linear Statistical Models*, Richard D. Irwin, Homewood, IL, 1974.
- [NOET 1976] G. E. Noether, *Introduction to Statistics, A Nonparametric Approach*, Houghton Mifflin, Boston, MA.
- [PUTN 1978] L. Putnam, "A general empirical solution to the macro software sizing and estimating problem," *IEEE Trans. on Soft. Eng.*, (July 1978).
- [SEN 1968] P. K. Sen, "Estimates of the regression coefficient based on kendall's tau", *J. Am. Stat. Assoc.*, **63** 1379–1389–(1968).

Appendix A

Bibliography

A.1 THEORY

A.1.1 Probability Theory

S. Asmussen, *Applied Probability and Queues*, Wiley, New York, 1987.

K. L. Chung, *A Course in Probability Theory*, 2nd edition, Academic Press, New York, 2000.

A. B. Clarke and R. L. Disney, *Probability and Random Processes for Engineers and Scientists*, Wiley, New York, 1970 (introductory).

W. Feller, *An Introduction to Probability Theory and Its Applications*, 2 vols., Wiley, New York, 1968 (introductory and advanced).

B. V. Gnedenko and I. A. Ushakov, *Theory of Probability*, 6th ed., G & B Science Publishers, 1997.

R. Nelson, *Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modelling*, Springer-Verlag, New York, 1995.

E. Parzen, *Modern Probability Theory*, Wiley, New York, 1960.

S. M. Ross, *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco, 1970.

Probability and Statistics with Reliability, Queueing, and Computer Science Applications, Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

S. M. Ross, *Introduction to Probability Models*, Academic Press, New York, 1993.

A. N. Shiryaen, R. P. Boas, and A. N. Shiriaev, *Probability*, 2nd ed., Springer-Verlag, 1996.

A.1.2 Stochastic Processes

U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd ed., Wiley, New York, 1984 (intermediate).

E. Cinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1975 (advanced).

D. R. Cox, *Renewal Theory*, Methuen, London, 1962 (advanced).

D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*, Chapman & Hall, London, 1965.

S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes*, Academic Press, New York, 1975.

E. P. C. Kao, *An Introduction to Stochastic Processes*, Duxbury Press, 1997.

G. Kemeny and J. L. Snell, *Finite Markov Chains*, Van Nostrand-Reinhold, New York, 1960.

M. Kijima, *Markov Processes for Stochastic Modelling*, Chapman & Hall, New York, 1997.

V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, Chapman & Hall, London, 1995.

V. G. Kulkarni, *Modeling, Analysis, Design, and Control of Stochastic Systems*, Springer, New York, 1999.

J. Medhi, *Stochastic Processes*, Wiley Eastern Limited, New Delhi, India, 1994.

E. Parzen, *Stochastic Processes*, Holden-Day, San Francisco, 1962 (intermediate).

N. U. Prabhu, *Stochastic Processes: Basic Theory and Its Applications*, Macmillan, New York, 1965 (advanced).

S. M. Ross, *Stochastic Process*, 2nd ed., Wiley, New York, 1995.

H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, Academic Press, New York, 1994.

H. C. Tijms, *Stochastic Models: An Algorithmic Approach*, John Wiley & Sons, New York, 1995.

A.1.3 Queuing Theory

U. N. Bhat and I. V. Basawa (eds.), *Queueing and Related Models*, Oxford Univ. Press, 1992.

G. Bolch, S. Greiner, H. De Meer and K. S. Trivedi, *Queueing Networks and Markov Chains*, Wiley, New York, 1998.

O. J. Boxma and R. Syski (eds.), *Queueing Theory and Its Applications*, North-Holland, 1988.

S. C. Bruell and G. Balbo, *Computational Algorithms for Closed Queueing Networks*, Elsevier North-Holland, New York, 1980.

J. W. Cohen, *The Single Server Queue*, 2nd ed., North Holland, New York, 1982.

R. B. Cooper, *Introduction to Queueing Theory*, 2nd ed., North-Holland, New York, 1981 (introductory).

D. R. Cox and W. L. Smith, *Queues*, CRC Press, 1999.

J. N. Daigle, *Queueing Theory for Telecommunications*, Addison-Wesley, 1992.

B. V. Gnedenko and I. N. Kovalenko, *Introduction to Queueing Theory*, Springer-Verlag, 1989.

D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 3rd ed., Wiley, New York, 1998.

F. P. Kelley, *Reversibility and Stochastic Networks*, Wiley, New York, 1979 (advanced).

A. Y. Khintchine, *Mathematical Methods in Queueing Theory*, Griffen, London, 1960.

L. Kleinrock, *Queueing Systems*, Vol. I, *Theory*, Wiley, New York, 1975 (intermediate to advanced).

L. Kleinrock, *Queueing Systems*, Vol. II, Wiley, New York, 1976.

- L. Lipsky, *Queueing Theory: A Linear Algebraic Approach*, Macmillan, New York, 1992.
- C. H. Ng, *Queueing Modelling Fundamentals*, Wiley, 1997.
- H. Perros, *Queueing Networks With Blocking: Exact and Approximate Solutions*, Oxford Univ. Press, 1994.
- N. U. Prabhu, *Foundations of Queueing Theory*, Kluwer Academic Publishers, 1997.
- H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation. Vol. 1: Vacation and Priority Systems. Vol. 2: Finite Systems. Vol. 3: Discrete Time Systems*. North-Holland, 1991, 1993, and 1993.
- L. Takacs, *Introduction to the Theory of Queues*, Oxford Univ. Press, New York, 1972.
- N. M. Van Dijk, *Queueing Networks and Product Form: A Systems Approach*, Wiley, New York, 1993.
- J. Walrand, *An Introduction to Queueing Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- R. Wolff, *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, 1989.

A.1.4 Reliability Theory

- R. E. Barlow, *Engineering Reliability*, Wiley, New York, 1998.
- R. E. Barlow and F. Proschan, *Mathematical Theory of Reliability*, Wiley, New York, 1966 (advanced).
- R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, c/o Gordon Pledger, Silver Spring, MD, 1979.
- R. Billinton and R. N. Allan, *Reliability Evaluation of Engineering Systems: Concepts and Techniques*, Pitman Publishing, Marshfield, MA, 1983.
- A. Birolini, *Reliability Engineering: Theory and Practice*, Springer-Verlag, 1999.
- B. S. Dhillon and C. Singh, *Engineering Reliability: New Techniques and Applications*, Wiley, New York, 1981.

B. S. Dhillon, *Reliability Engineering in Systems Design and Operation*, Van Nostrand Reinhold, New York, 1983.

I. N. Kovalenko, N. Yu. Kuznetsov and P. A. Pegg, *Mathematical Theory of Reliability of Time-Dependent Systems*, Wiley, Chichester, 1997.

L. M. Leemis, *Reliability Probabilistic Models and Statistical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

K. B. Misra, *Reliability Analysis and Prediction: A Methodology Oriented Treatment*, Elsevier, Amsterdam, 1992.

K. B. Misra (ed.), *New Trends in System Reliability Evaluation*, Elsevier, Amsterdam, 1993.

S. Ozekici (ed.), *Reliability and Maintenance of Complex Systems*, Springer-Verlag, Berlin, 1996.

W. Schneeweiss, *The Fault Tree Method*, LiLoLe-Verlag, Hagen, Germany, 1999.

W. Schneeweiss, *Petri Nets for Reliability Modeling*, LiLoLe-Verlag, Hagen, Germany, 1999.

M. L. Shooman, *Probabilistic Reliability: An Engineering Approach*, 2nd ed., McGraw-Hill, 1990 (introductory).

P. A. Tobias and D. C. Trindale, *Applied Reliability*, 2nd ed., CRC Press, 1995.

A.1.5 Statistics

D. A. Berry and B. W. Lindgren, *Statistics: Theory and Methods*, 2nd ed., Duxbury Press, 1995.

M. H. Degroot, *Probability and Statistics*, 2nd ed., Addison-Wesley, 1986.

N. R. Draper and H. Smith, *Applied Regression Analysis*, 3rd ed., Wiley, New York, 1998 (intermediate).

R. V. Hogg and E. A. Tanis, *Probability and Statistical Inference*. 6th ed., Prentice-Hall, New York, 2001 (intermediate).

M. Hollander and D. A. Wolfe, *Non-parametric Statistical Methods*, Wiley, New York, 1973 (intermediate).

- M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 2, *Inference and Relationship*, 4th ed., Oxford Univ. Press, New York, 1979 (advanced).
- H. J. Larson, *Introduction to Probability Theory and Statistical Inference*, 3rd ed., Wiley, New York, 1982.
- R. A. Johnson, I. Miller and J. E. Freund, *Miller and Freund's Probability and Statistics for Engineers*, 5th ed., Prentice-Hall, 1993 (introductory).
- A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics*, 3rd ed., McGraw-Hill, New York, 1979 (intermediate to advanced).
- M. H. Kuther, C. J. Nachtsheim, J. Neter and W. Wasserman, *Applied Linear Statistical Models*, 4th ed., McGraw-Hill, New York, 1996.
- G.E. Noether, *Elements of Nonparametric Statistics*, Wiley, New York, 1967.
- G. E. Noether, *Introduction to Statistics: The Nonparametric Way*, Springer-Verlag, New York, 1991.
- E. S. Pearson and H. O. Hartley, *Biometrika Tables for Statisticians*, 3rd ed., Cambridge Univ. Press, Cambridge, UK, 1966.
- M. R. Spiegel, *Schaum's Outline of Theory and Problems of Statistics*, 2nd ed., McGraw-Hill, New York, 1998 (introductory).
- R. E. Walpole, *Introduction to Statistics*, 3rd ed., Macmillan, New York, 1982.
- S. S. Wilks, *Mathematical Statistics*, Wiley, New York, 1962 (advanced).

A.2 APPLICATIONS

A.2.1 Computer Performance Evaluation

- A. O. Allen, *Introduction to Computer Performance Analysis with Mathematica*, AP (Academic Press) Professional, 1994.
- M. Ajmone-Marsan, G. Balbo and G. Conte, *Performance Models of Multiprocessor Systems*, MIT Press, Cambridge, MA, 1986.
- M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, Wiley, New York, 1995.

- K. M. Chandy and R. T. Yeh, *Current Trends in Programming Methodology*, Vol. III, *Software Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1978 (intermediate to advanced).
- P. J. Courtois, *Decomposability: Queuing and Computer System Applications*, Academic Press, New York, 1977 (advanced).
- L. Donatiello and R. Nelson (eds.), *Performance Evaluation of Computer and Communication Systems*, Lecture Notes in Computer Science, Springer-Verlag, 1993.
- L. Dowdy and C. Lowery, *P.S. to Operating Systems*, Prentice-Hall, 1993.
- D. Ferrari, *Computer Systems Performance Evaluation*, Prentice-Hall, Englewood Cliffs, NJ, 1978 (intermediate).
- D. Ferrari, G. Serrazi, and A. Zeigner, *Measurement and Tuning of Computer Systems*, Prentice-Hall, 1983.
- R. German, *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*, Wiley, New York, 2000.
- N. J. Guenther, *The Practical Performance Analyst: Performance by Design Techniques for Distributed Systems*, McGraw-Hill, 1998.
- G. Haring, C. Lindemann, and M. Reiser (eds.), *Performance Evaluation—Origins and Directions*, Lecture Notes in Computer Science, Springer-Verlag, 2000.
- P. G. Harrison and N. M. Patel, *Performance Modeling of Communication Networks and computer Architectures*, Addison-Wesley, 1993.
- J. Hillston, *A Compositional Approach to Performance Modelling*, Cambridge Univ. Press, 1996.
- B. R. Haverkort, *Performance of Computer Communication System: A Model-Based Approach*, Wiley, New York, 1998.
- B. R. Haverkort, R. Marie, K. S. Trivedi, and G. Rubino (eds.), *Performability Modelling Tools and Techniques*, Wiley, New York, in press.
- R. Jain, *The Art of Computer Systems Performance Analysis Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley, New York, 1991.
- K. Kant, *Introduction to Computer System Performance Evaluation*, McGraw Hill, New York, 1992.

- S. S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press, New York, 1983.
- E. D. Lazowska, J. Zahorjan, G. S. Graham and K. C. Sevcik, *Quantitative System Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
- C. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*, Wiley, 1998.
- E. A. Macnair and C. H. Sauer, *Elements of Practical Performance Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- D. A. Menasce, V. A. F. Almeida, and L. Dowdy *Capacity Planning and Performance Modeling: from Mainframes to Client-Server Systems.*, Prentice-Hall, 1994.
- M. K. Molloy, *Fundamentals of Performance Modeling*, Macmillan, New York, 1989.
- C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1981 (introductory).
- C. U. Smith, *Performance Engineering of Software Systems*, Addison-Wesley Pub. Co., 1990.
- J. R. Spirn, *Program Behavior: Models and Measurements*, Elsevier, New York, 1977.
- A. Thomasian, *Database Concurrency Control: Methods, Performance, and Analysis*, Kluwer, 1996.
- .

A.2.2 Communications

- V. F. Alisouskas and W. Tomasi, *Digital and Data Communications*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- T. S. Rappaport, *Wireless Communications: Principles & Practice*, Prentice-Hall, Englewood, 1996.
- H. Stark and F. B. Tuteur, *Modern Electrical Communications*, Prentice-Hall, Englewood Cliffs, NJ, 1979.
- A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, Reading, MA, 1995.

A.2.3 Analysis of Algorithms

M. Hofri, *Probabilistic Analysis of Algorithms: On Computing Methodologies for Computer Algorithms Performance Evaluation*, Springer-Verlag, New York, 1987.

D. E. Knuth, *The Art of Computer Programming*, Vol 1, *Fundamental Algorithms*, 3rd ed., Addison-Wesley, Reading, Mass., 1997 (Intermediate to Advanced).

D. E. Knuth, *The Art of Computer Programming*, Vol 2, *Seminumerical Algorithms*, 3rd ed., Addison-Wesley, Reading, Mass., 1997 (Intermediate to Advanced).

D. E. Knuth, *The Art of Computer Programming*, Vol 3, *Sorting and Searching*, 2nd ed., Addison-Wesley, Reading, Mass., 1998 (Intermediate to Advanced).

.

A.2.4 Simulation

K. Bagchi and G. Zobrist (eds.), *State-of-the Art in Performance Modeling and Simulation. Modeling and Simulation of Advanced Computer Systems: Applications and Systems*, Gordon & reach Publishers, Newark, NJ, 1998.

J. Banks, J. S. Carson II, B. L. Nelson and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 2000.

G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York, 1995.

J. P. C. Kleijnen and W. Van Groenendaal, *Simulation: A Statistical Perspective*, Wiley, New York, 1992.

B. L. Nelson, *Stochastic Modeling: Analysis and Simulation*, McGraw-Hill, New York, 1994.

R. Y. Rubinstein and B. Melamed, *Modern Simulation and Modeling*, Wiley, New York, 1997.

R. Y. Rubinstein, *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*, Wiley, New York, 1986.

A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 2nd ed., McGraw-Hill, New York, 1991.

T. J. Schriber, *Introduction to Simulation*, Wiley, New York, 1991.

S. V. Hoover and R. F. Perry, *Simulation: A Problem-Solving Approach*, Addison-Wesley, Reading, MA, 1989.

A.2.5 Computer-Communication Networks

H. Akimaru and K. Kawashima, *Teletraffic: Theory and Application*, Springer-Verlag, Heidelberg, 1993.

D. Bertsekas, and R. Gallager, *Data Networks*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 1992.

J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer, Boston, 1990.

G. Kesidis, *ATM Network Performance*, Kluwer Academic Publishers, 1996.

P. J. B. King, *Computer and Communication Systems Performance Modelling*, Prentice-Hall International (UK), 1990.

D. D. Kouvatsos, *ATM Networks: Performance Modeling and Analysis*, Chapman & Hall, New York, 1996.

K. Kummerle, J. Limb and F. Tobagi (eds.), *Advances in Local Area Networks*, IEEE Press, New York, 1987.

D. A. Menasce and V. A. F. Almeida, *Capacity Planning for Web Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1998.

M. Moshe and R. Rom, *Multiple Access Protocols: Performance and Analysis*, Springer Verlag, 1990.

R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issue*, Artech House, Boston, 1993.

K. Park and W. Willinger, *Self-Similar Network Traffic and Performance Evaluation*, Wiley, New York, 2000.

F. Paul, *Quality of Service: Delivering QOS on the Internet and in Corporate Networks*, Wiley, New York, 1998.

T. G. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*, Springer-Verlag, 2000.

K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer-Verlag, New York, 1995.

M. Schwartz, *Broadband Integrated Networks*, Prentice-Hall, New Jersey, 1996.

H. Takagi (ed.), *Stochastic Analysis of Computer and Communication Systems*, Elsevier Science Publishers / North-Holland, 1990.

J. Walrand and P. Varaiya, *High-Performance Communication Networks*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 2000.

A.2.6 Operations Research

F. H. Hillier and G. J. Lieberman, *Operations Research*, 6th ed., McGraw-Hill, New York, 1995.

J. Moder and S. E. Elmaghraby, *Handbook of Operations Research*, Vols. 1, 2, Van Nostrand-Reinhold, New York, 1978.

B. D. Sivazlian and L. E. Stanfel, *Analysis of Systems in Operations Research*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

A.2.7 Fault-Tolerant Computing

T. Anderson and B. Randell, *Computing Systems Reliability*, Cambridge Univ. Press, New York, 1979.

J. E. Arsenault and J. A. Roberts, *Reliability and Maintainability of Electronic Systems*, Computer Science Press, Potomac, MD, 1980.

D. Avresky (ed.), *Hardware and Software Fault Tolerance in Parallel Computing Systems*, Ellis Horwood, UK, 1992.

P. Jalote, *Fault Tolerance in Distributed Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

J.-C. Laprie (ed.), *Dependability: Basic Concepts and Terminology*, Springer-Verlag, Vienna, 1992.

P. A. Lee, T. and T. Anderson, *Fault Tolerance: Principles and Practice*, Springer-Verlag, Vienna, 1990.

Y.-H. Lee and C. M. Krishna (eds.), *Readings in Real-Time Systems*, IEEE Press, 1993.

M. R. Lyu (ed.), *Software Fault Tolerance*, Wiley, New York, 1995.

S. Osaki and T. Nishio, *Reliability Evaluation of Some Fault-Tolerant Computer Architectures*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1980.

- D. K. Pradhan, *Fault-Tolerant Computer System Design*, Prentice Hall, Englewood Cliffs, NJ, 1996.
- S. Rai and D. P. Agrawal, *Advances in Distributed Computing Network Reliability*, IEEE Press, 1990.
- S. Rai and D. P. Agrawal, *Distributed Computing Network Reliability*, IEEE Press, 1990.
- R. A. Sahner, K. S. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer System: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, 1996.
- D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation*, 3rd, ed., A. K. Peters, Natick, MA, 1998.

A.2.8 Software Reliability

- M. R. Lyu (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, New York, 1995.
- J. D. Musa, *Software Reliability Engineered Testing*(Software Development), McGraw-Hill, New York, 1998.
- J. D. Musa, A. Iannino and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York, 1986.
- M. L. Shooman, *Software Engineering: Reliability, Development and Management*, McGraw-Hill, New York, 1980.
- N. D. Singpurwalla and S. P. Wilson, *Statistical Methods in Software Engineering: Reliability and Risk*, Springer-Verlag, New York, 1999.
- M. Xie, *Software Reliability Modeling*, World Scientific, Singapore, 1991.

A.2.9 Numerical Solutions

- G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins Univ. Press, Baltimore, 1989.
- W. Grassman (ed.), *Computational Probability*, Kluwer Academic Publishers, Amsterdam, 2000.
- C. Meyer and R. J. Plemmons (eds.), *Linear Algebra, Markov Chains, and Queueing Models*, IMA Volumes in Mathematics and its Applications, Vol. 48, Springer-Verlag, Heidelberg, 1993.

M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Dover, New York, 1995.

W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton Univ. Press, Princeton, NJ, 1994.

W. J. Stewart (ed.), *Computation with Markov Chains*, Kluwer Academic Publishers, Boston, 1995.

Appendix B

Properties of Distributions

TABLE B.1. Discrete distributions

Distribution	Parameters	pmf, $p_{\hat{X}}(i)$	PGF	Mean $E[X]$	Variance $Var[X]$
Bernoulli	$p,$ $0 \leq p \leq 1$	$p_{\hat{X}}(0) = p,$ $p_{\hat{X}}(1) = q = 1 - p$	$(1 - p) + pz$	p	$p(1 - p)$
Binomial	$n \geq 1, p,$ $0 \leq p \leq 1$	$\binom{n}{i} p^i (1 - p)^{n-i},$ $i = 0, 1, \dots, n$	$(1 - p + pz)^n$	np	$np(1 - p)$
Geometric	$p,$ $0 < p \leq 1$	$p(1 - p)^{i-1},$ $i = 1, 2, \dots$	$\frac{pz}{1 - (1-p)z} z$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Modified geometric	$p,$ $0 < p \leq 1$	$p(1 - p)^i$	$\frac{p}{1 - (1-p)z} z$	$\frac{1-p}{p}$	$\frac{1-p}{p^2}$
Negative binomial	$0 < p \leq 1,$ $r = 1, 2, \dots$	$\binom{i-1}{r-1} p^r (1 - p)^{i-r},$ $i = r, r+1, \dots$	$[\frac{pz}{1 - (1-p)z}]^r$	$\frac{1}{p}$	$\frac{r(1-p)}{p^2}$
Poisson	$\alpha,$ $\alpha > 0$	$\frac{e^{-\alpha} \alpha^i}{i!},$ $i = 0, 1, 2, \dots$	$e^{-\alpha(1-z)}$	α	α
Uniform	n	$p_X(i) = 1/n,$ $i = 1, 2, \dots, n$	$\frac{1}{n} \sum_{i=1}^n z^i$	$\frac{n+1}{2}$	$\frac{n^2-1}{12}$

Probability and Statistics with Reliability, Queueing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

TABLE B.2. table

Distribution	Parameters	pdf, $f(x)$	Transform ^a	Mean	Variance
Erlang,	$r \geq 1, \lambda > 0$	$\frac{\lambda e^{-\lambda x} (\lambda x)^{r-1}}{(r-1)!}, x > 0$	$\left(\frac{\lambda}{\lambda+s}\right)^r$	$\frac{r}{\lambda}$	$\frac{r}{\lambda^2}$
ERL(λ, r)					
Exponential	$\lambda > 0$	$\lambda e^{-\lambda x}, x > 0$	$\frac{\lambda}{\lambda+s}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
EXP(λ)					
Gamma,	$\alpha > 0, \lambda > 0$	$\frac{\lambda e^{-\lambda x} (\lambda x)^{\alpha-1}}{\Gamma_\alpha}, x > 0$	$\frac{\alpha}{(\lambda+s)^\alpha}$	$\frac{\alpha}{\lambda}$	$\frac{\alpha^2}{\lambda^2}$
GAM(λ, α)					
Hyper-exponential	$\alpha_1, \alpha_2, \dots, \alpha_r$	$\sum_{i=1}^r \alpha_i \lambda_i e^{-\lambda_i x}, x > 0$	$\sum_{i=1}^r \frac{\alpha_i \lambda_i}{\lambda_i + s}$	$\sum_{i=1}^r \frac{\alpha_i}{\lambda_i}$	$2 \sum_{i=1}^r \frac{\alpha_i}{\lambda_i} - \left(\sum_{i=1}^r \frac{\alpha_i}{\lambda_i}\right)^2$
	$\alpha_i \geq 0, \sum_{i=1}^r \alpha_i = 1,$				
	$\lambda_1, \lambda_2, \dots, \lambda_r > 0$				
Hypo-exponential	$\lambda_1, \lambda_2, \dots, \lambda_r > 0$	$\sum_{i=1}^r \alpha_i \lambda_i e^{-\lambda_i x}, x > 0$	$\prod_{i=1}^r \frac{\lambda_i}{\lambda_i + s}$	$\sum_{i=1}^r \frac{1}{\lambda_i}$	$\sum_{i=1}^r \frac{1}{\lambda_i^2}$
HYPO	$(\lambda_1, \lambda_2, \dots, \lambda_r)$	$\lambda_i \neq \lambda_j, i \neq j$	$\alpha_i = \prod_{j \neq i, j=1}^r \frac{\lambda_j}{(\lambda_j - \lambda_i)}$		
Normal	μ, σ^2	$\frac{1}{\sigma \sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$	$N_X(\tau) =$	μ	σ^2

(continued)

TABLE B.2. table

Distribution	Parameters	pdf, $f(x)$	Transform ^a	Mean	Variance
$N(\mu, \sigma^2)$	$-\infty < \mu < \infty,$ $\sigma^2 > 0$	$-\infty < x < \infty$	$e^{i\tau\mu - \frac{\tau^2\sigma^2}{2}}$		
Uniform	a, b $-\infty < a < b < \infty$	$\frac{1}{b-a}, a < x < b$	$\frac{e^{-as} - e^{-bs}}{s(b-a)}$	$\frac{(b-a)^2}{12}$	
Weibull	$\lambda > 0$ $\alpha > 0$	$\lambda \alpha x^{\alpha-1} e^{-\lambda x^\alpha},$ $x > 0$	$\left(\frac{1}{\lambda}\right)^{1/\alpha} \Gamma\left(1 + \frac{1}{\alpha}\right)$	$\left[\left(\frac{1}{\lambda}\right)^{2/\alpha} \Gamma\left(1 + \frac{2}{\alpha}\right)\right]^{-\frac{1}{2}}$	
Log-logistic	λ, κ		$\frac{\pi}{\lambda^\kappa \sin(\pi/\kappa)}$	$\frac{2\pi}{\lambda^2 \kappa \sin(2\pi/\kappa)}$	
			$t > 0$	$\kappa > 1, \tan^{-1}(\lambda^\kappa) \neq \pi$	$\kappa > 2, \tan^{-1}(\lambda^\kappa) \neq \pi$
Pareto	$\alpha, k > 0$	$\alpha k^\alpha x^{-\alpha-1}$ $x \geq k$	$\frac{k^{-\alpha}}{\alpha-1}, \alpha > 1$ $\infty, \alpha \leq 1$	$\frac{k^{2-\alpha}}{\alpha-2} - \left(\frac{k_\alpha}{\alpha-1}\right)^2, \alpha > 2$ $\infty, \alpha \leq 2$	

^a Laplace–Stieltjes transform except for normal distribution

Appendix C

Statistical Tables

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

TABLE C.1. Binomial distribution function

$$B(x; n, p) = \sum_{k=0}^x \binom{n}{k} p^k (1-p)^{n-k}$$

n	x	p									
		0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
2	0	0.9025	0.8100	0.7225	0.6400	0.5625	0.4900	0.4225	0.3600	0.3025	0.2500
	1	0.9975	0.9900	0.9775	0.9600	0.9375	0.9100	0.8775	0.8400	0.7975	0.7500
3	0	0.8574	0.7290	0.6141	0.5120	0.4219	0.3430	0.2746	0.2160	0.1664	0.1250
	1	0.9927	0.9720	0.9393	0.8960	0.8438	0.7840	0.7183	0.6480	0.5748	0.5000
	2	0.9999	0.9990	0.9966	0.9920	0.9844	0.9730	0.9571	0.9360	0.9089	0.8750
4	0	0.8145	0.6561	0.5220	0.4096	0.3164	0.2401	0.1785	0.1296	0.0915	0.0625
	1	0.9860	0.9477	0.8905	0.8192	0.7383	0.6517	0.5630	0.4752	0.3910	0.3125
	2	0.9995	0.9963	0.9880	0.9728	0.9492	0.9163	0.8735	0.8208	0.7585	0.6875
	3	1.0000	0.9999	0.9995	0.9984	0.9961	0.9919	0.9850	0.9744	0.9590	0.9375
5	0	0.7738	0.5905	0.4437	0.3277	0.2373	0.1681	0.1160	0.0778	0.0503	0.0313
	1	0.9774	0.9185	0.8352	0.7373	0.6328	0.5282	0.4284	0.3370	0.2562	0.1875
	2	0.9988	0.9914	0.9734	0.9421	0.8965	0.8369	0.7648	0.6826	0.5931	0.5000
	3	1.0000	0.9995	0.9978	0.9933	0.9844	0.9692	0.9460	0.9130	0.8688	0.8125
	4	1.0000	1.0000	0.9999	0.9997	0.9990	0.9976	0.9947	0.9898	0.9815	0.9687
6	0	0.7351	0.5314	0.3771	0.2621	0.1780	0.1176	0.0754	0.0467	0.0277	0.0156
	1	0.9672	0.8857	0.7765	0.6554	0.5339	0.4202	0.3191	0.2333	0.1636	0.1094
	2	0.9978	0.9842	0.9527	0.9011	0.8306	0.7443	0.6471	0.5443	0.4415	0.3438
	3	0.9999	0.9987	0.9941	0.9830	0.9624	0.9295	0.8826	0.8208	0.7447	0.6563
	4	1.0000	0.9999	0.9996	0.9984	0.9954	0.9891	0.9777	0.9590	0.9308	0.8906
7	5	1.0000	1.0000	1.0000	0.9999	0.9998	0.9993	0.9982	0.9959	0.9917	0.9844
	6	0.6983	0.4783	0.3206	0.2097	0.1335	0.0824	0.0490	0.0280	0.0152	0.0078
8	1	0.9556	0.8503	0.7166	0.5767	0.4449	0.3294	0.2338	0.1586	0.1024	0.0625
	2	0.9962	0.9743	0.9262	0.8520	0.7564	0.6471	0.5323	0.4199	0.3164	0.2266
	3	0.9998	0.9973	0.9879	0.9667	0.9294	0.8740	0.8002	0.7102	0.6083	0.5000
	4	1.0000	0.9998	0.9988	0.9953	0.9871	0.9712	0.9444	0.9037	0.8471	0.7734
	5	1.0000	1.0000	0.9999	0.9996	0.9987	0.9962	0.9910	0.9812	0.9643	0.9375
9	6	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998	0.9994	0.9984	0.9963	0.9922
	0	0.6634	0.4305	0.2725	0.1678	0.1001	0.0576	0.0319	0.0168	0.0084	0.0039
	1	0.9428	0.8131	0.6572	0.5033	0.3671	0.2553	0.1691	0.1064	0.0632	0.0352
	2	0.9942	0.9619	0.8948	0.7969	0.6785	0.5518	0.4278	0.3154	0.2201	0.1445
10	3	0.9996	0.9950	0.9786	0.9437	0.8862	0.8059	0.7064	0.5941	0.4770	0.3633
	4	1.0000	0.9996	0.9971	0.9896	0.9727	0.9420	0.8939	0.8263	0.7396	0.6367
	5	1.0000	1.0000	0.9998	0.9988	0.9958	0.9887	0.9747	0.9502	0.9115	0.8555
	6	1.0000	1.0000	1.0000	0.9999	0.9996	0.9987	0.9964	0.9915	0.9819	0.9648
11	7	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998	0.9993	0.9983	0.9961
	0	0.6302	0.3874	0.2316	0.1342	0.0751	0.0404	0.0207	0.0101	0.0046	0.0020
	1	0.9288	0.7748	0.5995	0.4362	0.3003	0.1960	0.1211	0.0705	0.0385	0.0195
	2	0.9916	0.9470	0.8591	0.7382	0.6007	0.4628	0.3373	0.2318	0.1495	0.0898
	3	0.9994	0.9917	0.9661	0.9144	0.8343	0.7297	0.6089	0.4826	0.3614	0.2539
12	4	1.0000	0.9991	0.9944	0.9804	0.9511	0.9012	0.8283	0.7334	0.6214	0.5000
	5	1.0000	0.9999	0.9994	0.9969	0.9900	0.9747	0.9464	0.9006	0.8342	0.7461
	6	1.0000	1.0000	1.0000	0.9997	0.9987	0.9957	0.9888	0.9750	0.9502	0.9102
	7	1.0000	1.0000	1.0000	1.0000	0.9999	0.9996	0.9986	0.9962	0.9909	0.9805
13	8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997	0.9992	0.9980

Source: Irwin Miller and John E. Freund, *Probability and Statistics for Engineers*, 2nd ed., ©1977, pp. 477–481. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, N.J.

n	x	p									
		0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
10	0	0.5987	0.3487	0.1969	0.1074	0.0563	0.0282	0.0135	0.0060	0.0025	0.0010
	1	0.9139	0.7361	0.5443	0.3758	0.2440	0.1493	0.0860	0.0464	0.0233	0.0107
	2	0.9885	0.9298	0.8202	0.6778	0.5256	0.3828	0.2616	0.1673	0.0996	0.0547
	3	0.9990	0.9872	0.9500	0.8791	0.7759	0.6496	0.5138	0.3823	0.2660	0.1719
	4	0.9999	0.9984	0.9901	0.9672	0.9219	0.8497	0.7515	0.6331	0.5044	0.3770
	5	1.0000	0.9999	0.9986	0.9936	0.9803	0.9527	0.9051	0.8338	0.7384	0.6230
	6	1.0000	1.0000	0.9999	0.9991	0.9965	0.9894	0.9740	0.9452	0.8980	0.8281
	7	1.0000	1.0000	1.0000	0.9999	0.9996	0.9984	0.9952	0.9877	0.9726	0.9453
	8	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9995	0.9983	0.9955	0.9893
	9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997	0.9990
11	0	0.5688	0.3138	0.1673	0.0859	0.0422	0.0198	0.0088	0.0036	0.0014	0.0005
	1	0.8981	0.6974	0.4922	0.3221	0.1971	0.1130	0.0606	0.0302	0.0139	0.0059
	2	0.9848	0.9104	0.7788	0.6174	0.4552	0.3127	0.2001	0.1189	0.0652	0.0327
	3	0.9984	0.9815	0.9306	0.8389	0.7133	0.5696	0.4256	0.2963	0.1911	0.1133
	4	0.9999	0.9972	0.9841	0.9496	0.8854	0.7897	0.6683	0.5328	0.3971	0.2744
	5	1.0000	0.9997	0.9973	0.9883	0.9657	0.9218	0.8513	0.7535	0.6331	0.5000
	6	1.0000	1.0000	0.9997	0.9980	0.9924	0.9784	0.9499	0.9006	0.8262	0.7256
	7	1.0000	1.0000	1.0000	0.9998	0.9988	0.9957	0.9878	0.9707	0.9390	0.8867
	8	1.0000	1.0000	1.0000	1.0000	0.9999	0.9994	0.9980	0.9941	0.9852	0.9673
	9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	0.9993	0.9978	0.9941
12	10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	0.9995
	0	0.5404	0.2824	0.1422	0.0687	0.0317	0.0138	0.0057	0.0022	0.0008	0.0002
	1	0.8816	0.6590	0.4435	0.2749	0.1584	0.0850	0.0424	0.0196	0.0083	0.0032
	2	0.9804	0.8891	0.7358	0.5583	0.3907	0.2528	0.1513	0.0834	0.0421	0.0193
	3	0.9978	0.9744	0.9078	0.7946	0.6488	0.4925	0.3467	0.2253	0.1345	0.0730
	4	0.9998	0.9957	0.9761	0.9274	0.8424	0.7237	0.5833	0.4382	0.3044	0.1938
	5	1.0000	0.9995	0.9954	0.9806	0.9456	0.8822	0.7873	0.6652	0.5269	0.3872
	6	1.0000	0.9999	0.9993	0.9961	0.9857	0.9614	0.9154	0.8418	0.7393	0.6128
	7	1.0000	1.0000	0.9999	0.9994	0.9972	0.9905	0.9745	0.9427	0.8883	0.8062
	8	1.0000	1.0000	1.0000	0.9999	0.9996	0.9983	0.9944	0.9847	0.9644	0.9270
13	9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	0.9997	0.9989	0.9968
	10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998
	11	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9998
	0	0.5133	0.2542	0.1209	0.0550	0.0238	0.0097	0.0037	0.0013	0.0004	0.0001
	1	0.8646	0.6213	0.3983	0.2336	0.1267	0.0637	0.0296	0.0126	0.0049	0.0017
	2	0.9755	0.8661	0.6920	0.5017	0.3326	0.2025	0.1132	0.0579	0.0269	0.0112
	3	0.9969	0.9658	0.8820	0.7473	0.5843	0.4206	0.2783	0.1686	0.0929	0.0461
	4	0.9997	0.9935	0.9658	0.9009	0.7940	0.6543	0.5005	0.3530	0.2279	0.1334
	5	1.0000	0.9991	0.9925	0.9700	0.9198	0.8346	0.7159	0.5744	0.4268	0.2905
	6	1.0000	0.9999	0.9987	0.9930	0.9757	0.9376	0.8705	0.7712	0.6437	0.5000
14	7	1.0000	1.0000	0.9998	0.9988	0.9944	0.9818	0.9538	0.9023	0.8212	0.7095
	8	1.0000	1.0000	1.0000	0.9998	0.9990	0.9960	0.9874	0.9679	0.9302	0.8666
	9	1.0000	1.0000	1.0000	1.0000	0.9999	0.9993	0.9975	0.9922	0.9797	0.9539
10	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997	0.9987	0.9959	0.9888	
11	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9995	0.9983	
12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	
0	0.4877	0.2288	0.1028	0.0440	0.0178	0.0068	0.0024	0.0008	0.0002	0.0001	
1	0.8470	0.5846	0.3567	0.1979	0.1010	0.0475	0.0205	0.0081	0.0029	0.0009	

<i>n</i>	<i>x</i>	<i>p</i>									
		0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
14	2	0.9699	0.8416	0.6479	0.4481	0.2811	0.1608	0.0839	0.0398	0.0170	0.0065
	3	0.9958	0.9559	0.8535	0.6982	0.5213	0.3552	0.2205	0.1243	0.0632	0.0287
	4	0.9996	0.9908	0.9533	0.8702	0.7415	0.5842	0.4227	0.2793	0.1672	0.0898
	5	1.0000	0.9985	0.9885	0.9561	0.8883	0.7805	0.6405	0.4859	0.3373	0.2120
	6	1.0000	0.9998	0.9978	0.9884	0.9617	0.9067	0.8164	0.6925	0.5461	0.3953
	7	1.0000	1.0000	0.9997	0.9976	0.9897	0.9685	0.9247	0.8499	0.7414	0.6047
	8	1.0000	1.0000	1.0000	0.9996	0.9978	0.9917	0.9757	0.9417	0.8811	0.7880
	9	1.0000	1.0000	1.0000	1.0000	0.9997	0.9983	0.9940	0.9825	0.9574	0.9102
	10	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	0.9989	0.9961	0.9886	0.9713
	11	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9994	0.9978	0.9935
	12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997	0.9991
	13	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999
15	0	0.4633	0.2059	0.0874	0.0352	0.0134	0.0047	0.0016	0.0005	0.0001	0.0000
	1	0.8290	0.5490	0.3186	0.1671	0.0802	0.0353	0.0142	0.0052	0.0017	0.0005
	2	0.9638	0.8159	0.6042	0.3980	0.2361	0.1268	0.0617	0.0271	0.0107	0.0037
	3	0.9945	0.9444	0.8227	0.6482	0.4613	0.2969	0.1727	0.0905	0.0424	0.0176
	4	0.9994	0.9873	0.9383	0.8358	0.6865	0.5155	0.3519	0.2173	0.1204	0.0592
	5	0.9999	0.9978	0.9832	0.9389	0.8516	0.7216	0.5643	0.4032	0.2608	0.1509
	6	1.0000	0.9997	0.9964	0.9819	0.9434	0.8689	0.7548	0.6098	0.4522	0.3036
	7	1.0000	1.0000	0.9994	0.9958	0.9827	0.9500	0.8868	0.7869	0.6535	0.5000
	8	1.0000	1.0000	0.9999	0.9992	0.9958	0.9848	0.9578	0.9050	0.8182	0.6964
	9	1.0000	1.0000	1.0000	0.9999	0.9992	0.9963	0.9876	0.9662	0.9231	0.8491
	10	1.0000	1.0000	1.0000	1.0000	0.9999	0.9993	0.9972	0.9907	0.9745	0.9408
	11	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9995	0.9981	0.9937	0.9824
	12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997	0.9989	0.9963
	13	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9995
	14	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
16	0	0.4401	0.1853	0.0743	0.0281	0.0100	0.0033	0.0010	0.0003	0.0001	0.0000
	1	0.8108	0.5147	0.2839	0.1407	0.0635	0.0261	0.0098	0.0033	0.0010	0.0003
	2	0.9571	0.7892	0.5614	0.3518	0.1971	0.0994	0.0451	0.0183	0.0066	0.0021
	3	0.9930	0.9316	0.7899	0.5981	0.4050	0.2459	0.1339	0.0651	0.0281	0.0106
	4	0.9991	0.9830	0.9209	0.7982	0.6302	0.4499	0.2892	0.1666	0.0853	0.0384
	5	0.9999	0.9967	0.9765	0.9183	0.8103	0.6598	0.4900	0.3288	0.1976	0.1051
	6	1.0000	0.9995	0.9944	0.9733	0.9204	0.8247	0.6881	0.5272	0.3660	0.2272
	7	1.0000	0.9999	0.9989	0.9930	0.9729	0.9256	0.8406	0.7161	0.5629	0.4018
	8	1.0000	1.0000	0.9998	0.9985	0.9925	0.9743	0.9329	0.8577	0.7441	0.5982
	9	1.0000	1.0000	1.0000	0.9998	0.9984	0.9929	0.9771	0.9417	0.8759	0.7728
	10	1.0000	1.0000	1.0000	1.0000	0.9997	0.9984	0.9938	0.9809	0.9514	0.8949
	11	1.0000	1.0000	1.0000	1.0000	1.0000	0.9997	0.9987	0.9951	0.9851	0.9616
	12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	0.9991	0.9965	0.9894
	13	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9994	0.9979
	14	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	0.9997
	15	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
17	0	0.4181	0.1668	0.0631	0.0225	0.0075	0.0023	0.0007	0.0002	0.0000	0.0000
	1	0.7922	0.4818	0.2525	0.1182	0.0501	0.0193	0.0067	0.0021	0.0006	0.0001
	2	0.9497	0.7618	0.5198	0.3096	0.1637	0.0774	0.0327	0.0123	0.0041	0.0012
	3	0.9912	0.9174	0.7556	0.5489	0.3530	0.2019	0.1028	0.0464	0.0184	0.0064
	4	0.9988	0.9779	0.9013	0.7582	0.5739	0.3887	0.2348	0.1260	0.0596	0.0245

TABLE C.2. Poisson distribution function

$$F(x; \alpha) = \sum_{k=0}^x e^{-\alpha} \frac{\alpha^k}{k!}$$

$\alpha \backslash x$	0	1	2	3	4	5	6	7	8	9
0.02	0.980	1.000								
0.04	0.961	0.999	1.000							
0.06	0.942	0.998	1.000							
0.08	0.923	0.997	1.000							
0.10	0.905	0.995	1.000							
0.15	0.861	0.990	0.999	1.000						
0.20	0.819	0.982	0.999	1.000						
0.25	0.779	0.974	0.998	1.000						
0.30	0.741	0.963	0.996	1.000						
0.35	0.705	0.951	0.994	1.000						
0.40	0.670	0.938	0.992	0.999	1.000					
0.45	0.638	0.925	0.989	0.999	1.000					
0.50	0.607	0.910	0.986	0.998	1.000					
0.55	0.577	0.894	0.982	0.998	1.000					
0.60	0.549	0.878	0.977	0.997	1.000					
0.65	0.522	0.861	0.972	0.996	0.999	1.000				
0.70	0.497	0.844	0.966	0.994	0.999	1.000				
0.75	0.472	0.827	0.959	0.993	0.999	1.000				
0.80	0.449	0.809	0.953	0.991	0.999	1.000				
0.85	0.427	0.791	0.945	0.989	0.998	1.000				
0.90	0.407	0.772	0.937	0.987	0.998	1.000				
0.95	0.387	0.754	0.929	0.984	0.997	1.000				
1.00	0.368	0.736	0.920	0.981	0.996	0.999	1.000			
1.1	0.333	0.699	0.900	0.974	0.995	0.999	1.000			
1.2	0.301	0.663	0.879	0.966	0.992	0.998	1.000			
1.3	0.273	0.627	0.857	0.957	0.989	0.998	1.000			
1.4	0.247	0.592	0.833	0.946	0.986	0.997	0.999	1.000		
1.5	0.223	0.558	0.809	0.934	0.981	0.996	0.999	1.000		
1.6	0.202	0.525	0.783	0.921	0.976	0.994	0.999	1.000		
1.7	0.183	0.493	0.757	0.907	0.970	0.992	0.998	1.000		
1.8	0.165	0.463	0.731	0.891	0.964	0.990	0.997	0.999	1.000	
1.9	0.150	0.434	0.704	0.875	0.956	0.987	0.997	0.999	1.000	
2.0	0.135	0.406	0.677	0.857	0.947	0.983	0.995	0.999	1.000	

Source: Reprinted by kind permission from E. C. Molina, *Poisson's Exponential Binomial Limit*, D. Van Nostrand Company, Inc., Princeton, NJ, 1974.

$\alpha \backslash x$	0	1	2	3	4	5	6	7	8	9
2.2	0.111	0.355	0.623	0.819	0.928	0.975	0.993	0.998	1.000	
2.4	0.091	0.308	0.570	0.779	0.904	0.964	0.988	0.997	0.999	1.000
2.6	0.074	0.267	0.518	0.736	0.877	0.951	0.983	0.995	0.999	1.000
2.8	0.061	0.231	0.469	0.692	0.848	0.935	0.976	0.992	0.998	0.999
3.0	0.050	0.199	0.423	0.647	0.815	0.916	0.966	0.988	0.996	0.999
3.2	0.041	0.171	0.380	0.603	0.781	0.895	0.955	0.983	0.994	0.998
3.4	0.033	0.147	0.340	0.558	0.744	0.871	0.942	0.977	0.992	0.997
3.6	0.027	0.126	0.303	0.515	0.706	0.844	0.927	0.969	0.988	0.996
3.8	0.022	0.107	0.269	0.473	0.668	0.816	0.909	0.960	0.984	0.994
4.0	0.018	0.092	0.238	0.433	0.629	0.785	0.889	0.949	0.979	0.992
4.2	0.015	0.078	0.210	0.395	0.590	0.753	0.867	0.936	0.972	0.989
4.4	0.012	0.066	0.185	0.359	0.551	0.720	0.844	0.921	0.964	0.985
4.6	0.010	0.056	0.163	0.326	0.513	0.686	0.818	0.905	0.955	0.980
4.8	0.008	0.048	0.143	0.294	0.476	0.651	0.791	0.887	0.944	0.975
5.0	0.007	0.040	0.125	0.265	0.440	0.616	0.762	0.867	0.932	0.968
5.2	0.006	0.034	0.109	0.238	0.406	0.581	0.732	0.845	0.918	0.960
5.4	0.005	0.029	0.095	0.213	0.373	0.546	0.702	0.822	0.903	0.951
5.6	0.004	0.024	0.082	0.191	0.342	0.512	0.670	0.797	0.886	0.941
5.8	0.003	0.021	0.072	0.170	0.313	0.478	0.638	0.771	0.867	0.929
6.0	0.002	0.017	0.062	0.151	0.285	0.446	0.606	0.744	0.847	0.916
	10	11	12	13	14	15	16			
2.8	1.000									
3.0	1.000									
3.2	1.000									
3.4	0.999	1.000								
3.6	0.999	1.000								
3.8	0.998	0.999	1.000							
4.0	0.997	0.999	1.000							
4.2	0.996	0.999	1.000							
4.4	0.994	0.998	0.999	1.000						
4.6	0.992	0.997	0.999	1.000						
4.8	0.990	0.996	0.999	1.000						
5.0	0.986	0.995	0.998	0.999	1.000					
5.2	0.982	0.993	0.997	0.999	1.000					
5.4	0.977	0.990	0.996	0.999	1.000					
5.6	0.972	0.988	0.995	0.998	0.999	1.000				
5.8	0.965	0.984	0.993	0.997	0.999	1.000				
6.0	0.957	0.980	0.991	0.996	0.999	0.999	1.000			

$\alpha \backslash x$	0	1	2	3	4	5	6	7	8	9
6.2	0.002	0.015	0.054	0.134	0.259	0.414	0.574	0.716	0.826	0.902
6.4	0.002	0.012	0.046	0.119	0.235	0.384	0.542	0.687	0.803	0.886
6.6	0.001	0.010	0.040	0.105	0.213	0.355	0.511	0.658	0.780	0.869
6.8	0.001	0.009	0.034	0.093	0.192	0.327	0.480	0.628	0.755	0.850
7.0	0.001	0.007	0.030	0.082	0.173	0.301	0.450	0.599	0.729	0.830
7.2	0.001	0.006	0.025	0.072	0.156	0.276	0.420	0.569	0.703	0.810
7.4	0.001	0.005	0.022	0.063	0.140	0.253	0.392	0.539	0.676	0.788
7.6	0.001	0.004	0.019	0.055	0.125	0.231	0.365	0.510	0.648	0.765
7.8	0.000	0.004	0.016	0.048	0.112	0.210	0.338	0.481	0.620	0.741
8.0	0.000	0.003	0.014	0.042	0.100	0.191	0.313	0.453	0.593	0.717
8.5	0.000	0.002	0.009	0.030	0.074	0.150	0.256	0.386	0.523	0.653
9.0	0.000	0.001	0.006	0.021	0.055	0.116	0.207	0.324	0.456	0.587
9.5	0.000	0.001	0.004	0.015	0.040	0.089	0.165	0.269	0.392	0.522
10.0	0.000	0.000	0.003	0.010	0.029	0.067	0.130	0.220	0.333	0.458
	10	11	12	13	14	15	16	17	18	19
6.2	0.949	0.975	0.989	0.995	0.998	0.999	1.000			
6.4	0.939	0.969	0.986	0.994	0.997	0.999	1.000			
6.6	0.927	0.963	0.982	0.992	0.997	0.999	0.999	1.000		
6.8	0.915	0.955	0.978	0.990	0.996	0.998	0.999	1.000		
7.0	0.901	0.947	0.973	0.987	0.994	0.998	0.999	1.000		
7.2	0.887	0.937	0.967	0.984	0.993	0.997	0.999	1.000		
7.4	0.871	0.926	0.961	0.980	0.991	0.996	0.998	0.999	1.000	
7.6	0.854	0.915	0.954	0.976	0.989	0.995	0.998	0.999	1.000	
7.8	0.835	0.902	0.945	0.971	0.986	0.993	0.997	0.999	1.000	
8.0	0.816	0.888	0.936	0.966	0.983	0.992	0.996	0.998	0.999	1.000
8.5	0.763	0.849	0.909	0.949	0.973	0.986	0.993	0.997	0.999	0.999
9.0	0.706	0.803	0.876	0.926	0.959	0.978	0.989	0.995	0.998	0.999
9.5	0.645	0.752	0.836	0.898	0.940	0.967	0.982	0.991	0.996	0.998
10.0	0.583	0.697	0.792	0.864	0.917	0.951	0.973	0.986	0.993	0.997
	20	21	22							
8.5	1.000									
9.0	1.000									
9.5	0.999	1.000								
10.0	0.998	0.999	1.000							

TABLE C.3. Distribution function of standard normal random variable

<i>z</i>	0	1	2	3	4	5	6	7	8	9
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.5	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.6	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9990	.9993	.9995	.9997	.9998	.9998	.9999	.9999	1.0000

Notes: (1) if a normal variable X is not "standard," its value must be standardized: $Z = (X - \mu)/\sigma$; then $F_X(x) = F_Z(\frac{x-\mu}{\sigma})$; (2) for $z \geq 4$, use $F_Z(z) = 1$ to four decimal places; for $z \leq -4$, use $F_Z(z) = 0$ to four decimal places; (3) the entries opposite $z = 3$ are for 3.0, 3.1, 3.2, and so on; (4) for $z < 0$, use $F_z(z) = 1 - F_z(-z)$.

Source: Reprinted by permission from B. W. Lindgren and G. W. McElrath, *Introduction to Probability and Statistics*, 2nd ed. (copyright ©1966 by B. W. Lindgren and G. W. McElrath, published by Macmillan Publishing Co.).

TABLE C.4. Critical values of the Student's *t*-distribution

α df ^a	0.1	0.05	0.025	0.01	0.005	0.001
1	3.078	6.314	12.706	31.821	63.657	318.309
2	1.886	2.920	4.303	6.965	9.925	22.327
3	1.638	2.353	3.182	4.541	5.841	10.215
4	1.533	2.132	2.776	3.747	4.604	7.173
5	1.476	2.015	2.571	3.365	4.032	5.893
6	1.440	1.943	2.447	3.143	3.707	5.208
7	1.415	1.895	2.365	2.998	3.499	4.785
8	1.397	1.860	2.306	2.896	3.355	4.501
9	1.383	1.833	2.262	2.821	3.250	4.297
10	1.372	1.812	2.228	2.764	3.169	4.144
11	1.363	1.796	2.201	2.718	3.106	4.025
12	1.356	1.782	2.179	2.681	3.055	3.930
13	1.350	1.771	2.160	2.650	3.012	3.852
14	1.345	1.761	2.145	2.624	2.977	3.787
15	1.341	1.753	2.131	2.602	2.947	3.733
16	1.337	1.746	2.120	2.583	2.921	3.686
17	1.333	1.740	2.110	2.567	2.898	3.646
18	1.330	1.734	2.101	2.552	2.878	3.610
19	1.328	1.729	2.093	2.539	2.861	3.579
20	1.325	1.725	2.086	2.528	2.845	3.552
21	1.323	1.721	2.080	2.518	2.831	3.527
22	1.321	1.717	2.074	2.508	2.819	3.505
23	1.319	1.714	2.069	2.500	2.807	3.485
24	1.318	1.711	2.064	2.492	2.797	3.467
25	1.316	1.708	2.060	2.485	2.787	3.450
26	1.315	1.706	2.056	2.479	2.779	3.435
27	1.314	1.703	2.052	2.473	2.771	3.421
28	1.313	1.701	2.048	2.467	2.763	3.408
29	1.311	1.699	2.045	2.462	2.756	3.396
30	1.310	1.697	2.042	2.457	2.750	3.385
40	1.303	1.684	2.021	2.423	2.704	3.307
60	1.296	1.671	2.000	2.390	2.660	3.232
120	1.289	1.658	1.980	2.358	2.617	3.160
∞	1.282	1.645	1.960	2.326	2.576	3.090

^aDegree of freedom.

TABLE C.5. Critical values of χ^2 distribution

$P \backslash v$	0.995	0.975	0.050	0.025	0.010	0.005
1	0.0 ⁴ 3927	0.0 ³ 9821	3.84146	5.02389	6.63490	7.87944
2	0.010025	0.050636	5.99146	7.37776	9.21034	10.5966
3	0.071722	0.215795	7.81473	9.34840	11.3449	12.8382
4	0.206989	0.484419	9.48773	11.1433	13.2767	14.8603
5	0.411742	0.831212	11.0705	12.8325	15.0863	16.7496
6	0.675727	1.237344	12.5916	14.4494	16.8119	18.5476
7	0.989256	1.689869	14.0671	16.0128	18.4753	20.2777
8	1.344413	2.17973	15.5073	17.5345	20.0902	21.9550
9	1.734933	2.70039	16.9190	19.0228	21.6660	23.5894
10	2.15586	3.24697	18.3070	20.4832	23.2093	25.1882
11	2.60322	3.81575	19.6751	21.9200	24.7250	26.7568
12	3.07382	4.40379	21.0261	23.3367	26.2170	28.2995
13	3.56503	5.00875	22.3620	24.7356	27.6882	29.8195
14	4.07467	5.62873	23.6848	26.1189	29.1412	31.3193
15	4.60092	6.26214	24.9958	27.4884	30.5779	32.8013
16	5.14221	6.90766	26.2962	28.8454	31.9999	34.2672
17	5.69722	7.56419	27.5871	30.1910	33.4087	35.7185
18	6.26480	8.23075	28.8693	31.5264	34.8053	37.1565
19	6.84397	8.90652	30.1435	32.8523	36.1909	38.5823
20	7.43384	9.59078	31.4104	34.1696	37.5662	39.9968
21	8.03365	10.2829	32.6706	35.4789	38.9322	41.4011
22	8.64272	10.9823	33.9244	36.7807	40.2894	42.7957
23	9.26042	11.6886	35.1725	38.0756	41.6384	44.1813
24	9.88623	12.4012	36.4150	39.3641	42.9798	45.5585
25	10.5197	13.1197	37.6525	40.6465	44.3141	46.9279
26	11.1602	13.8439	38.8851	41.9232	45.6417	48.2899
27	11.8076	14.5734	40.1133	43.1945	46.9629	49.6449
28	12.4613	15.3079	41.3371	44.4608	48.2782	50.9934
29	13.1211	16.0471	42.5570	45.7223	49.5879	52.3356
30	13.7867	16.7908	43.7730	46.9792	50.8922	53.6720
40	20.7065	24.4330	55.7585	59.3417	63.6907	66.7660
50	27.9907	32.3574	67.5048	71.4202	76.1539	79.4900
60	35.5345	40.4817	79.0819	83.2977	88.3794	91.9517
70	43.2752	48.7576	90.5312	95.0232	100.425	104.215
80	51.1719	57.1532	101.879	106.629	112.329	116.321
90	59.1963	65.6466	113.145	118.136	124.116	128.299
100	67.3276	74.2219	124.342	129.561	135.807	140.169

Note: The first column lists the number of degrees of freedom (v). The headings of the other columns give probabilities (P) for χ^2 to exceed the entry value. For $v > 100$, treat $\sqrt{2\chi^2} - \sqrt{2v-1}$ as a standard normal variable.

Source: Reprinted by kind permission from D. A. Fraser, *Statistics, An Introduction*, Wiley, New York, 1958.

TABLE C.6. Critical values of the F distribution

5% (lightface type) and 1% (boldface type) points for the distribution of F

		Degrees of freedom for numerator (v_1)																						
		Degrees of freedom for denominator (v_2)																						
		2	3	4	5	6	7	8	9	10	11	12	14	16	20	24	30	40	50	75	100	200	500	∞
1	161	200	216	225	230	234	237	239	241	242	243	244	245	246	248	249	250	251	252	253	254	254	254	254
	4052	4999	5403	5403	5625	5764	5859	5928	5981	6022	6096	6082	6106	6142	6169	6208	6284	6286	6302	6323	6334	6352	6361	6366
2	18.51	19.00	19.16	19.25	19.30	19.36	19.37	19.39	19.40	19.41	19.42	19.43	19.44	19.45	19.46	19.47	19.48	19.49	19.49	19.49	19.50	19.50	19.50	
	98.49	99.01	99.17	99.25	99.30	99.33	99.34	99.36	99.38	99.40	99.42	99.43	99.44	99.45	99.46	99.47	99.48	99.49	99.49	99.49	99.50	99.50	99.50	
3	10.13	9.55	9.28	9.12	9.01	8.94	8.88	8.84	8.81	8.78	8.76	8.74	8.71	8.69	8.66	8.64	8.62	8.60	8.58	8.57	8.56	8.54	8.54	8.53
	34.12	30.81	29.46	28.71	28.24	27.91	27.67	27.49	27.34	27.23	27.13	27.05	26.92	26.83	26.69	26.60	26.50	26.41	26.30	26.27	26.23	26.18	26.14	26.12
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.93	5.91	5.87	5.84	5.80	5.77	5.74	5.71	5.70	5.68	5.66	5.65	5.64	5.63
	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.54	14.45	14.37	14.24	14.15	14.02	13.93	13.83	13.74	13.69	13.61	13.57	13.52	13.48	13.46
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.78	4.74	4.70	4.68	4.64	4.60	4.56	4.53	4.50	4.46	4.44	4.42	4.40	4.38	4.37	4.36
	16.26	13.27	12.06	11.39	10.97	10.67	10.45	10.27	10.15	10.05	9.96	9.89	9.77	9.68	9.55	9.47	9.38	9.29	9.24	9.17	9.13	9.07	9.04	9.02
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.03	4.00	3.96	3.92	3.87	3.84	3.81	3.77	3.75	3.72	3.71	3.69	3.68	3.67
	13.74	10.92	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.79	7.72	7.60	7.52	7.39	7.31	7.23	7.14	7.09	7.02	6.99	6.94	6.90	6.88	
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.63	3.60	3.57	3.52	3.49	3.44	3.41	3.38	3.34	3.32	3.29	3.28	3.25	3.24	3.23
	12.25	9.55	8.45	7.85	7.46	7.19	7.00	6.84	6.71	6.62	6.54	6.47	6.35	6.27	6.15	6.07	5.98	5.90	5.85	5.78	5.75	5.70	5.67	5.66
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.34	3.31	3.28	3.23	3.20	3.15	3.12	3.08	3.05	3.03	3.00	2.98	2.96	2.94	2.93
	11.26	8.65	7.59	7.01	6.63	6.37	6.19	6.03	5.91	5.82	5.74	5.67	5.56	5.48	5.36	5.28	5.20	5.11	5.06	5.00	4.96	4.91	4.88	4.86

Source: Reprinted by permission from G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 7th ed., ©1980 by The Iowa State University Press, Ames, Iowa.

9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.18	3.13	3.10	3.07	3.02	2.98	2.93	2.90	2.86	2.82	2.80	2.77	2.76	2.73	2.72	2.71	
10	10.56	8.02	6.99	6.42	6.06	5.80	5.62	5.47	5.35	5.26	5.18	5.11	5.00	4.92	4.80	4.73	4.64	4.56	4.51	4.45	4.41	4.36	4.33	4.31
11	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.97	2.94	2.91	2.86	2.82	2.77	2.74	2.70	2.67	2.64	2.61	2.59	2.56	2.55	
12	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.86	2.82	2.79	2.74	2.70	2.65	2.61	2.57	2.53	2.50	2.47	2.45	2.42	2.41	
13	9.65	7.20	6.22	5.67	5.32	5.07	4.88	4.74	4.63	4.54	4.46	4.40	4.29	4.21	4.10	4.02	3.94	3.86	3.80	3.74	3.70	3.66	3.62	3.60
14	4.75	3.88	3.49	3.26	3.11	3.00	2.92	2.85	2.80	2.76	2.72	2.69	2.64	2.60	2.54	2.50	2.46	2.42	2.40	2.36	2.35	2.32	2.31	
15	4.67	3.80	3.41	3.18	3.02	2.92	2.84	2.77	2.72	2.67	2.63	2.60	2.55	2.51	2.46	2.42	2.38	2.34	2.32	2.28	2.26	2.24	2.22	
16	9.07	6.70	5.74	5.20	4.86	4.62	4.44	4.30	4.19	4.10	4.02	3.96	3.85	3.78	3.67	3.59	3.51	3.42	3.37	3.30	3.27	3.21	3.18	3.16
17	4.60	3.74	3.34	3.11	2.96	2.85	2.77	2.70	2.65	2.60	2.56	2.53	2.48	2.44	2.40	2.35	2.31	2.27	2.24	2.21	2.19	2.16	2.14	
18	8.86	6.51	5.56	5.03	4.69	4.46	4.28	4.14	4.03	3.94	3.86	3.80	3.70	3.62	3.51	3.43	3.34	3.26	3.21	3.14	3.11	3.06	3.02	3.00
19	4.54	3.68	3.29	3.06	2.90	2.79	2.70	2.64	2.59	2.55	2.51	2.48	2.43	2.39	2.33	2.29	2.25	2.21	2.18	2.15	2.12	2.08	2.07	
20	8.68	6.36	5.42	4.89	4.56	4.32	4.14	4.00	3.89	3.80	3.73	3.67	3.56	3.48	3.36	3.29	3.20	3.12	3.07	3.00	2.97	2.92	2.89	2.87
21	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.45	2.42	2.37	2.33	2.28	2.24	2.20	2.16	2.13	2.09	2.07	2.04	2.02	
22	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.61	3.55	3.45	3.37	3.25	3.18	3.10	3.01	2.96	2.89	2.86	2.80	2.77	2.75
23	4.45	3.59	3.20	2.96	2.81	2.70	2.62	2.55	2.50	2.45	2.41	2.38	2.33	2.29	2.23	2.19	2.15	2.11	2.08	2.04	2.02	1.99	1.97	
24	8.40	6.11	5.18	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.52	3.45	3.35	3.27	3.16	3.08	3.00	2.92	2.86	2.79	2.76	2.70	2.67	2.65
25	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.37	2.34	2.30	2.29	2.25	2.19	2.15	2.11	2.07	2.04	2.00	1.98	1.95	
26	8.28	6.01	5.09	4.58	4.25	4.01	3.85	3.71	3.60	3.51	3.44	3.37	3.27	3.19	3.07	3.00	2.91	2.83	2.78	2.71	2.68	2.62	2.59	2.57
27	4.38	3.52	3.13	2.90	2.74	2.63	2.55	2.48	2.43	2.38	2.34	2.31	2.26	2.21	2.15	2.11	2.07	2.02	2.00	1.96	1.94	1.91	1.88	
28	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.36	3.30	3.19	3.12	3.00	2.92	2.84	2.76	2.70	2.63	2.60	2.54	2.51	2.49
29	4.35	3.49	3.10	2.87	2.71	2.60	2.52	2.45	2.40	2.35	2.31	2.28	2.23	2.18	2.12	2.08	2.04	1.99	1.96	1.92	1.90	1.87	1.85	
30	8.10	5.85	4.94	4.43	4.10	3.87	3.71	3.56	3.45	3.37	3.30	3.23	3.13	3.05	2.94	2.86	2.77	2.69	2.63	2.56	2.53	2.47	2.44	2.42
31	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.28	2.25	2.20	2.15	2.10	2.04	2.00	1.96	1.93	1.89	1.87	1.84	1.82	
32	8.02	5.78	4.87	4.37	4.04	3.81	3.65	3.51	3.40	3.31	3.24	3.17	3.07	2.99	2.88	2.80	2.72	2.63	2.58	2.51	2.47	2.42	2.38	2.36
33	4.30	3.44	3.05	2.82	2.66	2.55	2.47	2.40	2.35	2.30	2.26	2.23	2.18	2.13	2.07	2.03	1.98	1.93	1.91	1.87	1.84	1.81	1.78	
34	7.94	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.18	3.12	3.02	2.94	2.83	2.75	2.67	2.58	2.53	2.46	2.42	2.37	2.33	2.31
35	4.28	3.42	3.03	2.80	2.64	2.53	2.45	2.38	2.32	2.28	2.24	2.20	2.14	2.10	2.04	2.00	1.96	1.91	1.88	1.84	1.82	1.77	1.76	
36	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.14	3.07	2.97	2.89	2.78	2.70	2.62	2.53	2.48	2.41	2.37	2.32	2.28	2.26
37	4.26	3.40	3.01	2.78	2.62	2.51	2.43	2.36	2.30	2.26	2.22	2.18	2.13	2.09	2.02	1.98	1.94	1.89	1.86	1.82	1.80	1.76	1.74	
38	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.25	3.17	3.09	3.03	2.93	2.85	2.74	2.66	2.58	2.49	2.44	2.36	2.33	2.27	2.23	2.21
39	4.24	3.38	2.99	2.76	2.60	2.49	2.41	2.34	2.28	2.24	2.20	2.16	2.11	2.06	2.00	1.96	1.92	1.87	1.84	1.80	1.77	1.74	1.72	
40	7.77	5.57	4.68	4.18	3.86	3.63	3.46	3.32	3.21	3.13	3.05	2.99	2.89	2.81	2.70	2.62	2.54	2.45	2.40	2.32	2.29	2.23	2.19	2.17

Degrees of freedom for denominator ₁		Degrees of freedom for numerator (v ₁)																						
		2	3	4	5	6	7	8	9	10	11	12	14	16	20	24	30	40	50	75	100	200	500	∞
26	4.22	3.37	2.89	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.18	2.15	2.10	2.05	1.99	1.95	1.90	1.85	1.82	1.78	1.76	1.72	1.70	1.69
27	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.17	3.09	3.02	2.96	2.86	2.77	2.66	2.68	2.60	2.41	2.36	2.28	2.25	2.19	2.15	2.13
28	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.30	2.25	2.20	2.13	2.10	2.08	2.03	1.97	1.93	1.88	1.84	1.80	1.76	1.74	1.71	1.68	1.67
29	7.68	5.49	4.60	4.11	3.79	3.56	3.39	3.26	3.14	3.06	2.98	2.93	2.83	2.74	2.63	2.55	2.47	2.38	2.33	2.26	2.21	2.16	2.12	2.10
30	4.20	3.34	2.95	2.71	2.56	2.44	2.36	2.29	2.24	2.19	2.15	2.12	2.06	2.02	1.96	1.91	1.87	1.81	1.78	1.75	1.72	1.69	1.67	1.65
31	7.64	5.45	4.57	4.07	3.76	3.53	3.36	3.23	3.11	3.03	2.95	2.90	2.80	2.71	2.60	2.52	2.44	2.35	2.30	2.22	2.18	2.13	2.09	2.06
32	4.18	3.33	2.93	2.70	2.54	2.43	2.35	2.28	2.22	2.18	2.14	2.10	2.05	2.00	1.94	1.90	1.85	1.80	1.77	1.73	1.71	1.68	1.65	1.64
33	7.60	5.52	4.54	4.04	3.73	3.50	3.33	3.20	3.08	3.00	2.92	2.87	2.77	2.68	2.57	2.49	2.41	2.32	2.27	2.19	2.15	2.10	2.06	2.03
34	4.17	3.32	2.92	2.69	2.53	2.42	2.34	2.27	2.21	2.16	2.12	2.09	2.04	1.99	1.93	1.89	1.84	1.79	1.76	1.72	1.69	1.66	1.64	1.62
35	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.06	2.98	2.90	2.84	2.74	2.66	2.55	2.47	2.38	2.29	2.24	2.16	2.13	2.07	2.03	2.01
36	4.15	3.30	2.90	2.67	2.51	2.40	2.32	2.25	2.19	2.14	2.10	2.07	2.02	1.97	1.91	1.86	1.82	1.76	1.74	1.69	1.67	1.64	1.61	1.59
37	7.50	5.34	4.46	3.97	3.66	3.42	3.25	3.12	3.01	2.94	2.86	2.80	2.70	2.62	2.51	2.42	2.34	2.25	2.20	2.12	2.08	2.02	1.98	1.96
38	4.13	3.28	2.88	2.65	2.49	2.38	2.30	2.23	2.17	2.12	2.08	2.05	2.00	1.95	1.89	1.84	1.80	1.74	1.71	1.67	1.64	1.61	1.59	1.57
39	7.44	5.29	4.42	3.93	3.61	3.38	3.21	3.08	2.97	2.89	2.82	2.76	2.66	2.58	2.47	2.38	2.30	2.21	2.15	2.08	2.04	1.98	1.94	1.91
40	4.08	3.23	2.84	2.61	2.45	2.34	2.26	2.21	2.15	2.10	2.06	2.03	1.99	1.93	1.87	1.82	1.78	1.72	1.69	1.65	1.62	1.59	1.56	1.55
41	7.39	5.25	4.38	3.89	3.58	3.35	3.18	3.04	2.94	2.86	2.78	2.72	2.62	2.54	2.43	2.35	2.26	2.17	2.12	2.04	2.00	1.94	1.90	1.87
42	4.10	3.25	2.85	2.62	2.46	2.35	2.26	2.19	2.14	2.09	2.05	2.02	1.96	1.92	1.85	1.80	1.76	1.71	1.67	1.63	1.60	1.57	1.54	1.53
43	7.35	5.21	4.34	3.86	3.54	3.32	3.15	3.02	2.91	2.82	2.75	2.69	2.59	2.51	2.40	2.32	2.22	2.14	2.08	2.00	1.97	1.90	1.86	1.84
44	4.08	3.23	2.84	2.61	2.45	2.34	2.26	2.21	2.15	2.10	2.06	2.03	1.99	1.93	1.87	1.82	1.78	1.74	1.69	1.66	1.63	1.58	1.56	1.55
45	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.88	2.80	2.73	2.66	2.56	2.49	2.37	2.29	2.20	2.11	2.05	1.97	1.94	1.88	1.84	1.81
46	4.07	3.22	2.83	2.59	2.44	2.32	2.24	2.17	2.11	2.06	2.02	1.99	1.94	1.89	1.82	1.78	1.73	1.68	1.64	1.60	1.57	1.54	1.51	1.49
47	7.27	5.15	4.29	3.80	3.49	3.26	3.10	2.96	2.86	2.77	2.70	2.64	2.54	2.46	2.35	2.26	2.17	2.08	2.02	1.94	1.91	1.85	1.80	1.78
48	4.06	3.21	2.82	2.58	2.43	2.31	2.23	2.16	2.10	2.05	2.01	1.98	1.92	1.88	1.81	1.76	1.72	1.66	1.63	1.60	1.58	1.55	1.52	1.50
49	7.24	5.12	4.26	3.78	3.46	3.24	3.07	2.94	2.84	2.75	2.68	2.62	2.52	2.44	2.32	2.24	2.15	2.06	2.00	1.92	1.88	1.82	1.78	1.75

46	4.05	3.20	2.81	2.57	2.42	2.30	2.22	2.14	2.09	2.04	2.00	1.97	1.91	1.87	1.80	1.75	1.71	1.65	1.62	1.57	1.54	1.51	1.48	1.46	
	7.21	5.10	4.24	3.76	3.44	3.22	3.05	2.92	2.82	2.73	2.66	2.60	2.50	2.42	2.30	2.22	2.13	2.04	1.98	1.90	1.86	1.80	1.76	1.72	
48	4.04	3.19	2.80	2.56	2.41	2.30	2.21	2.14	2.08	2.03	1.99	1.96	1.90	1.86	1.79	1.74	1.70	1.64	1.61	1.56	1.53	1.50	1.47	1.45	
	7.19	5.08	4.22	3.74	3.42	3.20	3.04	2.90	2.80	2.71	2.64	2.58	2.48	2.40	2.28	2.20	2.11	2.02	1.96	1.88	1.84	1.78	1.73	1.70	
50	4.03	3.18	2.79	2.56	2.40	2.29	2.20	2.13	2.07	2.02	1.98	1.95	1.90	1.85	1.78	1.74	1.69	1.63	1.60	1.55	1.52	1.48	1.46	1.44	
	7.17	5.06	4.20	3.72	3.41	3.18	3.02	2.88	2.78	2.70	2.62	2.56	2.46	2.39	2.26	2.18	2.10	2.00	1.94	1.86	1.82	1.76	1.71	1.68	
55	4.02	3.17	2.78	2.54	2.38	2.27	2.18	2.11	2.05	2.00	1.97	1.93	1.88	1.83	1.76	1.72	1.67	1.61	1.58	1.52	1.50	1.46	1.43	1.41	
	7.12	5.01	4.16	3.68	3.37	3.15	2.98	2.85	2.75	2.66	2.69	2.63	2.43	2.35	2.23	2.15	2.06	1.96	1.90	1.82	1.78	1.71	1.66	1.64	
60	4.00	3.15	2.76	2.52	2.37	2.25	2.17	2.10	2.04	1.99	1.95	1.92	1.86	1.81	1.75	1.70	1.65	1.59	1.56	1.50	1.48	1.44	1.41	1.39	
	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.56	2.50	2.40	2.32	2.20	2.12	2.03	1.93	1.87	1.79	1.74	1.68	1.63	1.60	
65	3.99	3.14	2.75	2.51	2.36	2.24	2.15	2.08	2.02	1.98	1.94	1.90	1.85	1.80	1.73	1.68	1.63	1.57	1.54	1.49	1.46	1.42	1.39	1.37	
	7.04	4.96	4.10	3.62	3.31	3.09	2.93	2.79	2.70	2.61	2.54	2.47	2.37	2.30	2.18	2.09	2.00	1.90	1.84	1.76	1.71	1.64	1.60	1.56	
70	3.98	3.13	2.74	2.50	2.35	2.32	2.14	2.07	2.01	1.97	1.93	1.89	1.84	1.79	1.74	1.72	1.67	1.62	1.56	1.53	1.47	1.45	1.40	1.37	1.35
	7.01	4.92	4.08	3.60	3.29	3.07	2.91	2.77	2.67	2.59	2.51	2.45	2.35	2.28	2.15	2.07	1.98	1.88	1.82	1.74	1.69	1.62	1.56	1.53	
80	3.96	3.11	2.72	2.48	2.33	2.21	2.12	2.05	1.99	1.95	1.91	1.88	1.82	1.77	1.70	1.65	1.60	1.54	1.51	1.45	1.42	1.38	1.35	1.32	
	6.96	4.88	4.04	3.56	3.25	3.04	2.87	2.74	2.64	2.55	2.48	2.41	2.32	2.24	2.11	2.03	1.94	1.84	1.78	1.70	1.65	1.57	1.52	1.49	
100	3.94	3.09	2.70	2.46	2.30	2.19	2.10	2.03	1.97	1.92	1.88	1.85	1.79	1.75	1.68	1.63	1.57	1.51	1.48	1.42	1.39	1.34	1.30	1.28	
	6.90	4.82	3.98	3.51	3.20	2.99	2.82	2.69	2.59	2.51	2.43	2.36	2.26	2.19	2.06	1.98	1.89	1.79	1.73	1.64	1.59	1.51	1.46	1.43	
125	3.92	3.07	2.68	2.44	2.29	2.17	2.08	2.01	1.95	1.90	1.86	1.83	1.77	1.72	1.65	1.60	1.55	1.49	1.45	1.40	1.37	1.31	1.27	1.25	
	6.84	4.78	3.94	3.47	3.17	2.95	2.79	2.65	2.56	2.47	2.40	2.33	2.23	2.15	2.03	1.94	1.85	1.75	1.68	1.62	1.59	1.54	1.46	1.40	1.37
150	3.91	3.06	2.67	2.43	2.27	2.16	2.07	2.00	1.94	1.89	1.85	1.82	1.76	1.71	1.64	1.59	1.54	1.47	1.44	1.37	1.34	1.29	1.25	1.22	
	6.81	4.75	3.91	3.44	3.13	2.92	2.76	2.62	2.53	2.44	2.37	2.30	2.20	2.12	2.00	1.91	1.83	1.72	1.66	1.61	1.51	1.43	1.37	1.33	
200	3.89	3.04	2.65	2.41	2.26	2.14	2.05	1.98	1.92	1.87	1.83	1.80	1.74	1.69	1.62	1.57	1.52	1.45	1.42	1.35	1.32	1.26	1.22	1.19	
	6.76	4.71	3.88	3.41	3.11	2.90	2.73	2.60	2.50	2.41	2.34	2.28	2.17	2.09	1.97	1.88	1.79	1.69	1.62	1.53	1.48	1.39	1.33	1.28	
400	3.86	3.02	2.62	2.39	2.23	2.12	2.03	1.96	1.90	1.85	1.81	1.78	1.72	1.67	1.60	1.54	1.49	1.42	1.38	1.32	1.28	1.22	1.16	1.13	
	6.70	4.66	3.83	3.36	3.06	2.86	2.69	2.55	2.46	2.37	2.29	2.23	2.12	2.04	1.92	1.84	1.74	1.64	1.57	1.47	1.42	1.32	1.24	1.19	
1000	3.85	3.00	2.61	2.38	2.22	2.10	2.02	1.95	1.89	1.84	1.80	1.76	1.70	1.65	1.58	1.53	1.47	1.41	1.36	1.30	1.26	1.19	1.13	1.08	
	6.66	4.62	3.80	3.34	3.04	2.82	2.66	2.53	2.43	2.34	2.26	2.20	2.09	2.01	1.89	1.81	1.71	1.61	1.54	1.44	1.38	1.30	1.19	1.11	
∞	3.84	2.99	2.60	2.37	2.21	2.09	2.01	1.94	1.88	1.83	1.79	1.75	1.69	1.64	1.57	1.52	1.46	1.40	1.35	1.28	1.24	1.17	1.11	1.00	
	6.64	4.60	3.78	3.32	3.02	2.80	2.64	2.51	2.41	2.32	2.24	2.18	2.07	1.99	1.87	1.79	1.69	1.59	1.52	1.41	1.36	1.25	1.15	1.00	

TABLE C.7. Rank-sum critical values

	(2, 4)		(4, 4)		(6, 7)	
3	11 .067	11 25	.029	28	56 .026	
	(2, 5)	12 24	.057	30	54 .051	
3	13 .047	(4, 5)			(6, 8)	
	(2, 6)	12 28	.032	29	61 .021	
3	15 .036	13 27	.056	32	58 .054	
4	14 .071	(4, 6)			(6, 9)	
	(2, 7)	12 32	.019	31	65 .025	
3	17 .028	14 30	.057	33	63 .044	
4	16 .056	(4, 7)			(6, 10)	
	(2, 8)	13 35	.021	33	69 .028	
3	19 .022	15 33	.055	35	67 .047	
4	18 .044	(4, 8)			(7, 7)	
	(2, 9)	14 38	.024	37	68 .027	
3	21 .018	16 36	.055	39	66 .049	
4	20 .036	(4, 9)			(7, 8)	
	(2, 10)	15 41	.025	39	73 .027	
4	22 .030	17 39	.053	41	71 .047	
5	21 .061	(4, 10)			(7, 9)	
	(3, 3)	16 44	.026	41	78 .027	
6	15 .050	18 42	.053	43	76 .045	
	(3, 4)	(5, 5)			(7, 10)	
6	18 .028	18 37	.028	43	83 .028	
7	17 .057	19 36	.048	46	80 .054	
	(3, 5)	(5, 6)			(8, 8)	
6	21 .018	19 41	.026	49	87 .025	
7	20 .036	20 40	.041	52	84 .052	
	(3, 6)	(5, 7)			(8, 9)	
7	23 .024	20 45	.024	51	93 .023	
8	22 .048	22 43	.053	54	90 .046	
	(3, 7)	(5, 8)			(8, 10)	
8	25 .033	21 49	.023	54	98 .027	
9	24 .058	23 47	.047	57	95 .051	
	(3, 8)	(5, 9)			(9, 9)	
8	28 .024	22 53	.021	63	108 .025	
9	27 .042	25 50	.056	66	105 .047	
	(3, 9)	(5, 10)			(9, 10)	
9	30 .032	24 56	.028	66	114 .027	
10	29 .050	26 54	.050	69	111 .047	
	(3, 10)	(6, 6)			(10, 10)	
9	33 .024	26 52	.021	79	131 .026	
11	31 .056	28 50	.047	83	127 .053	

Notes: The sample sizes are shown in parentheses (n_1, n_2) . The probability associated with a pair of critical values is the probability that $R \leq$ smaller value, or equally, it is the probability that $R \geq$ larger value. These probabilities are the closest ones to .025 and .05 that exist for integer values of R . The approximate .025 values should be used for a two-sided test with $\alpha = .05$, and the approximate .05 values for a one-sided test.

Source: Reprinted by kind permission from W. J. Dixon and F. J. Massey, *Introduction to Statistical Analysis*, 3rd ed., McGraw-Hill, New York, 1969.

TABLE C.8. Critical values of Kolmogorov-Smirnov D_n statistic

Sample size	Level of significance for $D_n = \sup F_n(x) - F_0(x) $			
	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
1	0.9000	0.9500	0.9750	0.9950
2	0.6838	0.7764	0.8419	0.9293
3	0.5648	0.6360	0.7076	0.8290
4	0.4927	0.5652	0.6239	0.7342
5	0.4470	0.5095	0.5633	0.6685
6	0.4104	0.4680	0.5193	0.6166
7	0.3815	0.4361	0.4834	0.5758
8	0.3583	0.4096	0.4543	0.5418
9	0.3391	0.3875	0.4300	0.5133
10	0.3226	0.3687	0.4093	0.4889
11	0.3083	0.3524	0.3912	0.4677
12	0.2958	0.3382	0.3754	0.4491
13	0.2847	0.3255	0.3614	0.4325
14	0.2748	0.3142	0.3489	0.4176
15	0.2659	0.3040	0.3376	0.4042
16	0.2578	0.2947	0.3273	0.3920
17	0.2504	0.2863	0.3180	0.3809
18	0.2436	0.2785	0.3094	0.3706
19	0.2374	0.2714	0.3014	0.3612
20	0.2316	0.2647	0.2941	0.3524
25	0.2079	0.2377	0.2640	0.3166
30	0.1903	0.2176	0.2417	0.2899
35	0.1766	0.2019	0.2243	0.2690
> 35	$1.07/\sqrt{n}$	$1.22/\sqrt{n}$	$1.36/\sqrt{n}$	$1.63/\sqrt{n}$

Source: Adapted from Table 1 of L. H. Miller, "Table of percentage points of Kolmogorov statistics," *J. Am. Stat. Assoc.* **51**, 113, (1956), and Table 1 of F. J. Massey, Jr., "The Kohnogorov-Smirnov test for goodness-of-fit," *J. Am. Stat. Assoc.* **46**, 70, (1951), with permission of the authors and publisher.

TABLE C.9. Critical values of Kolmogorov–Smirnov \hat{D}_n statistic for the exponential distribution with an unknown mean θ

Sample size	<i>Level of significance for $\hat{D}_n = \sup \hat{F}_n(x) - F_0(x, \theta)$</i>				
	$\alpha = 0.20$	$\alpha = 0.15$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
3	0.451	0.479	0.511	0.551	0.600
4	0.396	0.422	0.449	0.487	0.548
5	0.359	0.382	0.406	0.442	0.504
6	0.331	0.351	0.375	0.408	0.470
7	0.309	0.327	0.350	0.382	0.442
8	0.291	0.308	0.329	0.360	0.419
9	0.277	0.291	0.311	0.341	0.399
10	0.263	0.277	0.295	0.325	0.380
11	0.251	0.264	0.283	0.311	0.365
12	0.241	0.254	0.271	0.298	0.351
13	0.232	0.245	0.261	0.287	0.338
14	0.224	0.237	0.252	0.277	0.326
15	0.217	0.229	0.244	0.269	0.315
16	0.211	0.222	0.236	0.261	0.306
17	0.204	0.215	0.229	0.253	0.297
18	0.199	0.210	0.223	0.246	0.289
19	0.193	0.204	0.218	0.239	0.283
20	0.188	0.199	0.212	0.234	0.278
25	0.170	0.180	0.191	0.210	0.247
30	0.155	0.164	0.174	0.192	0.226
> 30	$0.86/\sqrt{n}$	$0.91/\sqrt{n}$	$0.96/\sqrt{n}$	$1.06/\sqrt{n}$	$1.25/\sqrt{n}$

Source: Adapted from Table 1 of H. W. Lilliefors, “On the Kolmogorov–Smirnov test for the exponential with mean unknown,” *J. Am. Stat. Assor.* **64**, 388, (1969), with permission of the author and publisher.

TABLE C.10. Critical values of Kolmogorov-Smirnov \hat{D}_n statistic for the normal distribution with an unknown mean μ and an unknown variance σ^2

Sample size	Level of significance for $\hat{D}_n = \sup \hat{F}_n(x) - F_0(x, \mu, \sigma^2) $				
	$\alpha = 0.20$	$\alpha = 0.15$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
4	0.300	0.319	0.352	0.381	0.417
5	0.285	0.299	0.315	0.337	0.405
6	0.265	0.277	0.294	0.319	0.364
7	0.247	0.258	0.276	0.300	0.348
8	0.233	0.244	0.261	0.285	0.331
9	0.223	0.233	0.249	0.271	0.311
10	0.215	0.224	0.239	0.258	0.294
11	0.206	0.217	0.230	0.249	0.284
12	0.199	0.212	0.223	0.242	0.275
13	0.190	0.202	0.214	0.234	0.268
14	0.183	0.194	0.207	0.227	0.261
15	0.177	0.187	0.201	0.220	0.257
16	0.173	0.182	0.195	0.213	0.250
17	0.169	0.177	0.189	0.206	0.245
18	0.166	0.173	0.184	0.200	0.239
19	0.163	0.169	0.179	0.195	0.235
20	0.160	0.166	0.174	0.190	0.231
25	0.142	0.147	0.158	0.173	0.200
30	0.131	0.136	0.144	0.161	0.187
> 30	$0.736/\sqrt{n}$	$0.768/\sqrt{n}$	$0.805/\sqrt{n}$	$0.886/\sqrt{n}$	$1.031/\sqrt{n}$

Source: Adapted from Table 1 of H. W. Lilliefors, "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *J. Am. Stat. Assoc.* **62**, 400, (1967), with permission of the author and publisher.

TABLE C.11. Probabilities for the Mann-Kendall nonparametric test for trend

S	Values of n				S	Values of n		
	4	5	8	9		6	7	10
0	0.625	0.592	0.548	0.540	1	0.500	0.500	0.500
2	0.375	0.408	0.452	0.460	3	0.360	0.386	0.431
4	0.167	0.242	0.360	0.381	5	0.235	0.281	0.364
6	0.042	0.117	0.274	0.306	7	0.136	0.191	0.300
8		0.042	0.199	0.238	9	0.068	0.119	0.242
10		0.0 ² 83	0.138	0.179	11	0.028	0.068	0.190
12			0.089	0.130	13	0.0 ² 83	0.035	0.146
14			0.054	0.090	15	0.0 ² 14	0.015	0.108
16			0.031	0.060	17		0.0 ² 54	0.078
18			0.016	0.038	19		0.0 ² 14	0.054
20			0.0 ² 71	0.022	21		0.0 ³ 20	0.036
22			0.0 ² 28	0.012	23			0.023
24			0.0 ³ 87	0.0 ² 63	25			0.014
26			0.0 ³ 19	0.0 ² 29	27			0.0 ² 83
28			0.0 ⁴ 25	0.0 ² 12	29			0.0 ² 46
30				0.0 ³ 43	31			0.0 ² 23
32				0.0 ³ 12	33			0.0 ² 11
34				0.0 ⁴ 25	35			0.0 ³ 47
36				0.0 ⁵ 28	37			0.0 ³ 18
					39			0.0 ⁴ 58
					41			0.0 ⁴ 15
					43			0.0 ⁵ 28
					45			0.0 ⁶ 28

Note: Repeated zeros are indicated by powers; for example, 0.0³47 stands for 0.00047.

Source: Adapted from Table A18, Appendix A of R. O. Gilbert, *Statistical Methods for Environmental Pollution Monitoring*, ©1987, Van Nostrand Reinhold Company, New York, with permission of the publisher. For an extension of this table for $n > 10$, refer to Table A.21 of M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, Wiley, New York, 1973.

Appendix D

Laplace Transforms

We have had several occasions to use Laplace transforms in this text. This appendix examines useful properties of the Laplace transform and gives a table of Laplace transforms that are often used. For further details about Laplace transform, see Schiff [SCHI 1999] or Bellman and Roth [BELL 1994].

Suppose that $f(t)$ is a piecewise-continuous function, defined at least for $t > 0$, and is of *exponential* order α , meaning that it does not grow any faster than the exponential $e^{\alpha t}$:

$$|f(t)| \leq M e^{\alpha t}, \quad t > 0$$

for some constant M . Then the Laplace transform of $f(t)$, denoted by $\bar{f}(s)$, is defined by the integral

$$\bar{f}(s) = \int_0^\infty e^{-st} f(t) dt \tag{D.1}$$

for any complex number s such that its real part $\operatorname{Re}(s) > \alpha$.

Important $[f(t), \bar{f}(s)]$ pairs are given in Table D.1.

The Laplace transform is generally used to simplify calculations. However, the technique will not be of value unless it is possible to recover the function $f(t)$ from its transform $\bar{f}(s)$. In fact, we have the following theorem, which unfortunately does not have an elementary proof and so it is stated without proof.

THEOREM D.1 (Correspondence or Uniqueness Theorem). If $\bar{f}(s) = \bar{g}(s)$ for all s , then $f(t) = g(t)$ for all t .

In other words, two functions that have the same transform are identical.

TABLE D.1. Important transform pairs

$f(t), t > 0$	$\bar{f}(s)$
1. c a constant	$\frac{c}{s}$
2. Unit step, $u(t - x)$	$\frac{e^{-sx}}{s} \quad x > 0$
3. Delta function, $\delta(t - x)$	$e^{-sx} \quad x > 0$
4. t	$\frac{1}{s^2}$
5. t^a	$\frac{\Gamma(a + 1)}{s^{a+1}}, \quad (a > -1)$
6. t^a	$\frac{a!}{s^{a+1}}, \quad (a = 0, 1, 2, \dots)$
7. $t^{-1/2}$	$\sqrt{\frac{\pi}{s}},$
8. e^{-at}	$\frac{1}{s + a}, \quad [\text{Re}(s) > a]$
9. te^{-at}	$\frac{1}{(s + a)^2}, \quad [\text{Re}(s) > a]$
10. $t^b e^{-at}$	$\frac{b!}{(s + a)^{b+1}}, \quad [b = 0, 1, 2, \dots, \text{Re}(s) > a]$
11. $t^\beta e^{-at}$	$\frac{\Gamma(\beta + 1)}{(s + a)^{\beta+1}}, \quad [\beta > 0, \text{Re}(s) > a]$

As a special case of Definition D.1, assume that $f(t)$ is the pdf of some nonnegative, absolutely continuous random variable X [in this case $f(t)$ is the short form for $f_X(t)$]; then $\bar{f}(s)$ is also called the Laplace-Stieltjes transform [also denoted by acronym LST or symbolically as $L_X(s)$] of the random variable X . In this case $\bar{f}(s)$ always exists for any positive α , since

$$|\bar{f}(s)| \leq \int_0^\infty |e^{-st} f(t)| dt \leq \int_0^\infty f(t) dt = 1 \quad \text{for } \text{Re}(s) > 0.$$

In this connection, the usefulness of the Laplace-Stieltjes transform stems from the convolution property and the moment generating property (besides the uniqueness property), as follows.

THEOREM D.2 (The Convolution Theorem). If X_1, X_2, \dots, X_n are independent random variables with respective Laplace–Stieltjes transforms $L_{X_1}, \dots, L_{X_n}(s)$, then the LST of the random variable

$$Z = \sum_{i=1}^n X_i$$

is given by

$$L_Z(s) = \prod_{i=1}^n L_{X_i}(s).$$

THEOREM D.3 (Moment Generating Property). Let X be a random variable possessing a Laplace–Stieltjes transform $L_X(s)$. Then the k th ($k = 1, 2, \dots$) moment of X is given by

$$E[X^k] = (-1)^k \left. \frac{d^k L_X(s)}{ds^k} \right|_{s=0}. \quad (\text{D.2})$$

Thus, if X denotes the time to failure of a system, then, from a knowledge of its LST $L_X(s)$, we can quickly obtain the system MTTF (mean time to failure), $E[X]$, while it may be considerably more difficult to obtain the density $f_X(t)$ and the reliability $R_X(t)$.

The Laplace transform is also used in solving differential equations, since it reduces an ordinary linear differential equation with constant coefficients into an algebraic equation in s . The solution in terms of s is then converted into a time function by an inversion that is unique by Theorem D.1.

The usefulness of the Laplace transform in solving differential equations is based on the fact that it is a *linear* operator and that the Laplace transform of any derivative of a function is easily computed from the transform of the function itself.

THEOREM D.4 (Linearity Property). Define the function

$$g(t) = \sum_{i=1}^n C_i f_i(t) \text{ for some constants } C_1, C_2, \dots, C_n. \text{ Then}$$

$$\bar{g}(s) = \sum_{i=1}^n C_i \bar{f}_i(s). \quad (\text{D.3})$$

THEOREM D.5 (Initial Value Theorem). Let f be a function such that f and its derivative f' are both of exponential order α . Then the Laplace transform of f' is given by

$$\bar{f}'(s) = s\bar{f}(s) - f(0). \quad (\text{D.4})$$

Proof: By Definition (D.1), we obtain

$$\bar{f}'(s) = \int_0^\infty \frac{df}{dt} e^{-st} dt.$$

Integrating by parts, we have

$$\bar{f}'(s) = e^{-st} f(t)|_0^\infty + s \int_0^\infty f(t) e^{-st} dt.$$

The second term on the right-hand side is simply $s\bar{f}(s)$. Consider the first term on the right-hand side. Since, by assumption, $f(t)$ grows more slowly than the exponential e^{+st} for sufficiently large s :

$$\lim_{t \rightarrow \infty} e^{-st} f(t) = 0$$

At the lower limit we obtain

$$\lim_{t \rightarrow 0} e^{-st} f(t) = f(0)$$

Hence the result follows. If a singularity occurs at $t = 0$, we must be careful to replace $f(0)$ by $f(0^-)$, which is the limit of $f(t)$ as t approaches 0 from the left-and side. Equation (D.4) holds if $f(x)$ is right continuous at $x = 0$.

Important properties of the Laplace transform are summarized in Table D.2.

The procedure of solving a differential equation by means of Laplace transform is illustrated in Figure D.1.

We now discuss the inversion of a Laplace transform. Consider the solution to the differential equation of Figure D.1 in the s domain:

$$\bar{y}(s) = \frac{1}{(a+s)(b+s)}.$$

Since this function does not occur on the right-hand side of Table D.1, we cannot use table lookup and we must use alternate methods for its inversion. A procedure quite often used is **partial fraction expansion** (or **decomposition**). In this case we rewrite

$$\begin{aligned} \bar{y}(s) &= \frac{C_1}{a+s} + \frac{C_2}{b+s} \\ &= \frac{(C_1 b + C_2 a) + (C_1 + C_2)s}{(a+s)(b+s)} \\ &= \frac{1}{(a+s)(b+s)}. \end{aligned}$$

TABLE D.2. Properties of Laplace transforms

<i>Function</i>	<i>Laplace transform</i>	
1. $f(t), t > 0$	$\bar{f}(s) = \int_0^\infty e^{-st} f(t) dt$	(definition)
2. $af(t) + bg(t)$	$a\bar{f}(s) + b\bar{g}(s)$	[linearity (superposition) property]
3. $f(at), a > 0$	$\frac{1}{a}\bar{f}\left(\frac{s}{a}\right)$	
4. $f(t - a), a \geq 0$	$e^{-as}\bar{f}(s)$	
5. $e^{-at}f(t), a \geq 0$	$\bar{f}(s + a)$	
6. $f'(t)$	$s\bar{f}(s) - f(0)$	
7. $f^{(n)}(t)$	$s^n\bar{f}(s) - s^{n-1}f(0) - s^{n-2}f'(0)$ $- \dots - f^{(n-1)}(0)$	
8. $\int_{-\infty}^t f(t')dt'$	$\frac{1}{s}\bar{f}(s)$	
9. $\int_{-\infty}^t f(\tau)g(t - \tau)d\tau$	$\bar{f}(s)\bar{g}(s)$	(convolution theorem)
10. $tf(t)$	$-\frac{d\bar{f}(s)}{ds}$	
11. $t^n f(t), n$ a positive integer	$(-1)^n \bar{f}^{(n)}(s)$	
12. $\frac{f(t)}{t}$	$\int_s^\infty \bar{f}(x)dx$	
13. $\int_0^\infty f(t)dt = \bar{f}(0)$		(integral property)
14. $\lim_{t \rightarrow 0} f(t) =$	$\lim_{s \rightarrow \infty} s\bar{f}(s)$	(initial value theorem ^a)
15. $\lim_{t \rightarrow \infty} f(t) =$	$\lim_{s \rightarrow 0} s\bar{f}(s)$	(final value theorem ^a)

^a Initial-value and final-value theorems apply only when all the poles of $s\bar{f}(s)$ lie on the left half of the s plane; that is, if we write $s\bar{f}(s)$ as the ratio of two polynomials $N(s)/D(s)$, then all the roots of the equation $D(s) = 0$ are the poles of $s\bar{f}(s)$, and these must satisfy the condition $\operatorname{Re}(s) < 0$.

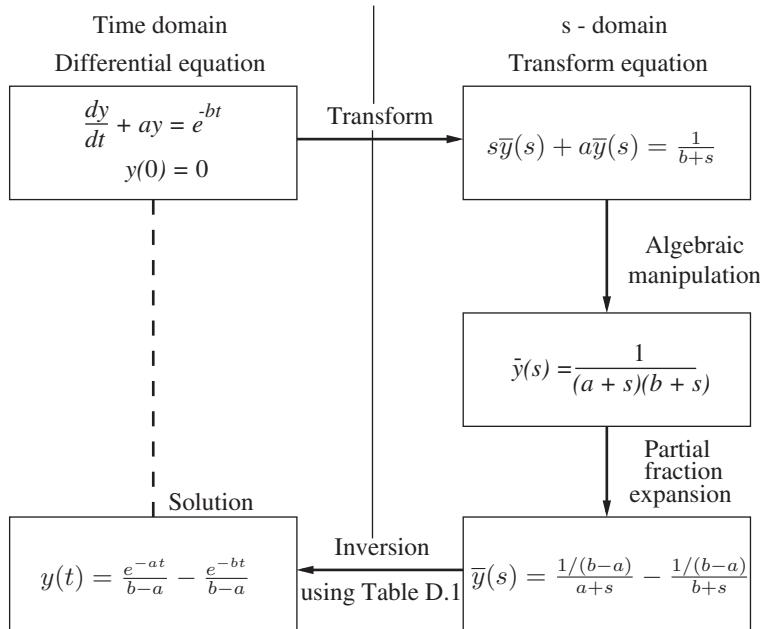


Figure D.1. Solution of a linear differential equation using Laplace transform procedure

Hence

$$C_1 b + C_2 a = 1 \quad \text{and} \quad C_1 + C_2 = 0$$

or

$$C_1 = \frac{1}{b-a} \quad \text{and} \quad C_2 = -\frac{1}{b-a}.$$

Thus

$$\bar{y}(s) = \frac{1/(b-a)}{a+s} - \frac{1/(b-a)}{b+s}.$$

Now, using the linearity property of Laplace transform and Table D.1 (entry 6), we conclude that

$$y(t) = \frac{1}{b-a}e^{-at} - \frac{1}{b-a}e^{-bt}, t > 0.$$

More generally, suppose $\bar{f}(s)$ is a rational function of s :

$$\begin{aligned}\bar{f}(s) &= \frac{N(s)}{D(s)} \\ &= \frac{N(s)}{\prod_{i=1}^d (s + a_i)},\end{aligned}\tag{D.5}$$

where we assume that the degree of polynomial $D(s)$ is at least one greater than the degree of $N(s)$ and that all the roots of $D(s) = 0$ are distinct. Then the partial fraction expansion of $\bar{f}(s)$ is

$$\bar{f}(s) = \sum_{i=1}^d \frac{C_i}{s + a_i}, \quad (\text{D.6})$$

where

$$C_i = \left. \left(\frac{N(s)}{D(s)} (s + a_i) \right) \right|_{s=-a_i}.$$

The inversion of (D.6) is easily obtained

$$f(t) = \sum_{i=1}^d C_i e^{-a_i t}.$$

REFERENCES

- [SCHI 1999] J. L. Schiff, *The Laplace Transform: Theory and Applications*, Springer-Verlag, New York, 1999.
- [BELL 1994] R. Bellman and R. Roth, *Laplace Transform*, World Scientific, Singapore, 1994.

Appendix E

Program Performance Analysis

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

TABLE E.1. table

Statement type	Average execution time, $E[T]$	Variance of execution time, $\text{Var}[T]$	Laplace transform, $L_T(s)$
If B then S_1 else S_2	$pE[X_1] + (1 - p)E[X_2]$	$p\text{Var}[X_1] + (1 - p)\text{Var}[X_2] + p(1 - p)(E[X_1] - E[X_2])^2$	$pL_{X_1}(s) + (1 - p)L_{X_2}(s)$
case i of 1: S_1 ; 2: S_2 ; end	$\sum_i p_i E[X_i]$	$\sum_i p_i (E[X_i^2] - (E[T])^2)$	$\sum_i p_i L_{X_i}(s)$
repeat S until B	$E[X]/p$	$\frac{\text{Var}[X]}{p} + \frac{1 - p}{p^2} (E[X])^2$	$\sum_{n=1}^{\infty} p_N(n) [L_X(s)]^n$
while B do S	$\frac{pE[X]}{1 - p}$	$\frac{p\text{Var}[X]}{1 - p} + \frac{p(E[X])^2}{(1 - p)^2}$	$\sum_{n=0}^{\infty} p_N(n) [L_X(s)]^n$
for $i := 1$ to n do S	$nE[X]$	$n\text{Var}[X]$	$[L_X(s)]^n$
begin $S_1; S_2; \dots; S_n$ end	$\sum_{i=1}^n E[X_i]$	$\sum_{i=1}^n \text{Var}[X_i]$	$\prod_{i=1}^n L_{X_i}(s)$

Notes: The execution time of statement group S_i is denoted by X_i . It is assumed that B is a Bernoulli random variable with parameter p . Independence assumptions are made wherever required. In the expressions for **while** and **repeat** statements, the roles of “success” and “failure” are reversed since a “success” of the loop test terminates execution of the **repeat** statement while a “failure” of the loop test terminates the execution of the **while** statement. Furthermore, the number of iterations of the **repeat** loop is geometrically distributed while the number of iterations of the **while** loop has a modified geometric distribution.

Author Index

- Abdulla, G., 203, 255, 315, 348
Abrams, M., 203, 255, 315, 348
Abramson, N., 388, 418
Agrawal, D. P., 802
Ajmone-Marsan, M., 418, 558, 569, 571, 796
Akimaru, H., 444, 571, 800
Akyildiz, I., 616, 620, 655
Alisouskas, V. F., 798
Allan, R. N., 794
Allen, A. O., 796
Almeida, V. A. F., 203, 254, 611, 658, 798, 800
Anderson, T., 801
Apostolakis, G., 348
Arlat, J., 532, 574, 699, 752
Arnold, T. F., 521, 571
Arsenault, J. E., 801
Ash, R. B., 29, 62, 200, 362, 365, 370, 418
Asmussen, S., 791
Avizienis, A., 190, 201
Avresky, D., 801

Bagchi, K., 799
Balakrishnan, M., 33, 62
Balbo, G., 418, 558, 569, 571, 613, 655, 656, 793, 796
Banks, J., 799
Bard, Y., 618, 777, 655, 790
Barlow, R. E., 133, 200, 263, 299, 320, 324, 326, 348, 794
Basawa, I. V., 793
Baskett, F., 625, 653, 655
Beaudry, M. D., 180, 200
Begain, K., 657

Bellman, R., 828, 834
Beounes, C., 532, 574
Berry, D. A., 795
Bertsekas, D., 293, 299, 388, 418, 800
Bestavros, A., 150, 201, 203, 254
Beutler, F. J., 433, 571, 587, 656
Bhandiwad, R. A., 598, 607, 609, 626, 659
Bhat, U. N., 200, 332, 348, 357, 409, 418, 496, 571, 702, 751, 792
Billinton, R., 794
Birolini, A., 794
Blake, I. F., 255
Blake, J. T., 170, 171, 200, 546, 571
Blakemore, A., 527, 545, 550, 552, 553, 555, 558, 562, 572
Boas, R. P., 792
Bobbio, A., 570, 571
Bolch, G., 366, 397, 418, 545, 553, 571, 578, 616, 618, 620, 626, 654, 655–657, 793
Bouricius, W. G., 280, 284, 299
Boxma, O. J., 578, 656, 793
Breiman, L., 125, 200
Breslau, L., 203, 255
Browne, J. C., 576
Bruell, S. C., 613, 655, 656, 793
Burke, P. J., 580, 587, 656
Burks, A. W., 119, 119
Buzen, J. P., 461, 572, 597, 598, 609, 656, 657

Cao, P., 203, 255
Cao, Y., 23, 63, 92, 120, 512, 513, 516, 571, 576

Probability and Statistics with Reliability, Queuing, and Computer Science Applications,
Second Edition. Kishor S. Trivedi.

© 2016 John Wiley & Sons, Inc. Published 2016 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/trivedi/probabilityandstatistics2e

- Carson II, J. S., 799
 Carter, W. C., 280, 284, 299
 Chandy, K. M., 576, 591, 620, 623,
 625, 631, 653, 654, 655, 656,
 658, 797, 798
 Chimento, P. F., 450, 527, 545, 550,
 552, 553, 555, 558, 562, 571, 572
 Chiola, G., 552, 571
 Choi, B. D., 572
 Choi, H., 397, 418, 535–537, 555,
 569, 571, 572, 574
 Choi, K. B., 572
 Chung, K. L., 791
 Ciardo, G., 527, 545, 550, 552, 553,
 555, 558, 562, 566, 567, 572,
 649, 656
 Cinlar, E., 493, 572, 712, 751, 792
 Clarke, A. B., 270, 299, 418, 791
 Clary, J., 165, 200
 Coffman, E. G., 327, 347, 348, 653,
 656, 779, 791
 Cohen, J. W., 793
 Constantinescu, C., 699, 751
 Conte, G., 418, 558, 569, 571, 796
 Conway, A. E., 626, 656
 Cooper, R. B., 793
 Courtois, P. J., 654, 656, 797
 Cox, D. R., 287, 291, 299, 792, 793
 Crouzet, Y., 699, 752
 Crovella, M. E., 150, 201, 203, 254

 Daduna, H., 578, 641, 656, 658
 Daigle, J. N., 793
 De Meer, H., 366, 397, 418, 545, 553,
 571, 578, 618, 620, 626, 654,
 656, 793
 de Oliveira, A., 203, 254
 Degroot, M. H., 795
 Demetrescu, C., 77, 119
 Deng, S., 150, 201
 Denning, P. J., 327, 347, 348, 461,
 572, 597, 657, 779, 791
 Deswarthe, Y., 531, 575
 Devarakonda, M. V., 709, 710, 751
 Dhillon, B. S., 794, 795

 Disney, R. L., 270, 299, 418, 791
 Donatelli, S., 558, 569, 571, 796
 Donatiello, L., 797
 Doss, K., 51, 62
 Dowdy, L., 657, 797, 798
 Draper, N. R., 776, 791, 795
 Dugan, J. B., 281, 299, 541, 572

 Elmaghriby, S. E., 801

 Fan, L., 203, 255
 Fayolle, G., 653, 656
 Feller, W., 355, 364, 418, 791
 Ferrari, D., 709, 710, 752, 797
 Fischer, W., 572
 Fishman, G. S., 157, 201, 752, 799
 Floyd, S., 150, 201
 Foley, R. D., 584, 658
 Fox, B. L., 89, 119, 549, 572
 Fox, E., 203, 255, 315, 348
 Franceschinis, G., 558, 569, 571, 796
 Freund, J. E., 796
 Fricks, R., 493, 570, 572
 Fujiwara, E., 173, 201
 Fuller, S. H., 499, 572
 Furchtgott, D. G., 180, 201

 Gallager, R., 293, 299, 388, 418, 800
 Garg, S., 486, 493, 572, 573, 768, 791
 Gaver, D. P., 271, 299
 Gay, F. A., 702, 752
 Geist, R. M., 438, 504, 573
 Georganas, N. D., 626, 656
 German, R., 397, 418, 419, 570, 573,
 797
 Ghanta, S., 655
 Gilbert, R. O., 768–770, 791
 Glynn, P. W., 89, 119, 549, 572
 Gnedenko, B. V., 791, 793
 Goševa-Popstojanova, K., 354, 372,
 419
 Goel, A. L., 468, 469, 573, 680, 752
 Gokhale, S., 407, 408, 419, 469, 551,
 573, 575
 Goldstine, H. H., 119, 119

- Golub, G. H., 802
Goodman, S. E., 17, 62
Gordon, W., 594, 596, 598, 657
Goyal, A., 543, 573
Graham, G. S., 658, 798
Grassman, W., 545, 551, 573, 802
Graybill, F. A., 796
Greiner, S., 366, 397, 418, 545, 553,
 571, 578, 618, 620, 626, 654,
 656, 657, 793
Gross, D., 397, 419, 444, 573, 654,
 793
Gross, D., 657
Guenther, N. J., 797

Hamming, R. W., 49, 60, 62, 125,
 200, 201
Han, D. H., 572
Han, J. J., 23, 63, 512, 513, 516, 576
Han, Y.-W., 59, 62
Haring, G., 464, 513, 573, 797
Harris, C. M., 397, 419, 444, 573,
 654, 657, 793
Harrison, P. G., 797
Hartigan, J. A., 709, 752
Hartley, H. O., 90, 120, 796
Hatfield, M. A., 761, 791
Haverkort, B. R., 516, 573, 576, 578,
 637, 657, 797
Hecht, H., 45, 62
Hedetniemi, S., 17, 62
Heidelberger, P., 527, 573
Heimann, D., 493, 573
Henley, E., 33, 62
Herzog, U., 631, 656
Hess, G., 275, 299
Hillier, F. H., 801
Hillston, J., 797
Hoel, P. G., 737, 752
Hofri, M., 298, 300, 799
Hogarth, J., 654, 657
Hogg, R. V., 795
Hollander, M., 795
Hong, D., 463, 573
Hoover, S. V., 800

Horgan, J. R., 407, 408, 419
Hosseini, M. M., 476, 573
Howard, J. H., 620, 623, 625, 654,
 656
Howe, R., 58, 62, 493, 495, 508,
 574-576
Hsueh, M.-C., 140, 201, 238, 255,
 570, 574
Huang, Y., 486, 573
Hui, J. Y., 800
Hunter, D., 210, 255
Hunter, S. W., 485, 574

Iannino, A., 802
Ibe, O., 58, 62, 493, 495, 569, 574
Ingle, A. D., 776, 791
Iyer, R. K., 140, 201, 238, 255, 570,
 574, 709, 710, 751

Jackson, J. R., 582, 657
Jai, A., 165, 200
Jain, R., 657, 797
Jalote, P., 801
Jang, J. S., 572
Jelinski, Z., 473, 574
Jirachiefpattana, A., 555, 574
Johnson, R. A., 796

Kaâniche, M., 531
Kang, S. H., 574
Kanoun, K., 532, 574
Kant, K., 797
Kao, E. P. C., 792
Karlin, S., 792, 793
Kawashima, K., 444, 571, 800
Keinze, M. G., 657
Kelly, F. P., 626, 654, 657, 793
Kelton, W. D., 799
Kemeny, G., 792
Kendall, M. G., 665, 752, 773, 791,
 796
Kerr, R. M., 476, 573
Kesidis, G., 800
Khintchine, A. Y., 320, 348, 793
Kijima, M., 792

- King, P. J. B., 800
 Kinicki, R. E., 761, 791
 Kintala, C. M., 486, 573
 Kleijnen, J. P. C., 715, 752, 799
 Kleinrock, L., 396, 419, 432, 442,
 461, 574, 607, 654, 657, 793
 Knuth, D. E., 86, 97, 102, 119, 155,
 201, 222, 255, 276, 277, 300,
 419, 799
 Kobayashi, H., 223, 255, 435, 574
 Koslov, J., 173, 201
 Kouvaratsos, D. D., 800
 Kovalenko, I., 62
 Kovalenko, I. N., 793, 795
 Krishna, C. M., 538, 576, 801
 Kulkarni, V. G., 309, 348, 397, 418,
 419, 555, 571, 641, 649, 652,
 653, 657, 658, 792
 Kumamoto, H., 33, 62
 Kummerle, K., 800
 Kuther, M. H., 796
 Kuznetsov, N. Yu., 795
 Lai, R., 555, 574, 626
 Lam, S., 657
 Laprie, J.-C., 532, 574, 801
 Larson, H. J., 80, 120, 796
 Lavenberg, S. S., 543, 573, 578, 613,
 618, 649, 657, 658, 798
 Law, A. M., 799
 Lazowska, E. D., 658, 798
 Lee, I., 140, 201, 238, 255
 Lee, K. W., 339, 348
 Lee, P. A., 45, 63, 801
 Lee, Y.-H., 801
 Leech, R. L., 609, 659
 Leemis, L. M., 80, 89, 120, 255, 678,
 694, 699, 752, 795
 Leland, W., 150, 201
 Levy, A., 349
 Lieberman, G. J., 801
 Lilliefors, H. W., 747, 752
 Limb, J., 800
 Lindemann, C., 397, 419, 552, 570,
 574, 797, 798
 Lindgren, B. W., 795
 Lipsky, L., 794
 Littlewood, B., 262, 300
 Liu, C. L., 15, 59, 62
 Liu, M., 787, 791
 Logothetis, D., 397, 419, 570, 574,
 659
 Lowery, C., 797
 Lucantoni, D., 501, 574
 Luo, T., 17, 62
 Lyu, M. R., 801, 802
 Macnair, E. A., 798
 Maieli, M. V., 776, 791
 Mainkar, V., 641, 643, 649, 652, 653,
 658
 Malhotra, M., 265, 300, 493, 550,
 551, 570, 574, 575
 Marie, R., 464, 513, 573, 637, 657,
 797
 Martin, A., 620, 656
 Martins, E., 699, 752
 Mathur, F. P., 190, 201
 McCabe, J., 419
 Medhi, J., 414, 419, 792
 Meier-Hellstern, K., 501, 572, 574
 Melamed, B., 587, 649, 656, 658,
 799
 Menasce, D. A., 611, 658, 798, 800
 Mendelson, H., 265, 279, 300
 Meyer, C., 802
 Meyer, J. F., 180, 201
 Miller, H. D., 792
 Miller, I., 796
 Misra, K. B., 33, 34, 62, 795
 Mitrani, I., 613, 649, 653, 656, 658
 Mittal, N., 493, 573
 Miyahara, H., 150, 201
 Moder, J., 801
 Molloy, M. K., 363, 419, 798
 Mood, A. M., 796
 Moranda, P. B., 473, 574
 Morse, P. M., 445, 575
 Moshe, M., 800
 Muntz, R. R., 625, 653, 655

- Muppala, J., 265, 300, 493, 516, 527, 545, 550–553, 555, 558, 562, 566, 567, 572, 573, 575, 576, 641, 649, 652, 653, 656, 658
Murata, M., 150, 201
Murata, T., 553, 575
Murata, Y., 92, 120
Musa, J. D., 469, 575, 802

Nabe, M., 150, 201
Nachtschiem, C. J., 796
Nagase, S., 92, 120
Nannicini, S., 293, 300
Narihisa, H., 199, 201
Nelson, B. L., 799
Nelson, R., 791, 797
Neter, J., 787, 791, 796
Neuts, M. F., 501, 574, 575, 803
Newell, G., 594, 596, 598, 657
Ng, C. H., 794
Ng, Y.-W., 290
Nicol, D. M., 799
Nishio, T., 801
Noether, G. E., 737, 739, 752, 775, 790, 791, 796
Nutt, G., 455, 575

Oh, C., 574
Ohba, M., 679, 752
Ohtera, H., 199, 201
Okumoto, K., 468, 469, 573, 575, 680, 752, 802
Onvural, R. O., 373, 419, 800
Ortalo, R., 531, 575
Osaki, S., 801
Ott, T., 150, 201
Ozekici, S., 795

Palacios, F. G., 625, 653, 655
Palm, C., 320, 349
Park, K., 800
Parzen, E., 363, 366, 409, 419, 469, 575, 791, 792
Patel, N. M., 797
Paul, F., 800

Paxson, V., 150, 201
Pearson, E. S., 90, 120, 796
Pecorella, T., 293, 300
Peebles, P. Z., 214, 255
Pegg, P. A., 795
Perros, H., 658, 794
Perry, R. F., 800
Phillips, G., 203, 255
Plemons, R. J., 802
Pliskin, J. S., 265, 279, 300
Port, S. C., 737, 752
Powell, D., 699, 752
Prabhu, N. U., 792, 794
Pradhan, D. K., 802
Proschan, F., 133, 200, 263, 299, 320, 324, 326, 348, 794
Puigjaner, R., 464, 513, 573
Puliafito, A., 33, 51, 63, 152, 164, 178, 201, 299, 300, 455, 484, 507, 527, 531, 546, 552, 555, 570, 571, 572, 574, 576, 642, 654, 658, 802
Putnam, L., 791

Rai, S., 52, 62, 802
Ramamoorthy, C. V., 59, 62
Ramani, S., 551, 575
Ramesh, A. V., 546, 575
Randall, R. B., 476, 573
Randell, B., 45, 63, 278, 300, 801
Rao, G. S., 382, 419
Rao, T. R., 173, 201
Rappaport, S., 463, 573
Rappaport, T. S., 798
Reibman, A., 546, 547, 551, 571, 575
Reiser, M., 613, 618, 649, 657, 658, 797
Rindos, A., 524, 525, 551, 575
Robertazzi, T. G., 800
Roberts, J. A., 801
Rom, R., 800
Romig, H. G., 81, 120
Rosetti, D. J., 570, 574

- Ross, K. W., 800
 Ross, S. M., 329, 349, 451, 506, 576,
 708, 752, 791, 792
 Roth, R., 828, 834
 Rubino, G., 573, 637, 657, 797
 Rubinstein, R. Y., 799
 Rudin, W., 125, 201
 Saeks, R., 165, 200
 Sahner, R. A., 33, 51, 63, 152, 164,
 178, 201, 299, 300, 455, 484,
 507, 527, 531, 546, 552, 555,
 576, 642, 654, 658, 802
 Sathaye, A., 493, 508, 574–576
 Sauer, C. H., 591, 620, 631, 654, 656,
 658, 798
 Schader, M., 80, 120
 Schassberger, R., 641, 658
 Schatzoff, M., 777, 790
 Schiff, J. L., 828, 834
 Schmid, F., 80, 120
 Schneweiss, W., 795
 Schneider, P. R., 280, 284, 299
 Schormans, J. A., 576
 Schriber, T. J., 800
 Schwartz, M., 801
 Sedgewick, R., 226, 255
 Sen, P. K., 770, 791
 Serazzi, G., 709, 710, 752, 797
 Sevcik, K. C., 349, 613, 618, 649,
 654, 655, 657–659, 798
 Shedler, G. S., 387, 419
 Shenker, S., 203, 255
 Shin, B.-C., 572
 Shin, K. G., 538, 576
 Shiriaev, A. N., 792
 Shooman, M. L., 795, 802
 Siewiorek, D. P., 165, 200, 776, 791,
 802
 Silverster, J. A., 576
 Simon, B., 584, 658
 Singh, C., 794
 Singpurwalla, N. D., 802
 Sivazlian, B. D., 801
 Sliski, D. J., 761, 791
 Smith, C. U., 798
 Smith, H., 776, 791, 795
 Smith, R. M., 551, 575
 Smith, W. E., 485, 574
 Smith, W. L., 793
 Snell, J. L., 792
 Spiegel, M. R., 796
 Spirn, J. R., 380, 384, 419, 798
 Standridge, C. R., 203, 255, 315,
 348
 Stanfel, L. E., 801
 Stark, H., 46, 63, 302, 310, 349, 798
 Sterbenz, P. H., 253, 255
 Stewart, W. J., 366, 419, 542, 545,
 551, 576, 803
 Stidham, S., 433, 576
 Stiffler, J. J., 540, 576
 Stone, C. J., 737, 752
 Stuart, A., 665, 752, 773, 791, 796
 Sun, H.-R., 23, 36, 58, 63, 92, 120,
 512, 513, 516, 571, 576
 Sung, D. K., 574
 Swarz, R. S., 802
 Syski, R., 793
 Takacs, L., 794
 Takagi, H., 654, 658, 794, 801
 Tang, D., 140, 201, 238, 255
 Tanis, E. A., 795
 Taylor, H. M., 792, 793
 Telek, M., 570, 571, 572
 Thomasian, A., 798
 Thompson, G. L., 271, 299
 Tijms, H. C., 793
 Tobagi, F., 800
 Tobias, P. A., 795
 Tohma, Y., 92, 120
 Tokunaga, K., 92, 120
 Tomasi, W., 798
 Tomek, L. A., 495, 569, 576
 Towsley, D. F., 576, 620, 623, 625,
 654, 656
 Treleaven, P. C., 45, 63
 Trindale, D. C., 795
 Tripathi, S. K., 349, 631, 654, 659

- Trivedi, K. S., 17, 23, 33, 36, 51, 52, 58, 62, 63, 80, 89, 92, 120, 152, 164, 170, 171, 178, 200, 201, 265, 281, 299, 299, 300, 354, 366, 372, 397, 407, 408, 418, 419, 438, 450, 455, 464, 469, 484, 486, 493, 495, 504, 507, 508, 512, 513, 516, 524, 525, 527, 531, 535–537, 541, 543, 545–547, 550–553, 555, 558, 562, 566, 567, 569, 570, 571–576, 578, 609, 618, 620, 626, 637, 641–643, 649, 652–654, 656–659, 699, 708, 752, 768, 791, 793, 797, 802
- Tsao, N.-K., 255
- Tung, C., 387, 419
- Tuteur, F. B., 46, 63, 302, 310, 349, 798
- Ushakov, I. A., 791
- Vaidyanathan, K., 708, 752, 768, 791
- Van Dijk, N. M., 551, 572, 626, 654, 659, 794
- Van Groenendaal, W., 715, 752, 799
- Van Loan, C. F., 802
- Van Moorsel, A., 768, 791
- Varaiya, P., 801
- Veeraraghavan, M., 52, 62
- Verrall, J. L., 262, 300
- Viniotis, I., 524, 525, 551, 575, 659
- Viterbi, A. J., 798
- von Neumann, J., 119, 119
- Walpole, R. E., 734, 752, 796
- Walrand, J., 794, 801
- Wang, C., 659
- Wang, H., 618, 659
- Wang, S. S., 576
- Wang, W., 535–537, 572
- Wasserman, W., 787, 791, 796
- Weide, B., 255
- Weikel, S., 165, 200
- Wetherell, C. J., 119, 120
- Whitt, W., 659
- Wilks, S. S., 741–743, 752, 796
- Williams, A. C., 598, 607, 609, 626, 659
- Williams, S., 203, 255, 315, 348
- Willinger, W., 800
- Wilson, S. P., 802
- Wirth, N., 222, 255
- Wolfe, D. A., 795
- Wolff, R., 392, 419, 435, 576, 794
- Wolfram, S., 81, 120, 697, 752
- Wolman, E., 255
- Wong, E., 618, 659
- Wong, W. E., 407, 408, 419
- Woo, L., 631, 656
- Woolet, S., 516, 524, 525, 551, 575, 576, 641, 643, 646, 647, 658, 659
- Wu, L. T., 180, 201
- Xie, M., 802
- Yadin, M., 649, 658
- Yajnik, S., 486, 573
- Yamada, S., 199, 201
- Yechiali, U., 265, 279, 300
- Yeh, R. T., 797
- Yousef, S. Y., 576
- Zahorjan, J., 349, 618, 658, 659, 798
- Zang, X., 36, 58, 63
- Zeigner, A., 709, 710, 752, 797
- Zipf, G. K., 203, 255
- Zobrist, G., 799

Subject Index

- Algebra of events, 7, 9, 10
Analysis of algorithms (programs),
 5, 253, 273, 294, 407–414, 585,
 799, 836
do while statement, 294
for statement, 253, 836
if statement, 273, 836
repeat statement, 836
while statement, 298, 836
concurrent, 503, 541, 655
hashing, 276
MAX, 97, 220, 225, 315
searching, 202
sorting, 221
structured program, 296, 836
with two stacks, 118, 403
Analysis of variance (ANOVA), 779
nonparametric (distribution-free),
 790
 two-way, 785–789
Arrival:
 PASTA theorem, 392
process, 578, 587
 Poisson, *see* Poisson process
rate, 129, 204, 578, 588, 674, 718,
 758
ATM multiplexer, 389
Autocorrelation function, 309, 311
Autocovariance, 714
Automatic protection switching, 512
Automatic repeat request (ARQ), 77
Availability, 333, 335
 analysis, 332–340, 448, 457, 475
 application software, 485, 486
Combined with performance, *see*
 Performability, 504–519, 569
estimation, 705, 707
hardware-software, 485, 486
instantaneous (or point), 333
interval (or average), 337
limiting (or steady-state), 335–337
model, 448, 449, 453, 459,
 474–495, 505, 508, 513, 543,
 568, 569
base repeater, 340
BTS system, 342
hierarchical, 493
multiprocessor, 508
WFS system, 338, 482
with detection delay, 478
with imperfect coverage, 479
of base transceiver system, 570
of standby redundant system, 453,
 486, 489
random-request, 339
Axioms of probability, 14
Balance equation, 430, 475, 497, 579,
 597
Base repeaters, 512
Base transceiver system, 51, 62, 247,
 342, 570
Bayes' rule, 40
 continuous analog, 260
Bernoulli process, 313
 interrupted, 374
 Markov modulated, 373
 nonhomogeneous, 315

- Bernoulli theorem, 250
 Bernoulli trials, 47–49, 67, 74, 82,
 85, 250, 296, 313
 generalized, 53
 nonhomogeneous, 51, 59
 Birth–death process
 continuous-time, 428
 discrete-time, 400, 407
 finite, 445–454
 Birth:
 pure birth process, 465–469
 rate, 428, 445, 456, 465
 Blocking probability, 505
 Bottleneck, 581, 584
- C program, 202
 Cache memory, 383, *see* Memory,
 cache
 direct-mapped, 383
 fully associative, 383
 miss, 382, 411
 set-associative, 383
 Cauchy–Schwarz inequality, 214, 234
 Central server model, 585, 591, 605,
 619, 623, 640
 Central-limit theorem, 251
 Channel diagram, 41, 354
 Chapman–Kolmogorov equation,
 356, 422
 Chebyshev inequality, 248, 249, 682
 Client-server system, 460, 462, 611,
 632
 Cluster analysis:
 k-means clustering, 709
 Codes:
 error-correcting, 49, 173, 293
 Hamming, 49, 60, 173
 repetition, 58
 Coefficient:
 confidence, 682, 691
 of correlation, 214, 753
 of determination, 764
 of determination, 762–764
 of variation, 207, 226–237
 Cold replication, 486
- Combinatorial problems, 19–24, 59
 Communication channel, 37, 41, 91,
 228, 432, 704
 binary, 41, 49, 60, 153, 353, 359,
 361, 371
 feedback, 96
 probability of transmission error,
 96, 153
 trinary, 46
 Communication network, 463
 Completion time distribution, 449
 Computer system:
 analysis, 1, 304, 737, 741, 761,
 773, 776
 performance analysis, 577, 654
 reliability analysis, 165, 290
 server, 671
 Concentrated distribution, 206
 Conditional:
 distribution, 257, 261, 308
 expectation, 273, 757
 mean exceedance, 278
 MTTF, 535
 pdf, 257, 259, 260, 267, 282, 469
 pmf, 96, 257, 258, 266, 270, 272,
 314, 324, 352
 probability, 24–26, 39, 257, 260,
 336
 reliability, 133
 transform, 283
 Confidence band, 747
 Confidence interval, 662, 681,
 683–686, 688, 691, 698, 703,
 705, 715, 717, 720, 746
 distribution-free, 701
 in linear regression, 765–768
 Continuous-time Markov chain
 (CTMC), 421
 Convolution:
 of densities, 112, 176
 theorem of generating functions,
 112
 theorem of Laplace transforms,
 830
 Correlation analysis, 772–774

- Counting process, 501
Covariance, 212, 310
Coverage, *see* Fault, coverage factor, 280, 289, 479, 494, 509, 523, 534, 535, 539 imperfect, 280, 290 parameter, 280 probability, 479 estimation, 699
CPU (central processing unit), 368, 761, 777 quantum-oriented, 435 scheduling discipline, 433 service time distribution, 140, 237, 252, 451 utilization, 451, 452, 460, 594, 595
Cumulative distribution function (CDF), 71
Cumulative hazard, 132
Cyclic queuing model, 592 for availability analysis, 451 of multiprogramming system, 450
Data structure analysis, 401–402
Database server, 237
Death:
process, 470–473
rate, 428
Defective distribution, 152, 447, 449, 469
Degree of freedom, 155, 192, 741–743, 767, 773, 782, 787, 790, 818
Dependent (response) variable (in regression), 757, 779
Dependent failures, 246, 263
Design of experiments, 779
Detection delay, 477
Device:
paging, 640
Discrete-event simulation, 504, 640, 648
Discrete-time Markov chain (DTMC), 351
Disk, 159, 209, 215, 398, 605, 628, 757, 779 average seek time, 159, 215, 758 scheduling, 628, 779
Distribution function, 71, 72 continuous, 122 discrete, 70 joint (or compound), 105, 110, 159 marginal, 105, 160 mixed, 125
Distribution of products, 199
Distribution of quotients (ratios), 195
Distribution of sums, 174, 179, 182, 191, 223, 290
Distribution:
 F (variance ratio), 196, 704, 783, 787, 819–822
 χ , 818
 t (Student's), 197, 198, 685, 773, 817
Bernoulli, 73, 672, 804
binomial, 74, 77, 81, 227, 249, 804, 808–812 normal (Laplace)
approximation, 148, 252 Poisson approximation, 88, 316 reproductive property, 113 bivariate normal, 757, 772, 774
Cauchy, 252 chi-square, 139, 686–692, 703, 781, 818 relation to Erlang and gamma distribution, 159, 193 relation to Erlang distribution, 158 relation to normal distribution, 155, 192 reproductive property, 193 completion time, 449 concentrated, 206 continuous uniform, 122, 230 Coxian stage-type, 287, 474, 623 defective, 152, 174, 272, 447, 449, 469

- Distribution: (*Continued*)
- diffuse, 206
 - discrete uniform, 93, 226
 - double-exponential, 150
 - Engset, 462
 - Erlang, 137, 138, 153, 179, 182, 222, 242, 268, 282, 331, 466, 471, 559, 690, 698, 805
 - relation to Poisson distribution, 181
 - exponential, 125–128, 131–134, 137, 162, 179, 181, 185, 204, 231, 235, 243, 309, 343, 805
 - generating a random deviate, 156
 - relation to Poisson distribution, 129, 135, 168, 317
 - gamma, 138, 139, 155, 192, 193, 671, 672, 674, 805
 - Gaussian, 143, 237, 245, 251
 - geometric, 82–84, 114, 228, 804
 - heavy-tailed, 278
 - hyperbolic, 150
 - hyperexponential, 140, 233, 234, 266, 671, 672, 805
 - hypergeometric, 91, 92
 - hypoexponential, 135, 136, 139, 182, 183, 186, 188, 223, 232, 287, 474, 475, 805
 - as a series of exponentials, 182, 183, 223, 473
 - generating a random deviate, 190
 - reproductive property, 183
 - log-logistic, 142, 235, 469, 470, 806
 - log-normal, 158
 - mixture, 140, 266, 268, 272, 287, 296
 - modified geometric, 84, 87, 116, 117, 229, 271, 402, 804
 - modified negative binomial, 87
 - multinomial, 107, 108, 321
 - negative binomial, 85–87, 114, 139, 315, 804
 - normal, 143–145, 155, 157, 191, 192, 237, 245, 251, 805
 - reproductive property, 191
 - standard, 144, 154, 192, 816
 - truncated, 147
 - Pareto, 150, 236, 262, 278, 806
 - Poisson, 134, 181, 271, 320, 674, 742, 804, 813–815
 - reproductive property, 113
 - truncated, 443
 - power-law, 150
 - Rayleigh, 199
 - three-parameter Weibull, 141
 - upside-down bathtub, 142
 - webpage requests, 203
 - Weibull, 141, 142, 235, 468, 677, 763, 806
 - generating a random deviate, 156
 - with mass at infinity, 152, 174, 447, 449, 469
 - with mass at origin, 124, 174, 444
- Downtime, 341, 478, 481, 509
- Drum:
- paging, 605
 - scheduling, 454
- Duplex system, 521, 523, 539
- Effect:
- interaction, 785
 - main, 785
- Empirical distribution, 669, 744, 750
- Equivalent failure rate, 458
- Erlang's *B* formula, 444, 464, 506
- Erlang's *C* formula, 440
- Error analysis, 254
- Error function, 146
- Estimation, 663
- availability, *see* Availability
 - estimation
 - interval, 662
 - Bernoulli distribution, 694
 - exponential distribution, 689
 - imperfect coverage, 699

- normal distribution, 683
Weibull distribution, 693
parameter:
 exponential distribution, 675
 Goel–Okumoto model, 679
 Semi-Markov process, 708
software reliability, 679
 Weibull distribution, 677
slopeseeSlope estimation, 770
 with dependent samples, 714–717
Estimator, 194, 664
 biased, 666
 consistent, 669
 efficient, 667
 maximum-likelihood, 672–676,
 689, 691, 742, 765, 772
 method of moments, 670–671
 unbiased, 664
Event space, 38, 49, 67
Event(s), 6
 algebra, 7
 cardinality, 8
 collectively exhaustive, 10
 complement, 7
 elementary, 6
 impossible (null), 6
 independence, 26
 intersection, 8
 measurable, 17, 121
 mutually exclusive (disjoint), 8, 10
 union, 8
 universal, 9
Expectation, 201, 210, 238, 755
 of functions, 209–215, 274
Extended reachability graph, 570
- Factor, 779, 785, 786
 level, 779, 785, 786
Factoring method, 36, 44, 52
Failure, 208
 density (pdf), 130, 132
 rate, 131, 132, 204, 245, 759, 763
 constant, 131
 decreasing (DFR), 134
 equivalent, 458
estimation, 675
 increasing (IFR), 134, 147
Failures in time (FIT), 131
Fault tree, 33–35, 38, 45, 46, 61
Fault:
 coverage, 280, 521, 540, 702, 776
 detection, 184, 474, 539–540, 702,
 761
 latent, 489
 recovery, 278
 tolerance, 801
 hardware, 31
 software, 45, 278
File server, 687
Final value theorem, 335, 832
Finite population model, 456, 460,
 462, 590, 609, 611, 632
Fixed-point equation, 464
Fixed-point iteration, 464
Floating-point numbers:
 distributions related to, 125, 159,
 200, 254, 751
FTP bursts, 150
Functions of normal random
 variables, 191
- Gamma function, 138
Generalized Goel–Okumoto software
 reliability model, 469
Generalized Stochastic Petri net
 (GSPN), 558, *see* Petri net
Goel–Okumoto software reliability
 model, 468, 679, 752
Gokhale–Trivedi software reliability
 model, 469, 681
Goodness-of-fit test, 740–751
 chi-square, 741–743
 Kolmogorov–Smirnov, 743–751
- Hazard rate (*see* Failure, rate), 131
Hierarchical model, 464, 493,
 504–519, 530, 538, 628, 631, 634,
 636, 637, 639, 640, 648, 650, 653
Hypothesis testing, 662, 718, 773,
 781, 786

- Hypothesis testing, (*Continued*)
 descriptive level of, 722, 735, 775
 power of, 718, 728
 significance level of, 718, 773, 781, 783, 787
 type I error, 718–726, 781
 type II error, 718–726
- Hypothesis:
 acceptance of, 718
 alternative, 718
 concerning two means, 732–737, 781
 distribution-free
 (nonparametric), 735, 737
 concerning variances, 738–740
 critical region of, 718
 null, 718
- Improper distribution, 152
- Independence:
 of events, 27
 mutual, 29
 pairwise, 29
 of random variables, 110
 mutual, 111, 161, 225
 pairwise, 111
- Independent process, 308
- Independent replications, 715
- Inference:
 statistical, 661
- Initial value theorem, 830, 832
- Integration by parts, 205, 238
- Interactive system, 460, 462, 609, 611, 620, 632, 639, 784
 saturation, 461
- Interarrival time, 126, 129, 303, 314
- Inverse transform method, 157
- Jackson's result, 583–589
- Jensen's method, 547
- Job scheduling, 590
- Joint distribution, 159
- Joint probability density function, 161, 674, 754
- Joint probability mass function, 105, 580, 587, 596, 625, 674, 759
- Kolmogorov's backward equation, 424
- Kolmogorov's forward equation, 424
 integral form, 428
- Kolmogorov–Smirnov test, 743–751, 824, 826
- Kronecker's delta function, 409, 428
- Laplace transform, 334, 828
 inverse, 225, 831
 use in solving differential equations, 466, 520, 831
- Laplace–Stieltjes transform, 217, 231, 267, 274, 283, 292
- Law of large numbers, 3, 251
- Least-squares curve fitting, 753, 758–761
 higher-dimensional, 776–778
- Life testing, 675, 688, 728, 730
- Lifetime, 126, 130, 162, 268, 284, 286, 326, 526, 693
 mean (*see* MTTF), 201
 residual, 342
- Likelihood function, 673, 765, 772
- Linear dependence, 754, 772
- Little's formula, 433, 461
- Log-logistic software reliability model, 469
- Logarithmic Poisson software reliability model, 469
- M/D/1*, 396
- M/E_k/1*, 305, 398, 713
- M/E_m/1/n + 1*, 559
- M/G/1*, 391, 627
 embedded Markov chain, 392
 output process, 582
 P–K mean-value formula, 396
 P–K transform equation, 395
 with feedback, 398
- M/G/∞*, 324, 325
- M/H_k/1*, 398

- $M/M/1$, 271, 431, 556, 631
 mean response time, 433, 587
 output process, 580
 parameter estimation, 703–705
 response-time distribution, 436
 with feedback, 589
- $M/M/1/n$, 438, 557
- $M/M/2$
 heterogeneous servers, 499, 504
- $M/M/\infty$, 444
- $M/M/c/n$, 509, 511, 561
- $M/M/i/n$, 559
- $M/M/m$, 438–443, 506, 589
- $M/M/n/n$, 561
- Machine repairman model, 456–461
- Mann–Kendall test, 768–770, 827
- Marginal density, 161, 260, 674
- Marginal distribution, 105, 160
- Markov chain, 308
 n -step transition probabilities, 356
 absorbing state, 407, 425, 519–538
 aperiodic, 363
 automated generation, 552
 availability models, 474
 continuous-time, 421
 homogeneous, 422, 424
 nonhomogeneous, 424, 551
- discrete-time, 351, 702
- embedded, 392, 399
- finite, 357, 407
- fundamental matrix, 409
- homogeneous, 309, 352, 371
- irreducible, 364, 426
- limiting probabilities of, 362, 426
- mean recurrence time, 363
- numerical solution, 547
- parameter estimation, 702–705
- periodic, 360, 363
- recurrent, 362, 426
- reliability models, 519
- state aggregation, 539
- state classification, 362
- stationary distribution of, 366
- steady-state probabilities, 365, 426
- steady-state solution, 542
- symbolic solution, 546
- transient solution, 546
- transition probability matrix, 353,
 369
- transition rate, 423
- Markov inequality, 247
- Markov modulated Poisson process
 (MMPP), 501
- Markov property, 356, 421
- Markov reward model, 427, 446, 507,
 509, 538
- Mathematical induction, principle,
 14, 361
- Matrix exponential, 427, 547
- Mean, 201
 population, 664, 719–730, 780,
 804, 805
 sample, 211, 250, 664, 759, 780
- Mean response time, 460
- Mean squared error, 755
- Mean value analysis (MVA), 613
- Measurement-based model, 708
- Median, 201, 701, 732, 735
- Memory:
 cache, *see* Cache memory
 main, 208, 670, 758, 775
 paged, 346, 455, 603, 787
- Memory: cache, 383
- Memoryless property:
 of exponential distribution, 125,
 128, 345
 of geometric distribution, 84, 371
- MMPP/ $M/1$ queue, 501, 504
- Mobile communication, 674
- Mode, 201
- Model:
 binomial, 75, 76
 Markov reward, 433, 562
 parameters, 75, 194, 224, 753, 772,
 786, 804, 805
 performability, 568
 Poisson, 3, 751
 validation, 2
- Moments, 205, 226
 central, 206

- Mortality curve, 134
 MTBF (mean time between failures), 333
 MTTF (mean time to failure), 204, 238, 242, 245, 293, 718
 computation, 526
 conditional, 534
 equivalent, 458
 estimation, 675, 689
 of k -out-of- n system, 243
 of BTS system, 247
 of hybrid k -out-of- n system, 290
 of interconnection network, 244
 of parallel system, 241, 246, 521
 of series system, 239, 246
 of shuffle exchange network, 244
 of standby redundant system, 282, 287, 289
 of TMR and TMR/simplex systems, 243
 of WFS system, 244
 MTTR (mean time to repair), 336
 Multi-level model, 464, 493, 504–519, 530, 538, 628, 631, 634, 636, 637, 639, 640, 648, 650, 653
 Multinomial:
 coefficient, 53
 distribution, 107
 expansion, 58
 Multiplication rule, 25, 260
 continuous analogue, 260
 generalized, 37, 352
 Multiprocessor system, 172, 246, 376, 440, 508
 memory interference, 377, 379
 performability, 568
 Multiprogramming, degree (level), 450, 592, 640, 753, 787
 Musa–Okumoto software reliability model, 469

 Near-coincident fault, 281, 478
 Network of queues see Queuing networks(s), 577
 Non-birth–death processes, 474–495

 Non-product-form queuing network (NPFQN), 628–637
 Nonhomogeneous continuous-time Markov chain (NHCTMC), 424, 427, 428, 524, 551, 572, 575
 Nonhomogeneous Markov model (NHCTMC), 523, 524
 Nonhomogeneous Poisson process (NHPP), 320, 467
 Normal equations of least squares, 759
 Numerical solution:
 steady-state, 542
 transient, 546

 Operating system, 440, 739
 Operating system availability, 485
 Order statistic, 164
 of exponential distribution, 187
 Outlier, 737
 Overhead, 777

 Paging, 347, 380, 640, 777
 thrashing, 455, 606
 Partial fraction expansion, 223, 831
 Perceived mean:
 queue length, 438, 504
 response time, 438
 Percentile, 702
 Performability, 180, 504–519, 538, 568, 572, 573, 576, 636
 Performability model, 180, 504–519, 538
 Performance evaluation, 662, 796
 Performance model, 382, 463–464, 496–503, 505, 509, 510, 565, 569
 web browser, 611
 Petri net, 552
 arc, 552
 inhibitor, 558
 input, 553
 multiplicity, 553
 output, 553
 enabling function, 562
 guard function, 562

- marking, 552
 - tangible, 558
 - vanishing, 558
- marking dependent:
 - arc multiplicity, 562
 - firing rate, 562
- place, 552, 553
 - input, 553
 - output, 553
- reachability graph, 553, 556, 557
 - extended, 558, 559
- reward rate, 562
- stochastic, 555, 649
 - generalized, 558
 - reward net, 562
- transition, 552, 553
 - firing time, 553, 555
 - immediate, 558
 - properties, 558
 - race policy, 555
 - timed, 558
 - vanishing loop, 558
- Poisson arrivals see time averages (PASTA), 392
- Poisson process, 309, 317, 465, 555
 - compound, 326
 - decomposition of, 321, 326
 - Markov modulated, 501
 - nonhomogeneous, 320, 339, 467, 468
 - superposition, 320
- Pollaczek–Khinchin:
 - mean-value formula, 396
 - transform equation, 395
- Population, 661
 - distribution, 738, 742, 765
 - parameters, 765
- Power method, 542
- Preemptive repeat, 450
- Preemptive resume, 450
- Preventive maintenance, 475
- Probability, 791
 - assignment, 14, 18
 - axioms, 14
 - conditional, 24
- measure, 17
- models, 2
- tree, 39
- Probability density function (pdf),
 - 122, 805
 - Cauchy, 198, 252, 665
 - exponential, 126, 204
 - joint (or compound), 161, 674, 754
 - marginal, 161, 260, 674
 - reciprocal, 125, 200
 - truncated normal, 146
- Probability generating function (PGF),
 - (PGF), 100–102, 217, 226–230, 292, 804
 - convolution property, 112
 - uniqueness property, 101
- Probability mass function (pmf),
 - 68–93, 804
 - joint (or compound), 105, 673, 759
 - marginal, 105, 580
- Probability plot:
 - exponential distribution, 749
 - Weibull distribution, 749
- Producer–consumer system, 566
- Product-form queuing
 - network(s)(PFQN), 582–620
- Program performance, 97, 104, 117, 118, 202, 221, 253, 276, 294, 346, 368, 401, 407, 411
 - concurrent, 176, 296, 503, 541, 566, 655
- Program:
 - control flow graph, 407
 - paging behavior, 346, 380
 - renewal model, 346
- Qualitative variable, 779
- Quality control, 732
- Quality of service (QoS), 505
- Queuing, 577, 793
 - network(s)see Queuing network(s), 577
 - notation, 303
 - Tagged customer, 648
 - theory, 303, 577, 793

- Queueing network(s), 577
 closed, 577, 590–611
 multiple job types, 624
 nonexponential service time distributions, 620, 627
 normalization constant, 596, 598–608, 626
 utilization, 601, 609, 619, 626
 joint probability, 580
 marginal probability, 580
 non-product-form, 628–637
 normalization constant, 618
 open, 577, 582–589
 product-form solution, 580, 581, 594, 598, 623, 626
 relative throughput, 593–595, 597, 618
 relative utilization, 593–595, 601, 603, 618, 619
- Random deviate (variate), 155, 156, 273
 Random experiment, 3
 Random incidence, 342–345
 Random interval, 681
 Random number, 155, 305
 Random process, 301
 Random sample, 191, 250, 663, 683, 718, 765, 772
 Random sums, 290
 Random variable(s), 121
 absolutely continuous, 122
 Cauchy, 252
 continuous, 121, 122, 201
 dependent, 758
 discrete, 68, 71, 73, 83, 85, 201, 759
 expectation, 201, 209, 274
 functions, 154, 191, 209
 independent, 94, 161, 205, 217, 225, 272, 757, 772, 779
 mixed, 124
 orthogonal, 216
 uncorrelated, 213, 216, 757, 772
 Random vectors:
 continuous, 159
 discrete, 104
 Random walk, 316
 Randomization method, 547
 Rank correlation coefficient, 775
 Reachability graph, 553, *see Petri net*
 Real-time system
 hard, 538
 soft, 537
 Reconfiguration, 508
 Recurrent process, 327
 Redundancy, 31
 k-out-of-*n*, 243
 hybrid *k*-out-of-*n*, 243, 495
 parallel, 31, 240, 246, 495
 standby, 178, 241, 495
 subsystem level, 60, 242, 495
 system level, 60
 triple modular, 50
 Regression, 663, 753, 765
 linear, 756
 nonlinear, 775
 Relative frequency, 3, 682
 Reliability, 38, 130, 280, 675, 692, 794
 Combined with performance, 180, 538, *see* Performability
 conditional, 133
 dependent components, 263
 hierarchical model, 531
 Markov models, 519–523
 nonhomogeneous Markov model, 523
 of *k*-out-of-*n* system, 49, 59, 77, 164, 168, 174, 187, 190, 525
 of base repeater, 38
 of base transceiver system, 51, 62
 of detector-redundant systems, 59
 of diode configurations, 55
 of fault-tolerant software, 532
 of hybrid *k*-out-of-*n* system, 189, 290, 473
 of interconnection network, 170

- of memory, 293
- of non-series-parallel systems, 44
- of nonidentical k -out-of- n system, 51
- of parallel systems, 30, 60, 165, 166, 472, 519
- of real-time system
 - hard deadline, 538
 - soft deadline, 537
- of recovery block, 532
- of series systems, 30, 59, 163, 245, 262, 271, 321
- of series-parallel systems, 32
- of shuffle exchange network, 170
- of software, *see* Software reliability
- of standby redundant system, 180, 281, 284, 288, 471, 532
- of standby redundant systems, 179, 282, 539
- of TMR system, 50, 59, 168, 185
 - with spare, 525
- of TMR/simplex system, 183
- of WFS system, 528
- product law, 31
- statistics, 692
- temperature-dependence, 275
 - with imperfect coverage, 280, 521
- Renewal counting process, 309, 328
 - decomposition, 331
 - superposition, 345
- Renewal density, 329
- Renewal process, 327–331
- Response time, 578, 587
 - distribution, 436, 471, 473, 511, 581, 641, 648
 - mean, 398, 433, 438, 499, 504, 584, 628, 630, 639
- Reward function, 353, 427, 562
- Safety, 523, 525, 534, 539
- Sample correlation coefficient, 774
- Sample mean, 192, 211, 250, 664, 668, 689, 714, 759, 780
 - variance of, 251, 665
- Sample space, 3, 5, 65
- continuous, 6
- countably infinite, 5
- discrete, 5, 17
- finite, 5, 19
- partition, 10
- sequential, 12, 66
- uncountable, 5, 17
- Sample variance, 666, 668, 685, 689, 723, 734
- Sampling distribution, 662, 664, 747
- Sampling:
 - from Bernoulli distribution, 694
 - from exponential distribution, 191, 689, 729, 747, 765
 - from finite population, 663, 665, 667
 - from normal distribution, 191, 683, 733, 747
 - from Weibull distribution, 693
- Saturation number, 461, 619
- Scatter diagram, 754, 774
- Scheduling discipline, 397, 433, 623
 - BCC (blocked calls cleared), 443, 462
 - FCFS (first-come, first-served), 304, 391, 433, 455, 578, 581, 623
 - PS (processor sharing), 435, 623
 - RR (round robin), 435
 - SLTF (shortest latency time first), 454
 - SRPT (shortest remaining processing time first), 435
- Searching, analysis, 202, 203
- Security modeling, 531
- Semi-Markov process, 490
 - parameter estimation, 708
- Sen's slope estimator, 770–771
- Service:
 - rate, 204
 - time distribution, 578
- Shuffle exchange network (SEN), 169
- Sigma field, 17, 121
- Sign test, 732
- Simulation, 751, 799

- Slope estimation, 768
 Sen's method, 770
- Software aging, 768
- Software availability, 485, 486
- Software fault tolerance, 45, 278, 486, 532
- Software performance, 97, 104, 117, 118, 202, 221, 253, 276, 294, 296, 346, 368, 401, 407, 411, 414
- Software reliability, 45, 259, 261, 278, 372, 411, 414, 486, 532, 679, 802
 growth model, 262, 468, 470, 473
 generalized Goel–Okumoto, 469
 Goel–Okumoto, 468, 679, 752
 hypergeometric, 92
 Jelinski–Moranda, 473
 Littlewood–Verrall model, 262
 log-logistic, 469, 681
 logarithmic Poisson, 469
- Software reliability growth model (SRGM), 92, 120, 199, 201, 259, 261, 262, 300, 354, 372, 419, 468–470, 473, 679, 681, 752, 802
- Software reliability model (SRM), 92, 120, 199, 201, 259, 261, 262, 300, 354, 372, 411, 414, 419, 468–470, 473, 532, 679, 681, 752
- Software testing, 199, 259, 261
- Sorting:
 analysis, 221
- Standard deviation, 206
- Standby redundancy, 179, 180, 281, 284, 288, 453, 465, 471, 486, 489, 532
 cold, 179, 180, 281, 453, 465, 471, 486
 hot, 284, 532
 warm, 284, 288, 489
- Standby system, 284, 465, 471, 489, 532
- Statistic, 664, 719, 773
 F (variance ratio), 740
 t , 734, 737
- chi-square, 739–742
- Kolmogorov–Smirnov, 744, 824, 826
- rank-sum, 736–737, 823
- Statistics, 191, 664, 795
 availability, 705
- Stochastic Petri net (SPN), 555
- Stochastic process, 301, 714, 792
 classification, 303, 307
 continuous state, 302
 continuous-time, 302
 discrete state, 302
 discrete-time, 302, 714
- Markov, 308, 351
- Markov regenerative, 570
- sample function (realization), 301, 304, 312
- semi-Markov (SMP), 309, 490, 570, 708
 estimation, 708
- state space, 301
- stationary, strictly, 307
- stationary, wide-sense, 310, 714
- Stochastic reward net (SRN), 562,
see Petri net, 649
- Storage allocation, dynamic, 208
- Structure function, 34
- Successive overrelaxation method (SOR), 544
 Convergence, 545
- Sum of disjoint products (SDP), 17, 35, 36, 38, 44, 46, 52, 62
- Sum of squares, 786
 about the mean, 763
 between treatments, 783
 residual (error), 763, 783
- Tandem network, 578
- Task arrival, 339
- Telephone:
 call congestion, 443, 462
 exchange, 208, 237, 440, 443, 462, 505, 521, 750
 traffic, 440
 trunks, 208

- Theorem:
- of Blackwell, 329
 - of elementary renewal, 329
 - of PASTA, 392
 - of key renewal, 329
 - of total expectation, 274
 - of total moments, 274
 - of total probability, 39
 - continuous analog, 260
 - of total transforms, 274, 283, 286
- Therorm
- Central-limit, 251
- Throughput, 451, 559, 594, 618, 635, 753, 787
- Time slicing, 47
- Time to failure, 126, 130
- Traffic intensity, 432
- Transform(s), 216, 226, 805
 - z (probability generating function), 101
 - characteristic function, 224, 225
 - convolution property, 112, 113, 217, 830
 - correspondence (uniqueness) theorem, 101, 218, 828
 - Fourier, 217
 - Laplace, 466, 472, 520–526, 828–834
 - Laplace–Stieltjes, 217, 231–234, 283, 290, 292
 - moment generating function, 217
 - moment generating property, 219, 522, 830
- Tree diagram, 11, 39, 285, 315, 403
- Trend detection, 768–768
- Uniformization method, 547
- Variance (population), 206, 211, 212, 250, 734, 738
 - sample, 666
- Variation:
- explained, 763
 - residual, 757, 779, 783
 - total, 763, 784
 - unexplained, 763
- Venn diagram, 11
- Warm standby, 489
- Web performance model, 611
- Web traffic, 150
- WFS example, 169, 244, 338, 482, 496, 528, 563, 565, 570
- Wireless:
- analysis of ARQ, 78, 293
 - availability, 570
 - model, 340, 342, 512
- cell:
- analysis, 204
 - control channel, 512
 - guard channels, 463
 - handoff calls, 463, 512, 566
 - performability model, 512
 - performance model, 463, 565
- protocol:
- Aloha, 388
 - slotted Aloha, 388
 - time division multiple access (TDMA), 23, 92
 - reliability model, 38, 51
- Workload characterization, 708